Robotica (2022), **40**, pp. 2878–2908
doi:10.1017/S0263574721002009

CAMBRIDGE
UNIVERSITY PRESS

REPLY

Kinematic and dynamic analysis of a dexterous multi-fingered delta robot for object catching

Sachin Kansal[1,*] and Sudipto Mukherjee[2]

[1]Computer Science Engineering Department, Thapar Institute of Engineering Technology, Patiala, Punjab, India and
[2]Mechanical Engineering Department, Indian Institute of Technology Delhi, New Delhi, India
*Corresponding author. E-mail: sachin.kansal@thapar.edu

Received: 9 June 2021; **Revised:** 13 December 2021; **Accepted:** 28 December 2021; **First published online:** 3 February 2022

Keywords: Delta robot, catching, RGB sensor, Recurdyn^{TM}, PID controller, multi-fingered, kinematics, dynamics

Summary

This paper presents a new combined 9-DOF [3R-3R-3R] parallel architecture to perform the object catching in the real-time scenario. The architecture design is intended to carry heavy payload objects for the catching. This paper covers modelling the new architecture, kinematics, and dynamic analyses to compute torques/forces at the actuators. The simulation of kinematic and dynamic analysis in MATLAB. Vision sensors, encoders, PID controller, and current limiting are used to perform object catching. The architecture is the only delta robot application designed to catch regular-shaped objects in real-time scenarios.

1. Introduction

This paper describes the work done so far in the field of the delta catching system. The work involves selecting the catching system, kinematic synthesis of delta manipulator, the design of components, dynamics analysis of the system, and control analysis.

Boney [1] has explained the story of the success of the delta parallel robot. Robots achieve acceleration up to 50g in experiments and 12g in industrial use. Delta robot is perfect for application where the light object must be placed from one place to another (10 g to 1 kg).

Murray [2] has discussed the kinematic analysis of the delta robot (Clavel's) by the geometric method. The geometric configuration of Clavel's delta manipulator is explained in detail. Tsai [3] has discussed the position analysis of the University of Maryland manipulator. The manipulator's graphical and algebraic solutions for direct and inverse kinematic analysis have been explained. Codourey [4] has demonstrated the dynamic modelling of robots (parallel) estimated torque control implementation. The number of forces and motion of the end effector is significantly less in the concept of micromanipulation.

Laribi [5] has explained the dimensional analysis of the delta robot for a specified workspace, i.e., designs a dimensional configuration of a delta manipulator for a given workspace. Kosinska [6] has explained the optimization of variables of the delta robot for a workspace. This paper has presented the methodology of deriving design constraints from the closed-loop configuration of the delta-4 parallel manipulator. The initial conceptual design has been taken from Clavel's delta configuration.

Tsai [7] has explained the methodology based on the principle of virtual work for static analysis of parallel robots.

The dynamic analyses of the multibody system are required to know the trajectory of the different coordinates for a given torque or for a required trajectory what should be the torque at another degree of freedom. There are various methods available for the formulation of the multibody system. Angeles and Lee [8] developed and implemented a computationally efficient and systematic algorithm for formulating the dynamic model of a constrained mechanical system. The dynamic equations of a mechanical system composed of rigid bodies coupled by holonomic constraints are derived using an orthogonal complement

© The Author(s), 2022. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

of the matrix of the velocity constraint equations. The kinematic constraints are expressed as a system of dependent homogeneous linear equations on the twist of the different rigid bodies. Angeles and Ma [9] used the concept of natural orthogonal complement used in the derivation of an $n \times n$ inertia matrix of an n-axis manipulator. The two methods are presented to compute the joint acceleration through the governing equation of motion. The first method is based on the Cholesky decomposition of the inertia matrix. The second method is based on a composition of the minimum norm and least square approximation of under-determined and over-determined linear algebraic systems, respectively.

Saha [10] states that inertia matrix manipulator decomposition is necessary. The joint accelerations are computed through the equations of motion. The recursive relations between the elements of the computed matrices reveal through the Gaussian elimination of the inertia matrix. Saha [11] has derived restricted dynamic equations of motion for the serial multibody systems. It has rigid bodies in the serial kinematic chain. The Euler–Lagrange independent equation of motion has been calculated. The Newton–Euler equations of motion of uncoupled bodies were used to obtain a recursive inverse and forward dynamic algorithm required to control and simulate the mechanical system. Saha [12] explained the method of recursive dynamics algorithms for multibody systems with serial, parallel, and closed-loop chains using the concept of decoupled natural orthogonal complement. The definition of the NOC and DeNOC and features of the DeNOC matrices are presented. The method illustrated forward and inverse dynamics with several multibody systems, e.g., a parallel manipulator and six DOF serial manipulators.

The DeNOC matrices can be written recursively for the number of links of a multibody system. The DeNOC method provides the equation of motion in terms of the joint angle, the same as the Euler-Lagrange formulation (energy-based approach). In this formulation, Newton–Euler (linear and angular) equation of motion is transferred to angular only (in terms of joint angles). This method reduced the large set of the equation to a minimal set of the equation by converting the equation of motion in terms of joint angle only, e.g., for one link Newton–Euler gives two equations of motion, while using the DeNOC, it is converted to only one equation of motion [13].

2. Robots for Catching

Through the years, researchers and engineers have developed various catching-based robots [14–18]. Major catching-based robotic systems were developed in the mid-1990s and 2000s. Philip W. Smith et al. [19] proposed vision-based robust robotic catching algorithms that are used for objects having non-rigid in nature. It is based on an image-based representation of the non-rigid structure and relative elasticity with no a priori physical model. The relative elasticity has many advantages as simplicity, comprehensiveness, generality. This method overcomes many limitations of existing non-rigid object catching.

Satoshi Yonemoto et al. [20] proposed mapping real-time human action with its corresponding virtual environment. In this approach, the scene constraints fall between the user motion and the virtual object. This approach has resulted in the action information and non-trivial to have the body postures representation. Yejun Wei et al. [21] described non-smooth domains and demonstrated and verified a motion planning algorithm for the system. This approach performs catching on a non-smooth object using four fingers. Park et al. [22] described a method that takes the visual data to produce force guidance data and an applicable tele-manipulative system. Mkhitaryan et al. [23] presented a vision-based haptic multisensory. The system interacts with objects of having fragile nature. Chen et al. [24] revealed the significance of active lighting control in the domain of robotic catching. Kansal et al. [25] proposes the vision-based kinematics analysis of the delta robot for object catching.

In this paper, robot catching application presented introduces a new robot application domain. A new manipulative parallel robot system is developed, entailing the design and analysis of a new system. In Section 2 (A), the proposed methodology has been discussed. Section 2 (B) discusses mechanical structure design. In Section 3, kinematic analysis and experimental setup used is discussed. In Section 4, dynamic analysis is discussed. In Section 5 catching the object is examined. Results and validation are discussed in Section 6.

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

Algorithm 1: *Extracting the pose of the cube, i.e., (three translations and three rotations and it come out to be 6-DOF) with respect to the fixed base coordinates system using the camera as a vision sensor. After that, inverse kinematics is executed for all three delta robots. The thetas (θ_1 to θ_9) from the three delta robots feeding the calculated thetas to the controller of the manipulating system is executed. Getting feedback from the encoder and PID control scheme is also implemented. The tuning of the PID values is achieved at every cycle using the optimizing technique (Genetic Algorithm). Dynamic analysis of the system using Decoupled Natural Orthogonal Complement (**DeNOC**) technique is applied; validation of the dynamic analysis results with the **Recurdyn**^{TM} software is also achieved. Validation of the kinematic analysis results with the **Recurdyn**^{TM} software is also completed. Knowing the torque v/s theta behaviour (both the directions) of the voice coil actuator (VCA), i.e., the actuator, the estimation of torque by giving the theta corresponding is also validated with the dynamic analysis results.*

Input: Three Poses [R_{vec}, $T_{vec]\,i}$ of the cube to the fixed base frame (i = No. of cameras). In this case, three cameras are used, i.e., poses will be [Pose1, Pose2, Pose3]. The link length of the lower leg ($\boldsymbol{l_l}$), link length of the upper leg ($\boldsymbol{l_u}$), internal camera matrix ($\boldsymbol{IC_{3x3}}$), joint angle (θ_n) range [0^o *to* $+30^o$].
Output: *Corresponding Thetas* [θ_1 to θ_9] and torques [Γ_1 to Γ_9], for each of the VCA actuator to do the catching.
Begin
Thread Cam$_1$. Thread Cam$_3$ [Cam$_1$, Cam$_2$, Cam$_3$ for three cameras and manager thread [Manager $_{Thread}$] for control the cameras threads
While (true)
1. Perform the Inverse kinematics, taking the position to the fixed base frame **[B_{ff}]** as input.
2. Perform the Velocity based kinematics, i.e., time-dependent kinematic analysis.
3. Perform the Inverse Dynamics, i.e., estimating the joint torques for the given input trajectory
4. A basic control strategy is implemented for demonstration purposes **[PID controller].**

End while
Begin
Validating the kinematic and dynamic analysis using Recurdyn^{TM} *software offline.*
End

2.1. Proposed method

Three parallel robots are arranged in symmetrical order with vision sensors and hardware interfacing, as shown in Fig. 1. This paper is limited to the kinematic and dynamic analyses of the system to catch the cube. In this method, a new architecture that comprises of three delta robots and each of the end effectors behave like one finger, so it comprises three fingers that do the catching in the real-time scenario. Three pre-calibrated vision sensors (Basler scout cameras) are used to estimate the pose of the three cube faces. The forward kinematics for all three fixed delta robots is achieved to know the shared workspace (W_{catch} = Intersection (W1, W2, W3)).

The delta robots are used as fingers because this architecture can go for a high payload and is dedicated to high-speed applications like manipulation. In this architecture, the moving frame has no rotation to the fixed frame, i.e., 3DOF has the only translation in the X, Y, Z axes, as shown in Fig. 2.

2.2. Mechanical structure

The delta robot is the three-DOF motion platform invented by a research team led by Prof. Reymond Clavel in the early 1980s as a manipulator for light and smaller objects. Since then, it has been used for various applications in industries and elsewhere. Their applications include driving simulators, medical devices, entertainment industries, vibration isolators, etc. Delta manipulator, as shown in Fig. 3, employs revolute joints and spherical joints to give platform output to translational motion [2]. A schematic diagram of the manipulator constructed with all revolute joints. The lower arm is labelled as links 1,

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

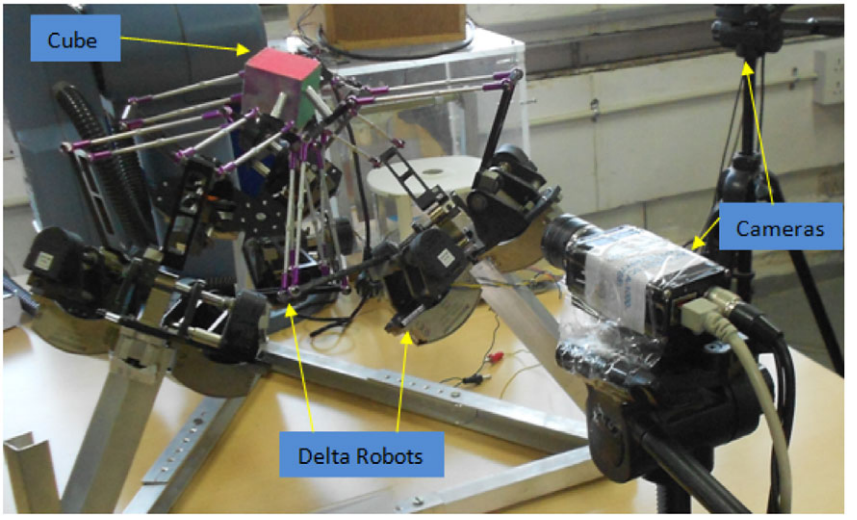


Figure 1. *Experimental setup.*

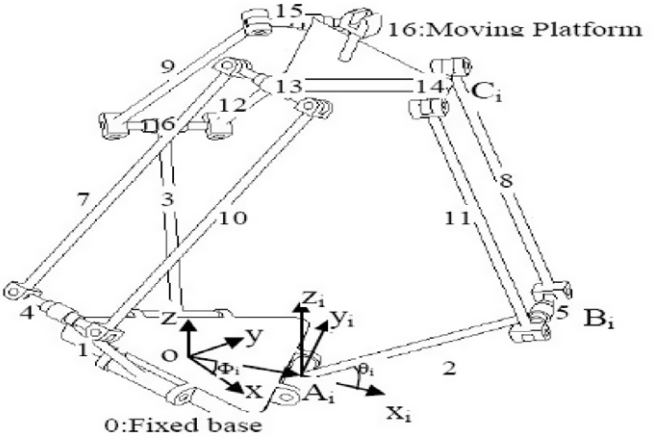


Figure 2. *Mechanical structure of individual manipulator [26].*

2, 3. The upper arms of the delta robot architecture comprise four parallelograms; links are shown in Fig. 2. Links 4, 7, 10, and 13 are marked as Ist limb, links 5, 8, 11, and 14 are marked as the IInd limb, and links 6, 9, 12, and 15 are marked as IIIrd limb.

A, B, C are the points where three parallel joints attach the lower arms, upper arms, the two surfaces, and each of the parallelograms has four revolute joints. Overall, there are 17 links and 21 revolute joints. The general degree of freedom equation is

$$n = s\ (r - p - 1) + \Sigma\, n_i \tag{1}$$

where S represents dimensions of working space (For spatial, s = 6), R represents several rigid bodies or links in the system = 17, and P represents nunmber of kinematic pairs of joints in the system = 21

n_i = relative degree of freedom of each joint.

$$DOF,\ n = = 6^{*}(17 - 21 - -1) + 21\ (1) = -9$$

The joints are redundant and have a 3DOF translation. Due to the four-bar parallelogram and the three revolute joints at points A, B, and C, the limb constrains the unfixed surface from rotating about the z and x-axis. The two limbs constraint the unfixed body from rotating about any axis. It is helpful in the x-y-z positioning device.

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

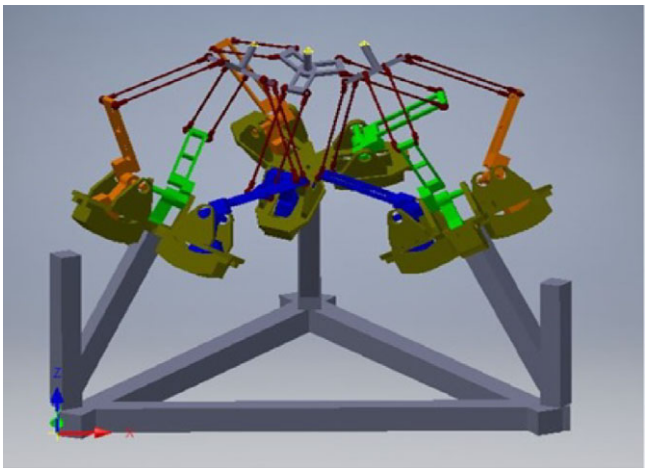


Figure 3. *Three delta combined system architecture [27].*

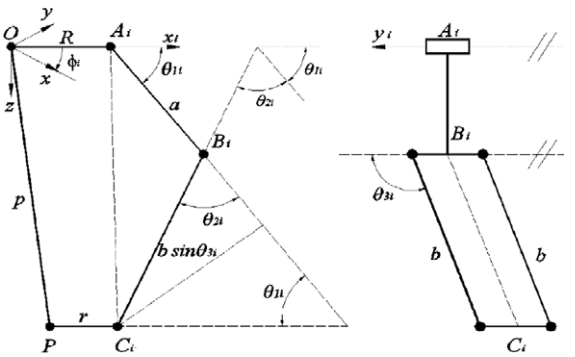


Figure 4. *Schematic diagram of delta mechanism.*

The delta manipulator used in this paper is shown in Fig. 4. The base platform was 0, and the top platform was 1, as shown in Fig. 4. The inclination of the second link due to a spherical joint cannot be shown in the X-Z plane. The angle between any two axes X, u_1, v_1 is 120^{o} and the direction of axes Z, w_1, and w_2 is the same.

3. Kinematic Analysis

3.1. System description and mathematical background

The three degrees of freedom delta robot is capable of 3D transitional of its moving surface. As in the proposed delta architecture, there are three identical Revolute-Universal-Universal (R-U-U) chains as legs. In the Figs. 5 and 6, points B_i, i = 1,2,3 are the hips, points, P_i, i = 1,2,3 are the ankles, and points, A_i, i = 1,2,3 are the knees. s_B represents the length of the equilateral triangle (base), and s_P represents the length of the equilateral triangle (moving). In our proposed delta architecture, the moving platform is inverted to the base. There is no rotation between the moving platform and the base platform, i.e., [R_{3x3} is Identity matrix]. Thus, only the translational vector between the moving and the fixed platform is to be considered.

'B' represents the cartesian reference frame (fixed). The origin is situated in the centre of the bottom triangle. 'P' is not a fixed Cartesian reference frame. The origin is located in the centre of the platform

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

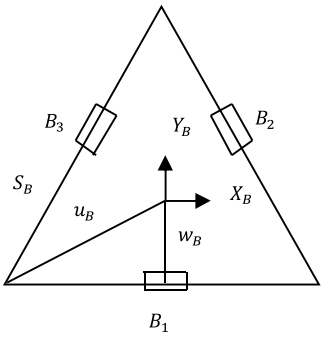


Figure 5. *Delta robot fixed base schematic.*

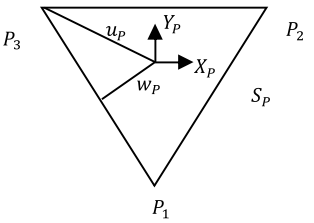


Figure 6. *Delta robot moving platform.*

triangle. The rotation matrix ${}^{B}_{P}R = \mathbf{I_3}$ always is constant as orientations of P and B are the same. The joint variables are $\theta = \{\theta_1, \theta_2, \theta_3\}^{\mathrm{T}}$, and the Cartesian parameters are ${}^{B}_{P}R = \{\mathrm{x\ y\ z}\}^{\mathrm{T}}$.The three upper legs of lengths 'L' and three lower legs of lengths 'l' have high symmetry. The geometric details of the delta robot having a fixed bottom and platform are depicted in Figs. 5 and 6, respectively.

In the bottom frame 'B', the fixed-bottom joint points B_i are constant. In the top frame 'P', the platform-fixed joint points P_i is constant. In the kinematic diagram (in Fig. 7), the three vector-loop closure equations for the parallel architecture are given below:

$$ {}^{B}_{1}B = \begin{Bmatrix} 0 \\ -w_B \\ 0 \end{Bmatrix} \quad {}^{B}_{2}B = \begin{Bmatrix} \frac{\sqrt{3}}{2}w_B \\ \frac{1}{2}w_B \\ 0 \end{Bmatrix} \quad {}^{B}_{3}B = \begin{Bmatrix} \frac{-\sqrt{3}}{2}w_B \\ \frac{1}{2}w_B \\ 0 \end{Bmatrix} $$

$$ {}^{p}_{1}P = \begin{Bmatrix} 0 \\ -u_P \\ 0 \end{Bmatrix} \quad {}^{p}_{2}P = \begin{Bmatrix} \frac{S_p}{2} \\ w_P \\ 0 \end{Bmatrix} \quad {}^{p}_{3}P = \begin{Bmatrix} \frac{-S_p}{2} \\ w_p \\ 0 \end{Bmatrix} $$

The vertices of the bottom triangle (fixed) are represented as

$$ {}^{B}_{1}b = \begin{Bmatrix} \frac{S_B}{2} \\ -w_B \\ 0 \end{Bmatrix} \quad {}^{B}_{1}b = \begin{Bmatrix} 0 \\ u_B \\ 0 \end{Bmatrix} \quad {}^{B}_{3}b = \begin{Bmatrix} \frac{-S_B}{2} \\ -w_B \\ 0 \end{Bmatrix} $$

where: $w_B = \frac{\sqrt{3}}{6}S_B$; $w_p = \frac{\sqrt{3}}{6}S_p$; $u_B = \frac{\sqrt{3}}{3}S_B$; $u_p = \frac{\sqrt{3}}{3}S_p$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

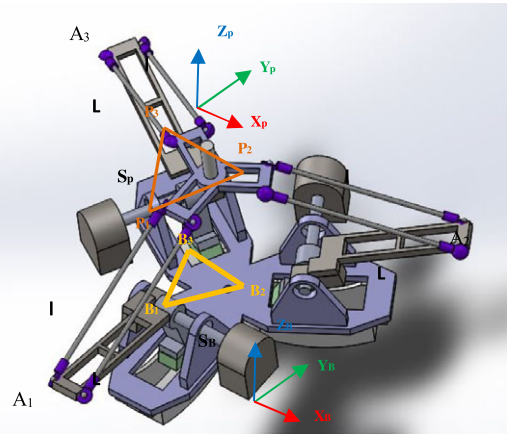


Figure 7. *Delta robot kinematic diagram.*

In the kinematic diagram, the three vector-loop equations as depicted in Fig. 7;

$$\left\{ {}_{i}^{B}B \right\} + \left\{ {}_{i}^{B}L \right\} + \left\{ {}_{i}^{B}l \right\} = \left\{ {}_{i}^{B}P \right\} + \left[{}_{P}^{B}R \right] \left\{ {}_{i}^{P}P \right\} = \left\{ {}_{P}^{B}P \right\} + \left\{ {}_{i}^{P}P \right\}$$

$$i = 1,2,3 \tag{2}$$

$\left[{}_{P}^{B}R \right] = [I_3]$. Delta robot does not perform rotations with respect to the delta fixed frame. The length of the lower leg should be correct, constant length.

$$l_i = |\left\{ {}_{i}^{B}l \right\}| = |\left\{ {}_{p}^{B}P \right\} + \{ {}_{i}^{p}P \} - \left\{ {}_{i}^{B}B \right\} - \left\{ {}_{i}^{B}L \right\}| i = 1,2,3 \tag{3}$$

It is appropriate to square constraint equations to stay away from the square-root in the Euclidean norms:

$$l_i^2 = |\left\{ {}_{i}^{B}l \right\}|^2 = l_{ix}^2 + l_{iy}^2 + l_{iz}^2 i = 1,2,3 \tag{4}$$

The Cartesian variables are ${}_{p}^{B}P = \{xyz\}^T$. The constant vector values for points P_i and B_i were given previously. The vectors $\left\{ {}_{i}^{B}L \right\}$ are dependent on the joint variables $\odot\{\theta_1\theta_2\theta_3\}^T$:

$${}_{1}^{B}\mathrm{L} = \begin{Bmatrix} 0 \\ -\mathrm{Lcos}\theta_1 \\ -\mathrm{Lsin}\theta_1 \end{Bmatrix} {}_{2}^{B}\mathrm{L} = \begin{Bmatrix} \frac{\sqrt{3}}{2}\mathrm{Lcos}\theta_2 \\ \frac{1}{2}\mathrm{Lcos}\theta_2 \\ -\mathrm{Lsin}\theta_2 \end{Bmatrix} {}_{3}^{B}\mathrm{L} = \begin{Bmatrix} -\frac{\sqrt{3}}{2}\mathrm{Lcos}\theta_3 \\ \frac{1}{2}\mathrm{Lcos}\theta_3 \\ -\mathrm{Lsin}\theta_3 \end{Bmatrix}$$

After applying the values into the vector equations, yields equation below:

$$\left\{ {}_{1}^{B}l \right\} = \begin{Bmatrix} \mathrm{x} \\ \mathrm{y} + \mathrm{Lcos}\theta_1 + a \\ \mathrm{z} + \mathrm{Lsin}\theta_1 \end{Bmatrix}; \left\{ {}_{2}^{B}l \right\} = \begin{Bmatrix} x - \frac{\sqrt{3}}{2}\mathrm{Lcos}\theta_2 + b \\ y - \frac{1}{2}\mathrm{Lcos}\theta_2 + c \\ \mathrm{z} + \mathrm{Lsin}\theta_2 \end{Bmatrix}; \left\{ {}_{3}^{B}l \right\} = \begin{Bmatrix} x + \frac{\sqrt{3}}{2}\mathrm{Lcos}\theta_3 - b \\ y - \frac{1}{2}\mathrm{Lcos}\theta_3 + c \\ \mathrm{z} + \mathrm{Lsin}\theta_3 \end{Bmatrix}$$

$$a = \mathrm{w_B} - \mathrm{u_p}; b = \frac{\mathrm{S_p}}{2} - \frac{\sqrt{3}}{2}\mathrm{w_B}; c = \mathrm{w_p} - \frac{1}{2}\mathrm{w_B}$$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

Table I. *Parameters of the delta architecture.*

Name	Meaning	Value (mm)
S_B	base side [fixed]	207.8
S_p	platform side [moving]	64
L	Upper leg length	110
l	Lower leg length [parallelogram]	136
h	Width of legs [parallelogram]	26
w_B	Distance from frame {0} to the near base side.	60
u_B	Frame {0} to a base vertex	120
w_p	From the {p} to the near platform side	18.5
u_p	From the {p} to a platform vertex	37

Table II. *The DH parameters [28] of the k^{th} leg of the delta platform.*

Link (i)	a_i (m)	b_i (m)	α_i (rad)	$\theta_i(JV)$ (rad)
1	0.110	0	$3\pi/2$	θ_{k1}
2	0	0	$\pi/2$	θ_{k2}
3	0.136	0	0	θ_{k3}

The three constraint equations result in the kinematics equations:

$$2L\,(y+a)\cos\theta_1 + 2zL\sin\theta_1 + x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 = 0 \tag{5}$$

$$-L\left(\sqrt{3}\,(x+b) + y + c\right)\cos\theta_2 + 2zL\sin\theta_2 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 = 0\ldots \tag{6}$$

$$L\left(\sqrt{3}\,(x-b) - y - c\right)\cos\theta_3 + 2zL\sin\theta_3 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 = 0 \tag{7}$$

The knee points are calculated using ${}^{B}_{i}A = {}^{B}_{i}B + {}^{B}_{i}L$; i = 1, 2, and 3:

$${}^{B}_{1}A = \begin{Bmatrix} 0 \\ -w_B - L\cos\theta_1 \\ -L\sin\theta_1 \end{Bmatrix};\ {}^{B}_{2}A = \begin{Bmatrix} \frac{\sqrt{3}}{2}(w_B + L\cos\theta_2) \\ \frac{1}{2}(w_B + L\cos\theta_2) \\ -L\sin\theta_2 \end{Bmatrix};\ {}^{B}_{3}A = \begin{Bmatrix} -\frac{\sqrt{3}}{2}(w_B + L\cos\theta_3) \\ \frac{1}{2}(w_B + L\cos\theta_3) \\ -L\sin\theta_3 \end{Bmatrix}$$

3.2. Geometric model

The delta robot is characterized by the Denavit–Hartenberg (DH) parameters, as shown in Table I. Voice coil actuator (VCA), i.e., the actuator having joint coordinates are given with variables $\theta 1$, $\theta 2$, $\theta 3$, $\theta 4$, $\theta 5$, $\theta 6$, $\theta 7$, $\theta 8$, $\theta 9$. Its geometric architecture is defined in Fig. 8 Table II.

Inverse kinematics of parallel robots is a process where actuated or active joint trajectories are to be computed for the respective poses of the platform motion. The inverse kinematics of parallel robots calculated both active and passive joint trajectories. The calculated results of inverse kinematics are needed for inverse dynamics to determine the driving torques/forces at the active joints.

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

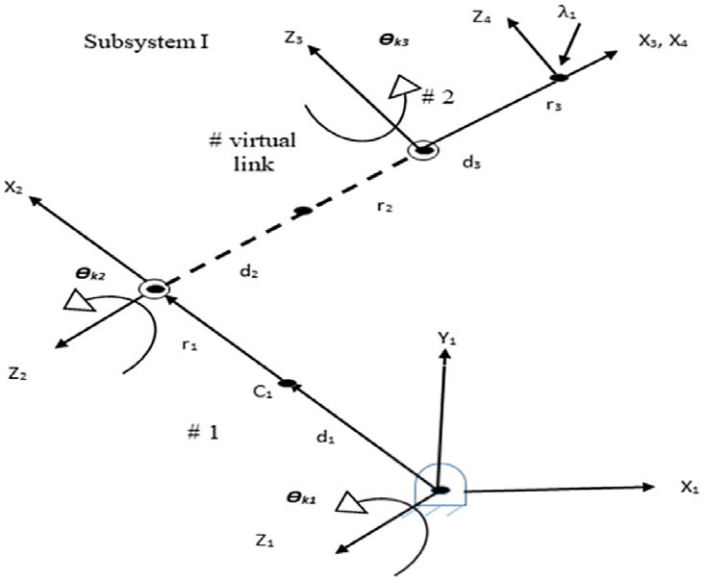


Figure 8. *Geometric architecture for subsystem I.*

3.3. Forward position kinematics solution [active joints]

The forward position kinematics is defined as the three actuated joint angles $\theta = \{\theta_1, \theta_2, \theta_3\}^{T}$ are given then compute the resulting Cartesian position of the unfixed surface control point (P), ${}^{B}_{P}R = \{x\ y\ z\}^{T}$. The forward position kinematics solution for parallel robots is tough.

The correct solution set is selected for a straightforward analytical solution, as there is translation-only motion in the 3DOF delta robot. Given $\Theta = \{\theta_1\theta_2\theta_3\}^{T}$, then compute the three absolute vector knee points using ${}^{B}_{i}A = {}^{B}_{i}B + {}^{B}_{i}L, i = 1,2,3$. The unfixed surface of the delta robot's orientation is constant and horizontal with $[{}^{B}_{i}A] = [I_3]$; therefore, all the virtual sphere centres (three in number) are ${}^{B}_{iv}A = {}^{B}_{i}A - {}^{p}_{i}P, i = 1,2,3$.

$$ {}_{1v}{}^{B}\mathrm{A} = \begin{Bmatrix} 0 \\ -w_B - L\cos\theta_1 + u_p \\ -L\sin\theta_1 \end{Bmatrix} $$

$$ {}_{2v}{}^{B}\mathrm{A} = \begin{Bmatrix} \frac{\sqrt{3}}{2}(w_B + L\cos\theta_2) - \frac{s_p}{2} \\ \frac{1}{2}(w_B + L\cos\theta_2) - w_p \\ -L\sin\theta_2 \end{Bmatrix} $$

$$ {}_{3v}{}^{B}\mathrm{A} = \begin{Bmatrix} -\frac{\sqrt{3}}{2}(w_B + L\cos\theta_3) + \frac{s_p}{2} \\ \frac{1}{2}(w_B + L\cos\theta_3) - w_p \\ -L\sin\theta_3 \end{Bmatrix} $$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

Delta robot forward position kinematics is determined by its point of intersection of the three spheres. This sphere is represented by a centre point called c and the scalar radius defined by r.

In this section, an analytical solution is computed as the intersection point of the given three spheres. In this concern, it also needs to solve transcendental equations which is of coupled in nature. So, to solve the forward kinematics solution, an intersection-three-sphere algorithm exists. If this is the case when the entire three-sphere centres $\{ {}^{B}_{iv}A \}$ have the same height (Z), then the solution will be in the zone of the algorithmic singularity. To fix the problem rotating the coordinates, i.e., all $\{ {}^{B}_{iv}A \}$, heights values are not the same.

As in this problem, two solutions were encountered when the three spheres intersect. When a sphere meeting is tangential, it results in one solution. The second case is an imaginary solution, and the data are inconsistent. The algorithm (spheres-intersection) compute solution sets, and the computer automatically chooses the solution below the base triangle. This approach to the forward position kinematics for the delta robot estimated results same as of solving the kinematics equations, depicted as ${}^{B}_{P}R = \{\text{x y z}\}^{\text{T}}$ given $\theta = \{\theta_{1,}\, \theta_2, \theta_3\}^{\text{T}}$.

3.4. Inverse position kinematics solution [active joints]

The inverse position kinematics problem is defined as the Cartesian position of the unfixed surface control point (P), ${}^{B}_{P}R = \{\text{x y z}\}\text{T}$ are given then compute the three actuated revolute joint angles $\theta = \{\theta 1, \theta 2, \theta 3\}$ T. The inverse position kinematics solution delta robot is not trivial, therefore calculated analytically. The inverse position kinematics problem is not dependent on the three RUU legs of a delta robot; therefore, it can be solved individually for each leg. As shown in the figure above, geometrically, inverse position kinematics solution of each leg, the point of intersection between the known sphere (defined by radius l, and it is centre on the moving triangle vertex ${}^{P}_{i}P$) and known circle (defined by radius L, and it is centre on the fixed triangle R joint point ${}^{B}_{i}B$).

The inverse position kinematics solved both geometrically and trigonometrically. The inverse position kinematics are now solved analytically. The three independent scalar inverse position kinematics equations are shown below:

$$\text{E}_\text{i}\cos\theta_\text{i} + \text{F}_\text{i}\sin\theta_\text{i} + \text{G}_\text{i} = 0, i = 1,2,3$$

where

$$\text{E}_1 = 2L\,(y + a)\;\text{F}_1 = 2zL \tag{8}$$

$$\text{G}_1 = x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 \tag{9}$$

$$\text{E}_2 = -L\left(\sqrt{3}\,(x + b) + y + c\right)\text{F}_2 = 2zL \tag{10}$$

$$\text{G}_2 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 \tag{11}$$

$$\text{E}_3 = L\left(\sqrt{3}\,(x - b) - y - c\right), \text{F}_3 = 2zL \tag{12}$$

$$\text{G}_3 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 \tag{13}$$

The equation $\text{E}_\text{i}\cos\theta_\text{i} + \text{F}_\text{i}\sin\theta_\text{i} + \text{G}_\text{i} = 0$ already used in robot, kinematics. A tangent half-angle solves it. If $\text{t}_\text{i} = tan\frac{\theta_i}{2}$, then $\cos\theta_\text{i} = \frac{1-t_i^2}{1+t_i^2}$ and $\sin\theta_\text{i} = \frac{2t_i}{1+t_i^2}$. If tangent half-angle substituted into the EFG equation,

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

then results below:

$$\mathrm{E_i}\left(\frac{1-t_i^2}{1+t_i^2}\right)+\mathrm{F_i}\left(\frac{2t_i}{1+t_i^2}\right)+\mathrm{G_i}=0\ \mathrm{E_i}\left(1-t_i^2\right)+\mathrm{F_i}\left(2t_i\right)+\mathrm{G_i}(1+t_i^2)=0(1.37)$$

$$(\mathrm{G_i}-\mathrm{E_i})t_i^2+(2\mathrm{F_i})t_i+(\mathrm{G_i}+\mathrm{E_i})=0\ldots. \tag{14}$$

Quadratic formula: $t_{i_{1,2}}=\frac{-\mathrm{F_i}\pm\sqrt{E_i^2+F_i^2-G_i^2}}{\mathrm{G_i}-\mathrm{E_i}}$. The θ_i solved by inverting the original tangent, as shown below:

$$\theta_\mathrm{i}=2tan^{-1}\left(\mathrm{t_i}\right)$$

Two solutions (positive and negative) are obtained from the quadratic formula. The first solution represents the knee left, and the second is the knee right of a delta robot. There are two possible solutions for each leg. Therefore, the eight possible solutions are there.

3.5. Delta robot velocity kinematics equations [active joints]

The velocity kinematics equations of the revolute-input delta robot are computed through the derivative of the position constraint equations.

$$2\mathrm{L}\dot{\mathrm{y}}\cos\theta_1-2\mathrm{L}\left(\mathrm{y}+\mathrm{a}\right)\dot{\theta}_1\sin\theta_1+2\mathrm{L}\dot{\mathrm{z}}\sin\theta_1+2\mathrm{Lz}\dot{\theta}_1\cos\theta_1+2\mathrm{x}\dot{\mathrm{x}}+2\left(\mathrm{y}+\mathrm{a}\right)\dot{\mathrm{y}}+2\mathrm{z}\dot{\mathrm{z}}=0 \tag{15}$$

$$-\mathrm{L}\left(\sqrt{3}\dot{\mathrm{x}}+\dot{\mathrm{y}}\right)\cos\theta_2+\mathrm{L}\left(\sqrt{3(}\mathrm{x}+\mathrm{b}\right)+\mathrm{y}+\mathrm{c})\dot{\theta}_2\cos\theta_2+2\mathrm{L}\dot{\mathrm{z}}\sin\theta_2+2\mathrm{Lz}\dot{\theta}_2\cos\theta_2+2(\mathrm{x}+\mathrm{b})\dot{\mathrm{x}}$$
$$+2(\mathrm{y}+\mathrm{c})\dot{\mathrm{y}}+2\mathrm{z}\dot{\mathrm{z}}=0 \tag{16}$$

$$\mathrm{L}\left(\sqrt{3}\dot{\mathrm{x}}-\dot{\mathrm{y}}\right)\cos\theta_3-\mathrm{L}\left(\sqrt{3}\left(\mathrm{x}-\mathrm{b}\right)-\mathrm{y}-\mathrm{c}\right)\dot{\theta}_3\sin\theta_3+2\mathrm{L}\dot{\mathrm{z}}\dot{\theta}_3\sin\theta_3+2\mathrm{Lz}\cos\theta_3-2\left(\mathrm{x}-\mathrm{b}\right)\dot{\mathrm{x}}$$
$$+2\left(\mathrm{y}+\mathrm{c}\right)\dot{\mathrm{y}}+2\mathrm{z}\dot{\mathrm{z}}=0 \tag{17}$$

Equation 15, 16, and 17 can be re-written as

$$\mathrm{x}\dot{\mathrm{x}}+\left(\mathrm{y}+\mathrm{a}\right)\dot{\mathrm{y}}+\mathrm{L}\dot{\mathrm{y}}\cos\theta_1+\mathrm{z}\dot{\mathrm{z}}+\mathrm{L}\dot{\mathrm{z}}\sin\theta_1=\mathrm{L}\left(\mathrm{y}+\mathrm{a}\right)\dot{\theta}_1\sin\theta_1-\mathrm{Lz}\dot{\theta}_1\cos\theta_1 \tag{18}$$

$$2\left(\mathrm{x}+\mathrm{b}\right)\dot{\mathrm{x}}+2\left(\mathrm{y}+\mathrm{c}\right)\dot{\mathrm{y}}-\mathrm{L}\left(\sqrt{3}\dot{\mathrm{x}}+\dot{\mathrm{y}}\right)\cos\theta_2+2\mathrm{z}\dot{\mathrm{z}}+2\mathrm{L}\dot{\mathrm{z}}\sin\theta_2$$
$$=-\mathrm{L}\left(\sqrt{3}\left(\mathrm{x}+\mathrm{b}\right)+\mathrm{y}+\mathrm{c}\right)\dot{\theta}_2\sin\theta_2-2\mathrm{Lz}\dot{\theta}_2\cos\theta_2 \tag{19}$$

$$2\left(\mathrm{x}-\mathrm{b}\right)\dot{\mathrm{x}}+2\left(\mathrm{y}+\mathrm{c}\right)\dot{\mathrm{y}}+\mathrm{L}\left(\sqrt{3}\dot{\mathrm{x}}-\dot{\mathrm{y}}\right)\cos\theta_3+2\mathrm{z}\dot{\mathrm{z}}+2\mathrm{L}\dot{\mathrm{z}}\sin\theta_3$$
$$=-\mathrm{L}\left(\sqrt{3}\left(\mathrm{x}-\mathrm{b}\right)-\mathrm{y}-\mathrm{c}\right)\dot{\theta}_3\sin\theta_3-2\mathrm{Lz}\dot{\theta}_3\cos\theta_3 \tag{20}$$

In matrix-vector form: $[\mathrm{A}]\{\mathrm{X}\}=[\dot{\mathrm{B}}]\{\dot{\theta}\}$

$$\begin{bmatrix}\mathrm{x} & \mathrm{y}+\mathrm{a}+\mathrm{L}\cos\theta_1 & \mathrm{z}+\mathrm{L}\sin\theta_1\\ 2\left(\mathrm{x}+\mathrm{b}\right)-\sqrt{3}\mathrm{L}\cos\theta_2 & 2\left(\mathrm{y}+\mathrm{c}\right)-\mathrm{L}\cos\theta_2 & 2(\mathrm{z}+\mathrm{L}\sin\theta_2)\\ 2\left(\mathrm{x}-\mathrm{b}\right)+\sqrt{3}\mathrm{L}\cos\theta_3 & 2\left(\mathrm{y}+\mathrm{c}\right)-\mathrm{L}\cos\theta_3 & 2(\mathrm{z}+\mathrm{L}\sin\theta_3)\end{bmatrix}\begin{Bmatrix}\dot{\mathrm{x}}\\ \dot{\mathrm{y}}\\ \dot{\mathrm{z}}\end{Bmatrix}=\begin{bmatrix}\mathrm{b}_{11} & 0 & 0\\ 0 & \mathrm{b}_{22} & 0\\ 0 & 0 & \mathrm{b}_{33}\end{bmatrix}\begin{Bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{Bmatrix} \tag{21}$$

where $\mathrm{b}_{11}=\mathrm{L}[(\mathrm{y}+\mathrm{a})\sin\theta_1-\mathrm{z}\cos\theta_1];\ \mathrm{b}_{22}=-\mathrm{L}[(\sqrt{3}(\mathrm{x}+\mathrm{b})+\mathrm{y}+\mathrm{c})\sin\theta_2+2\mathrm{z}\cos\theta_2]\ \mathrm{b}_{33}=\mathrm{L}[(\sqrt{3}(\mathrm{x}-\mathrm{b})-\mathrm{y}-\mathrm{c})\sin\theta_3-2\mathrm{z}\cos\theta_3]$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

3.6. Inverse position kinematics solution [active and passive joints]

In kinematic and dynamic analyses, the moving platform is assumed to be cut at three spherical joints and detached from the three identical legs. As shown in Fig. 6(a), the three-DOF delta platform has a fixed base connected to three identical legs through universal joints at three different points, namely, at Ai's. These legs are connected to the platform (moving) through three spherical joints at Bi's. Each leg consists of two links connected with a universal joint denoted as Ui's. Therefore, each leg has three-DOF. Referring to Fig. 6(b), the loop-closure equation for the kth loop is given as

$$\overrightarrow{B_i} + \overrightarrow{L_i} + \overrightarrow{l_i} = \overrightarrow{P} + p_i \tag{22}$$

$$\overrightarrow{L_i} + \overrightarrow{l_i} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ -\mathrm{u_p} \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ -\mathrm{w_B} \\ 0 \end{bmatrix} \tag{23}$$

$\overrightarrow{\mathrm{L_i}} + \overrightarrow{\mathrm{l_i}}$ can be found using the D-H parameter approach. $\overrightarrow{\mathrm{L_i}}$ is the vector from the base vertex and corresponding platform vertex, i.e., from Ai to Bi. The magnitude of this vector also represents leg length for the respective platform motion. Vector p is the measure of the displacement of the platform to the base. Here 'p' represents the position vector of point 'P' on the platform from the centre of the base, i.e., points O, $\overrightarrow{\mathrm{L_i}}$ *and* $\overrightarrow{\mathrm{l_i}}$ are the vectors locating the vertices of the base and platform in their respective frames, i.e., XYZ and Xp, Yp, Zp as depicted in Fig.6(a). Since the platform has three-DOF, its motion can be represented by three independent translations and about axes Xp, Yp, and Zp. If the values of xp, yp, zp, the vector $\mathrm{a_k}$ can be calculated from Eq. 2. The coordinate values of the base and platform vertices, i.e., $\mathrm{A_I}$, $\mathrm{A_{III}}$, and $\mathrm{B_I}$. . ., $\mathrm{B_{IX}}$ are taken from the CAD model to calculate the active and passive joint trajectories. The homogeneous transformation matrix between frame 3 and frame 1 can be given as the homogeneous transformation matrix (HTM).

$$T_k \equiv \begin{bmatrix} C\theta_{ki} & -S\theta_{ki}C\alpha_{ki} & -S\theta_{ki}S\alpha_{ki} & a_iC\theta_{ki} \\ S\theta_{ki} & C\theta_{ki}C\alpha_{ki} & -C\theta_{ki}S\alpha_{ki} & a_{ki}S\theta_{ki} \\ 0 & S\alpha_{ki} & C\alpha_{ki} & b_{ki} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

$$\text{Consider, } a_2 = 0;\ T_1 = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & a_1c\theta_1 \\ s\theta_1 & c\theta_1 & 0 & a_1s\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_2 = \begin{bmatrix} c\theta_2 & 0 & -s\theta_2 & a_2c\theta_2 \\ s\theta_2 & 0 & c\theta_2 & a_2s\theta_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_1T_2T_3 = \begin{bmatrix} c\theta_{12}c\theta_3 & -c\theta_{12}s\theta_3 & -s\theta_{12} & a_1c\theta_1 + a_2c\theta_{12} \\ s\theta_{12}c\theta_3 & -s\theta_{12}s\theta_3 & c\theta_{12} & a_1s\theta_1 + a_2s\theta_{12} \\ -s\theta_3 & -c\theta_3 & 0 & -a_3s\theta_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using equation 24 and putting in the triangular frame, represented as

$$\overrightarrow{L_i} + \overrightarrow{l_i} = \begin{bmatrix} -a_3s\theta_3 \\ a_1c\theta_1 + a_2c\theta_{12} \\ a_1s\theta_1 + a_2s\theta_{12} \end{bmatrix}, \tag{25}$$

$$\overrightarrow{L_i} + \overrightarrow{l_i} = \begin{bmatrix} x \\ y1 \\ z \end{bmatrix} \tag{26}$$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

where $y1 = y - u_p + w_B$

$$-a_3 s\theta_3 = x$$

$$\theta_3 = -sin^{-1}(x/a_3)$$

Using equations 25 and 26, square and adding resulted in the computation of the θ_2

$$\theta_2 = -sin^{-1}(k)$$

where $k = (x^2 + y^2 + z^2 - a_1{}^2 - a_3{}^2/2a_1a_3c\theta_3)$

$$y1 = a_1 c\theta_1 + a_2 c\theta_{12} \tag{27}$$

$$z\, a_1 s\theta_1 + a_2 s\theta_{12}. \tag{28}$$

$$y1^* (a_3 s\theta_2 c\theta_3) = a_1 c\theta_1 + a_2 c\theta_{12}{}^* a_3 s\theta_2 c\theta_3) \tag{29}$$

$$z^* (a_1 + a_3 c\theta_2 c\theta_3)\, a_1 s\theta_1 + a_2 s\theta_{12}{}^* (a_1 + a_3 c\theta_2 c\theta_3) \tag{30}$$

Multiplying $a_3 s\theta_2 c\theta_3$ in equations 27 and $a_1 + a_3 c\theta_2 c\theta_3$ in equation 28 and termed the equations 29 and 30, respectively. Using equations 29 and 30 and subtracting equation 27 from equation 28 resulted in the computation of theθ_1.

$$\theta_1 = tan^{-1}(q)$$

$$\text{where} q = (z(a_1 + a_3 c\theta_2 c\theta_3) - y1(a_3 s\theta_2 c\theta_3)/z(a_1 + a_3 c\theta_2 c\theta_3) + y1(a_3 s\theta_2 c\theta_3)) \tag{31}$$

Kinematic loop closure equation inherited from the equations 25 and 26, represented as

$$\begin{bmatrix} -a_3 s\theta_3 \\ a_1 c\theta_1 + a_2 c\theta_{12} \\ a_1 s\theta_1 + a_2 s\theta_{12} \end{bmatrix} = \begin{bmatrix} x \\ y1 \\ z \end{bmatrix} \tag{32}$$

Differentiate equation 32 with respect to the time

$$\begin{aligned} -a_3 c\theta_3 \dot{\theta_3} &= \dot{x} \\ a_3 \left(-c\theta_3 s\theta_{12} \dot{\theta_{12}} - s\theta_3 c\theta_{12} \dot{\theta_3}\right) - a_1 s\theta_1 \dot{\theta_1} &= \dot{y1} \\ a_3 \left(c\theta_3 c\theta_{12} \dot{\theta_{12}} - s\theta_3 s\theta_{12} \dot{\theta_3}\right) + a_1 c\theta_1 \dot{\theta_1} &= \dot{z} \end{aligned}$$

$[J\dot{\theta} = \dot{\vec{X}}]$ represented as

$$\begin{bmatrix} 0 & -a_3 c\theta_3 \dot{\theta_3} & 0 \\ -a_1 s\theta_1 - a_3 c\theta_3 s\theta_{12} & -a_3 c\theta_3 s\theta_{12} & -a_3 s\theta_3 c\theta_{12} \\ a_1 c\theta_1 + a_3 c\theta_3 c\theta_{12} & a_3 c\theta_3 c\theta_{12} - a_3 s\theta_3 s\theta_{12} & \end{bmatrix} \begin{bmatrix} \dot{\theta_1} \\ \dot{\theta_2} \\ \dot{\theta_3} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \ldots.$$

$$[J] \begin{bmatrix} \dot{\theta_1} \\ \dot{\theta_2} \\ \dot{\theta_3} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{33}$$

Again, for acceleration level, differentiate equation 1.57 to the $\dot{\theta}$ and represented as

$$J\ddot{\theta} + \dot{J}\dot{\theta} = \ddot{\vec{X}} \tag{34}$$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

where $\dot{\theta} = \left[\dot{\theta}_{k1}\dot{\theta}_{k2}\dot{\theta}_{k3}\right]^{\mathrm{T}}$ and $\mathrm{J_k}$ is the 3×3 Jacobian matrix related to the serial chain of the $\mathrm{k^{th}}$ leg. The Jacobian matrix $\mathrm{J_k}$ can be obtained by differentiating the first three elements of the last column of the homogeneous transformation matrix $\mathrm{T_k}$.

$$J_k \equiv \begin{bmatrix} 0 & -a_3 c\theta_3 \dot{\theta}_3 & 0 \\ -a_1 s\theta_1 - a_3 c\theta_3 s\theta_{12} & -a_3 c\theta_3 s\theta_{12} & -a_3 s\theta_3 c\theta_{12} \\ a_1 c\theta_1 + a_3 c\theta_3 c\theta_{12} & a_3 c\theta_3 c\theta_{12} & -a_3 s\theta_3 s\theta_{12} \end{bmatrix} \tag{35}$$

Similarly differentiating Eq. 34, the joint acceleration can be derived as

$$\ddot{\theta} = J_k^{-1}(\ddot{l}_k - \dot{J}_k \dot{\theta}_k) \tag{36}$$

where $\ddot{\theta}_k \equiv \left[\ddot{\theta}_{k1}\ddot{\theta}_{k2}\ddot{p}_{k3}\right]^T$. The derivative of the Jacobian matrix, $\dot{J}_k \equiv \left[\dot{J}_{1k}\dot{J}_{2k}\dot{J}_{3k}\right]$ in Eq. 2 is as follows:

$$j_{1k} \equiv \begin{bmatrix} 0 \\ -a_1 c\theta_1 \dot{\theta}_1 - a_3(c\theta_3 c\theta_{12}\dot{\theta}_{12}^{\cdot} - s\theta_3 s\theta_{12}\dot{\theta}_3) \\ -a_1 s\theta_1 \dot{\theta}_1 - a_3(c\theta_3 s\theta_{12}\dot{\theta}_{12}^{\cdot} - s\theta_3 c\theta_{12}\dot{\theta}_3) \end{bmatrix} \tag{37}$$

$$j_{2k} \equiv \begin{bmatrix} 0 \\ -a_3(c\theta_3 c\theta_{12}\dot{\theta}_{12}^{\cdot} - s\theta_3 s\theta_{12}\dot{\theta}_3) \\ -a_3(c\theta_3 s\theta_{12}\dot{\theta}_{12}^{\cdot} + c\theta_3 c\theta_{12}\dot{\theta}_3) \end{bmatrix} \tag{38}$$

$$j_{3k} \equiv \begin{bmatrix} a_3 s\theta_3 \dot{\theta}_3 \\ -a_3(-s\theta_3 s\theta_{12}\dot{\theta}_{12}^{\cdot} + c\theta_3 c\theta_{12}\dot{\theta}_3) \\ -a_3(s\theta_3 c\theta_{12}\dot{\theta}_{12}^{\cdot} + c\theta_3 s\theta_{12}\dot{\theta}_3) \end{bmatrix} \tag{39}$$

4. Dynamics Analysis

The mechanism model of the parallel delta robot is shown in Fig. 9(a), which has a fixed base and a moving platform. Each leg has two links, the first link is connected to the base and the second link by revolute joint and spherical joint, respectively, while the second link is connected to the platform by spherical joint. The base joints (revolute) are active. To simplify the dynamics of the delta robot, it is divided into four subsystems, as shown in Fig.10(b), and to compensate, Lagrange multipliers are used. The subsystems one, two, and three can be considered two-link manipulators with three-degree-of-freedom (since the rotation of the second link about its axis is not considered), in the effect of Lagrange multipliers. Since the legs of the delta robot are identical, we only need to derive the dynamic equation of one leg and the moving platform. Combining the above-said dynamics resulted in the dynamics of the delta robots for performing the catching.

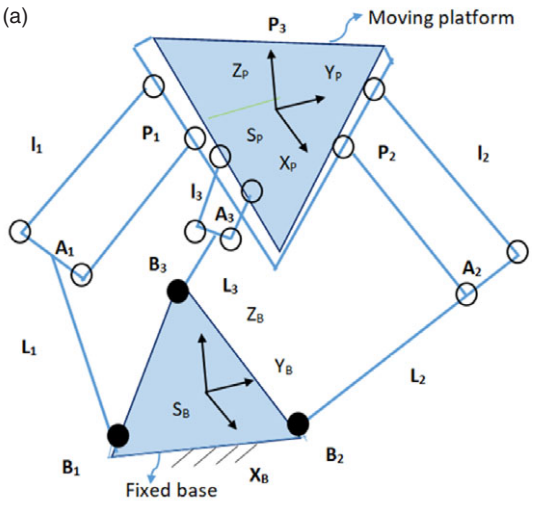
4.1. Dynamics of the legs

The dynamic analysis of the delta robot is performed using the DeNOC formulation [give reference]. Subsystem one, i.e., leg 1, has two links but three-DOF; therefore, to derive the dynamic equations, a virtual link was introduced to make the number of links equal to the degree of freedom of the leg. The unconstraint Newton–Euler equations for the i^{th} link of the k^{th} leg can be expressed as

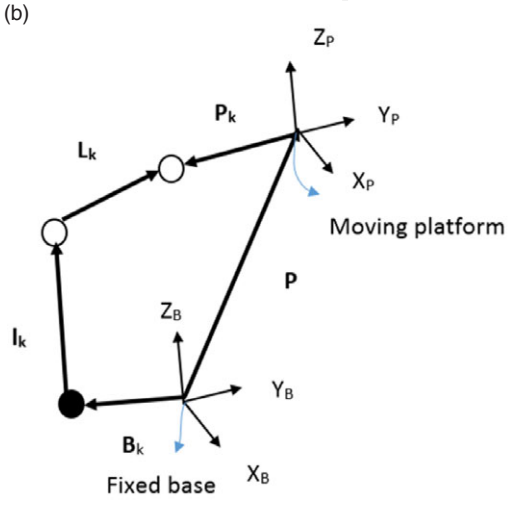
$$\begin{aligned} I_{ik}\omega_{ik} + \omega_{ik} \times I_{ik}\omega_{ik} &= n_{ik} \\ m_{ik}\ddot{c}_{ik} &= f_{ik} \end{aligned} \tag{40}$$

where I_{ik} and m_{ik} are the inertia tensor and mass of the link, respectively, ω_{ik} and $\ddot{c}_{ik}$ are the angular velocity and linear velocity of centre-of-mass (COM), respectively, while ω_{ik} and $\ddot{c}_{ik}$ are their corresponding

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



Skeletal model of delta platform



A kinematic loop

Figure 9 *(a) Skeletal model of delta platform. (b) A kinematic loop.*

time derivative, n_{ik} and f_{ik} are the moment and force about/at the COM of the link, respectively. The above expression can be written in matrix-vector as given below

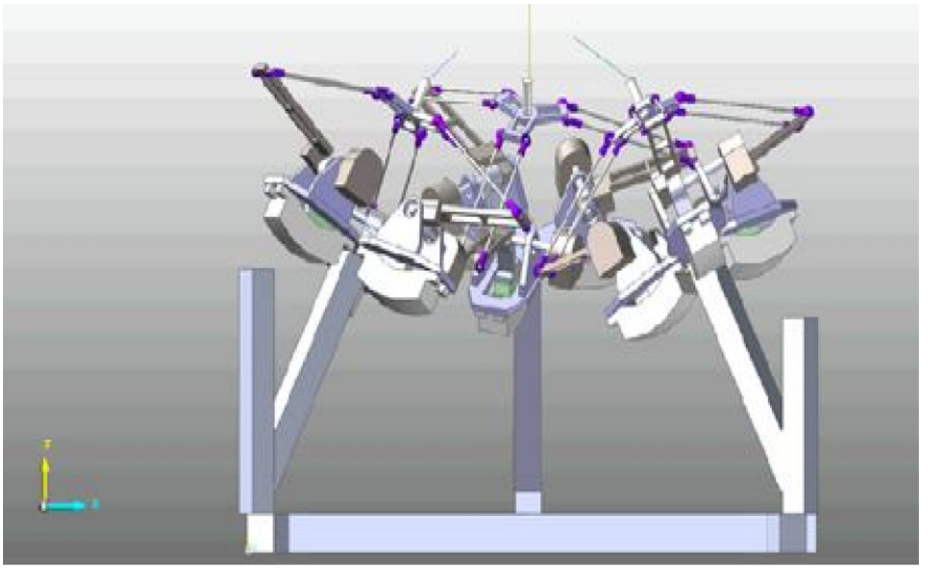
$$M_{ik}\dot{t}_{ik} + W_{ik}M_{ik}t_{ik} = \omega_{ik} \tag{41}$$

where $M_{ik} = \begin{bmatrix} I_{ik}O_3 \\ O_3 m_{ik} I_3 \end{bmatrix}$ is the mass matrix, $t_{ik} = \begin{bmatrix} \omega_{ik} \\ \dot{c}_{ik} \end{bmatrix}$ is the twist vector, $W_{ik} = \begin{bmatrix} \omega_{ik} I_3 O_3 \\ O_3 O_3 \end{bmatrix}$ is the angular velocity matrix, $\omega_{ik} = \begin{bmatrix} n_{ik} \\ f_{ik} \end{bmatrix}$ is the wrench vector, and 1_3 and O_3 are 3x3 identity and null matrix, respectively. The unconstraint equations for the k^{th} leg can be written as

$$M_{ik}\dot{t}_{ik} + W_{ik}M_{ik}t_{ik} = \omega_{ik} \tag{42}$$

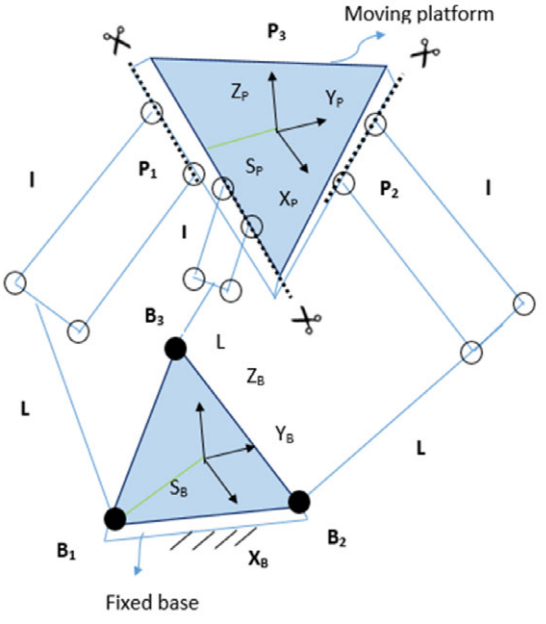
https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

(a)



RecurDyn^{TM} model of the experimental setup

(b)



Cut strategy

Figure 10 *(a) RecurDyn^{TM} model of the experimental setup. (b) Cut strategy.*

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

where $M_k = diag\left[\, M_{1k}\, M_{2k} M_{3k} \,\right]$

$$W_k = diag\left[\, W_{1k}\, W_{2k} W_{3k} \,\right]$$

$$t_k = \left[\, t_{1k}\, t_{2k} t_{3k} \,\right]^T$$

$$w_k = \left[\, w_{1k}\, w_{2k} w_{3k} \,\right]^T$$

4.1.1. Derivation of DeNOC matrices through kinematic constraints

The angular velocity and linear velocity of COM of the links in k^{th} can be written in recursive as given below

$$\omega_{1k} = \dot{\theta}_{1k} e_{1k}$$

$$\dot{c}_{1k} = \omega_{1k} \times d_{1k} = \dot{\theta}_{1k} e_{1k} \times d_{1k} \tag{43}$$

$$\omega_{2k} = \omega_{1k} + \dot{\theta}_{2k} e_{2k}$$

$$\dot{c}_{2k} = \dot{c}_{1k} + \omega_{1k} \times (r_{1k} + d_{2k}) + \dot{\theta}_{2k} e_{2k} \times d_{2k} \tag{44}$$

$$\omega_{3k} = \omega_{3k} + \dot{\theta}_{3k} e_{3k}$$

$$\dot{c}_{3k} = \dot{c}_{2k} + \omega_{2k} \times (r_{2k} + d_{3k}) + \dot{\theta}_{3k} e_{3k} \times d_{3k} \tag{45}$$

The corresponding twist vector for each link of the k^{th} leg can be expressed as

$$t_{1k} = p_{1k}\dot{\theta}_{1k};\; t_{2k} = B_{2,1k} t_{1k} + p_{2k}\dot{\theta}_{2k};\; t_{3k} = B_{3,2k} t_{2k} + p_{2k}\dot{\theta}_{3k} \tag{46}$$

where $P_{ik} = \begin{bmatrix} e_{ik} \\ e_{ik} \times d_{ik} \end{bmatrix}; i = 1,2,3$

$$B_{i,i-1k} = \begin{bmatrix} 1_3 & O_3 \\ c_{i,i-1k} \times 1_3 & 1_3 \end{bmatrix}$$

$$c_{i,i-1k} = -\left(r_{i-1k} + d_{ik}\right); i = 2,3.$$

The above twist vector can be written as $t_k = N_k \dot{\theta}_k$ with $N_k = N_{lk} N_{dk}$
where

$$N_{lk} \equiv \begin{bmatrix} 1_6 & O_6 & O_6 \\ B_{2,1k} & 1_6 & O_6 \\ B_{3,1k} & B_{3,2k} & 1_6 \end{bmatrix}; N_{dk} \equiv \begin{bmatrix} P_{1k} & \phi & \phi \\ \phi & P_{2k} & \phi \\ \phi & \phi & P_{3k} \end{bmatrix}; \dot{\theta}_k \equiv \begin{bmatrix} \dot{\theta}_{1k} \\ \dot{\theta}_{2k} \\ \dot{\theta}_{3k} \end{bmatrix}$$

1_6 and O_6 are the 6×6 identity and null matrix, respectively, while ϕ is the 6×1 zero vector. The 6×6 matrix $B_{3,1k}$ is the matrix multiplication of $B_{3,2k} B_{2,1k}$. The above N_k matrix is known as the Natural Orthogonal Complement (NOC) matrix, and its decoupled form $N_{lk} N_{dk}$ is known as Decoupled NOC, which would be used to project the unconstraint equations of motion on the kinematic constraint space as given below:

$$N_{dk}^T N_{lk}^T \left\{ M_k \dot{t}_k + W_k M_k t_k \right\} = N_{dk}^T N_{lk}^T w_k \tag{47}$$

$$I_k\left(\theta_k\right) \ddot{\theta}_k + C_k\left(\theta_k, \dot{\theta}_k\right) \dot{\theta}_k = -\gamma_k + \tau_k \tag{48}$$

where $I_k\left(\theta_k\right) \equiv N_k^T M_k N_k$ is the generalized inertia matrix (GIM); $C_k\left(\theta_k, \dot{\theta}_k\right) \equiv N_k^T \left[M_k \dot{N}_k + W_k M_k N_k \right]$ is the matrix of Coriolis forces; and $-\gamma_k + \tau_k \equiv N_k^T w_k$ in which γ_k the generalized is the force due to

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

gravity and τ_k represents the actuator torques. The above equations of motion considering the Lagrange multipliers λ can be expressed as

$$I_k\left(\theta_k\right)\ddot{\theta}_k + C_k\left(\theta_k, \dot{\theta}_k\right)\ddot{\theta}_k + \gamma_k = J_{k\,k}^T + \tau_k \tag{49}$$

where J_k is the Jacobian matrix and $k = 1, 2, 3$.

4.2. Dynamics of the platform

The platform is shown in Fig. 11. The local frame attached to the centroid 0_p of platform is$X_p - Y_p - Z_p$. The platform can only translate in 3D space, and therefore, the rotation matrix between the base frame and the platform frame is the identity matrix. The platform is connected to the legs at points p_1, p_2, and p_3. The position vector of the moving platform with respect to the base frame is given below

$$p = \left[xyz\right]^T \tag{50}$$

The dynamics of the platform using Newton's equation of motion can be expressed as

$$m_p\ddot{p} + \gamma_p = \lambda_1 + \lambda_2 + \lambda_3 \tag{51}$$

where m_p is the mass of the platform, $\ddot{p}$ is the acceleration of the platform, γ_p is the gravity force acting on the platform, and $\lambda_1, \lambda_2, \lambda_3$ representing the Lagrange's multipliers corresponding to the three-legs.

4.3. Dynamics of the delta robot

The dynamics of the delta robot can be expressed by combining the dynamics of the legs and the platform. The dynamics of the delta robot using Lagrange's multipliers and Jacobian matrices corresponding to each leg can be expressed in matrix-vector form as

$$I\left(\theta\right)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\ddot{\theta} + \gamma = J^T\lambda + \tau \tag{52}$$

where $I\left(\theta\right) \equiv \begin{bmatrix} I_1 & 0_3 & 0_3 & 0_3 \\ 0_3 & I_2 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & m_p I_3 \end{bmatrix}$

$$C(\theta, \dot{\theta}) \equiv \begin{bmatrix} C_1 & 0_3 & 0_3 & 0_3 \\ 0_3 & C_2 & 0_3 & 0_3 \\ 0_3 & 0_3 & C_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix};\ \gamma \equiv \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_p \end{bmatrix}$$

$$J \equiv \begin{bmatrix} J_1 & 0_3 & 0_3 \\ 0_3 & J_1 & 0_3 \\ 0_3 & 0_3 & J_1 \\ I_3 & I_3 & I_3 \end{bmatrix};\ \lambda \equiv \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \text{ and } \tau \equiv \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

The dynamic analyses of the multibody system are required to know the trajectory of different coordinates for a given torque or for a required trajectory what should be the torque at a different degree of freedom [29]. The dynamic modelling of the manipulator (parallel) is essential for their control because they are preferred in an application where precise positioning is required. Combining the equations of motion yields equation 53 and is represented as

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

$$\begin{bmatrix} I_{1_{3x3}} & 0_{3x3} & 0 & 0 \\ 0 & I_2 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & M_{pf} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \\ \ddot{\theta}_5 \\ \ddot{\theta}_6 \\ \ddot{\theta}_7 \\ \ddot{\theta}_8 \\ \ddot{\theta}_9 \\ \ddot{x}_p \\ \ddot{y}_p \\ \ddot{z}_p \end{bmatrix} + \begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 \\ 0 & 0 & C_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_{pf} \end{bmatrix}$$

$$= \begin{bmatrix} J^T_{c1_{3x3}} & 0 & 0 \\ 0 & J^T_{c2_{3x3}} & 0 \\ 0 & 0 & J^T_{c3_{3x3}} \\ 1_{3x3} & 1_{3x3} & 1_{3x3} \end{bmatrix} \begin{bmatrix} \lambda_1 x \dots z \\ \lambda_2 x \dots z \\ \lambda_3 x \dots z \end{bmatrix}_{9x1} + \begin{bmatrix} \tau_1 \\ 0 \\ 0 \\ \tau_1 \\ 0 \\ 0 \\ \tau_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{53}$$

The RHS of equation 53 can also be re-written for inverse dynamics as

$$\begin{bmatrix} 1 & J_{11} & J_{21} & J_{31} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_{12} & J_{22} & J_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_{13} & J_{23} & J_{33} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & J_{11} & J_{21} & J_{31} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{12} & J_{22} & J_{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{13} & J_{23} & J_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & J_{11} & J_{21} & J_{31} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_{12} & J_{22} & J_{32} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & J_{13} & J_{23} & J_{33} \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \lambda_{1x} \\ \lambda_{1y} \\ \lambda_{1z} \\ \tau_2 \\ \lambda_{2x} \\ \lambda_{2y} \\ \lambda_{2z} \\ \tau_3 \\ \lambda_{3x} \\ \lambda_{3y} \\ \lambda_{3z} \end{bmatrix} \tag{54}$$

4.4. Numerical illustration

This section presents the implementation of the inverse dynamics methodology for the three – DOF delta robots proposed in section IV (B).

$$\mathbf{J}_{\tau k} \equiv \begin{bmatrix} J_k(1,3) \\ J_k(2,3) \\ J_k(3,3) \end{bmatrix}; \lambda_k \equiv \begin{bmatrix} \lambda_{xk} \\ \lambda_{yk} \\ \lambda_{zk} \end{bmatrix}$$

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

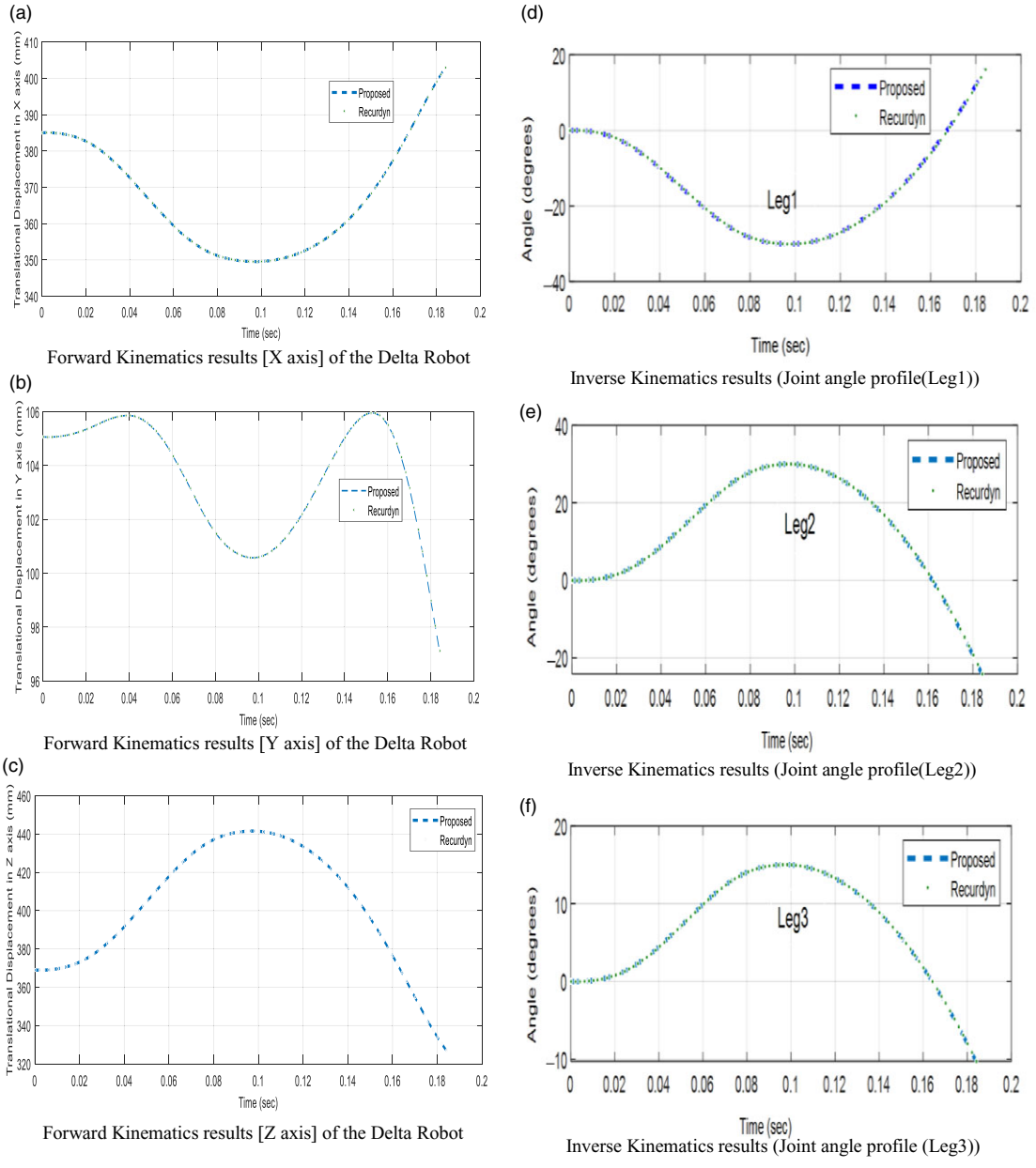


Figure 11 *(a) Forward kinematics results [X axis] of the delta robot. (b) Forward kinematics results [Y axis] of the delta robot. (c) Forward kinematics results [Z axis] of the delta robot. (d) Inverse kinematics results (Joint angle profile (Leg1)). (e) Inverse kinematics results (joint angle profile (Leg2)). (f) Inverse kinematics results (joint angle profile (Leg3)).*

Figure 10(a) shows the delta robot whose mass centre was offset to the geometric centre of the RecurDyn™ model for demonstration. Referring to Fig. 10(b), each cut joint is represented by three components of the Lagrange multipliers vectorλ, as the joints are spherical.

In a way, there are nine components of the Lagrange multipliers to be computed due to three cut joints. Using the methodology explained in section IV (D), forces at the legs were calculated as given next.

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

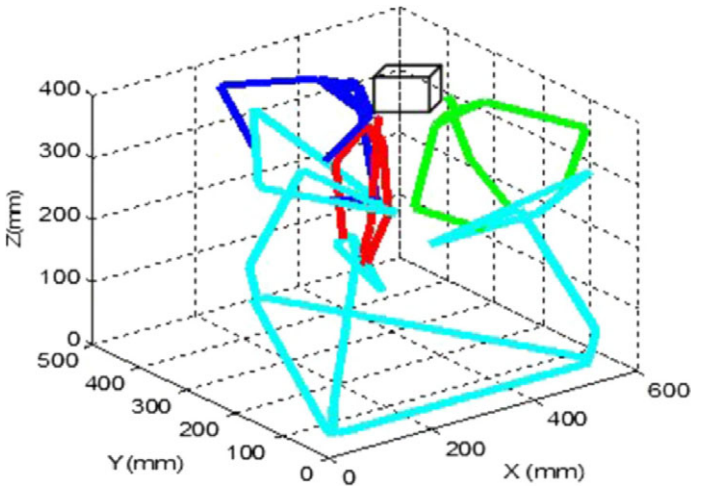


Figure 12. *Catching the free-fall cube in space.*

$$\phi_u^* \equiv \begin{bmatrix} \phi_{I1} \\ \phi_{I2} \\ \vdots \\ \phi_{VI1} \\ \phi_{VI2} \end{bmatrix}; \mathbf{J}_u \equiv \begin{bmatrix} \mathbf{J}_{Iu} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{J}_{IIu} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{J}_{VIu} \end{bmatrix} \tag{55}$$

where ϕ_u^* is the 12-dimensional vector related to unactuated joints of the legs, $\mathbf{J}_u$ is 12×18 matrix containing the elements of the Jacobian matrices associated with the unactuated joints of the legs, and $\mathbf{O}$ is the 2×3 matrix of zeros. The diagonal elements $\mathbf{J}_u$ have been represented using the k^{th} subsystem as

$$\mathbf{J}_{ku} \equiv \begin{bmatrix} J_k(1,1)J_k(2,1)J_k(3,1) \\ J_k(1,2)J_k(2,2)J_k(3,2) \end{bmatrix}$$

5. Catching

Once the intrinsic matrix [30,31] is computed for the fixed cameras, each runs in a thread. In the system, three threads run and give the pose of the cube to the fixed frame [32,33], as shown in Fig. 12. In each thread, communication with Matlab is performed synchronously.

Testing of the algorithm is performed by various trajectories such as free fall, vertical projection, and projection spatial and planar. The position estimation of the trajectories from one of the calibrated cameras is shown in section VII (F).

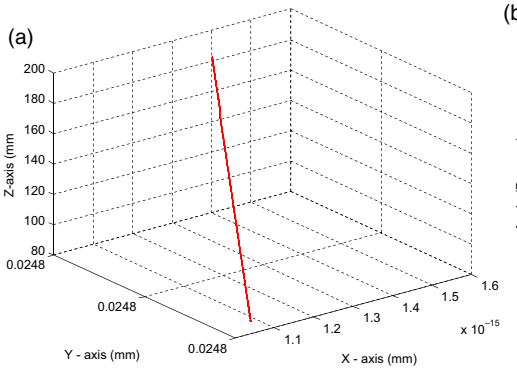
6. Results

6.1. Kinematics simulation results

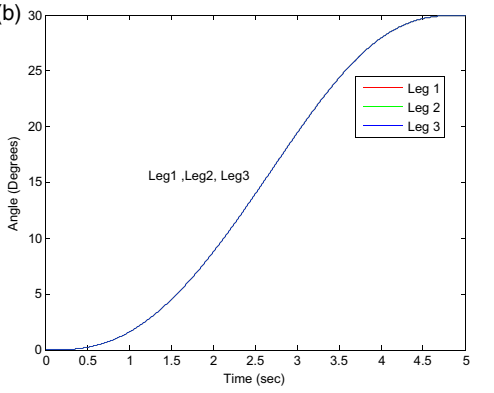
The 3D trajectories with respect to the fixed base frame of all three delta robots are shown in Fig. 13(a).

Given the position vectors to the fixed base frame and performing the inverse kinematics resulted in the joint angles of each of the Deltas as shown in Fig. 13(b)–(d). From Fig. 13(b)–(d), each leg of the delta robot starts from 0 degrees and reaches the final point at 30 degrees.

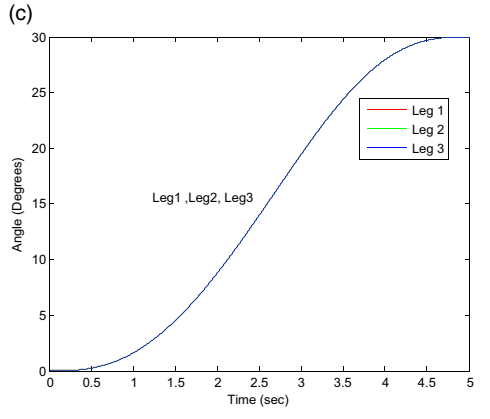
https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



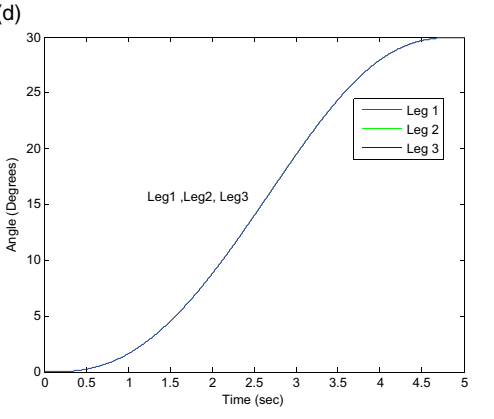
The 3D trajectory of Delta robot {1, 2, 3} with respect to Delta frame



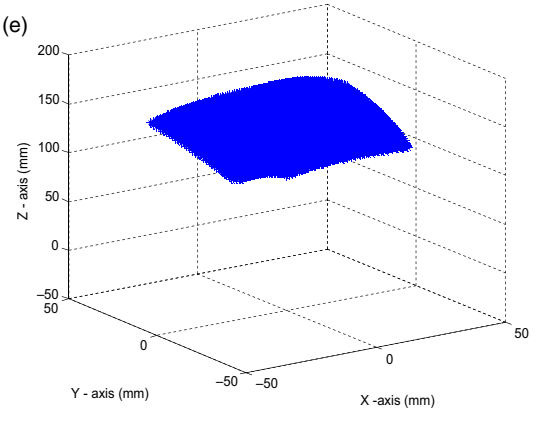
Joint Angles of Delta1 [Leg1, Leg2, Leg3]



Joint Angles of Delta2 [Leg1, Leg2, Leg3]



Joint Angles of Delta3 [Leg1, Leg2, Leg3]



The workspace of Delta1 [Leg1, Leg2, Leg3]

Figure 13 *(a) The 3D trajectory of delta robot {1, 2, 3} with respect to delta frame. (b) Joint angles of Delta1 [Leg1, Leg2, Leg3]. (c) Joint angles of Delta2 [Leg1, Leg2, Leg3]. (d) Joint angles of Delta3 [Leg1, Leg2, Leg3]. (e) The workspace of Delta1 [Leg1, Leg2, Leg3].*

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

Given the input, the joint angles and forward kinematics resulted in the position vectors to the fixed base frame of each delta, as shown in Fig. 13(e).

6.2. Kinematics results validated using RecurDyn™

The position and joint angles plots are given in Fig. 11(a)–(f), respectively. Note that to validate the results, the kinematic analyses were carried out using the commercial software RecurDyn™. The comparison in the above figures proves the correctness of the proposed methodology.

6.3. Dynamics analysis results

The dynamic analysis performed for the straight-line trajectory, i.e., initial $x_i = 1.0384\text{e-}15$, $y_i = 0.00002479$, $z_i = 0.0878350$; $P_1 = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$; and the final $x_f = 1.400\text{e-}15$, $y_f = 0.0000248$, $z_f = 0.1828063$; $P_2 = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix}$.

Case I. [From P1 to P2]

Figure 14(a)–(c) shows the position, velocity, and acceleration profile. The torque acting on the active joints (rotatory) is shown in Fig. 14(d). The torque acting on the passive joints (universal) are shown in Fig. 14(e)–(h).

Case II. [From P2 to P1]. The joint angle, position, velocity, and acceleration profile are shown in Fig. 15(a)–(d). The torque acting on the active joints (rotatory) is shown in Fig. 15(e).

Case III. Circular Trajectory. The joint angle, position, velocity, and acceleration profile are shown in Fig. 16(a)–(d). The torque acting on the active joints (rotatory) is shown in Fig. 16(e).

Case IV. Helical Trajectory. The joint angle, position, velocity, and acceleration profile are shown in Fig. 17(a)–(d). The torque acting on the active joints (rotatory) is shown in Fig. 17(e).

6.4. Real-time controlling of the delta manipulator

In this case, a planar circular path was input. The delta robot movement while following the path 02 times recorded by the encoders. The planar circular path is shown in Fig. 18(a).

In this case, a helical trajectory was given input. The delta robot movement following the path was recorded. As the helical path is provided to the delta robot, as shown in Fig. 19(a), to the inverse kinematics function and its corresponding theta angles estimated. The desired and measured end-effector (helical trajectory) position is shown in Fig. 19(b). The error in executing a circular profile is analysed. Desired angles in Leg1, Leg2 and Leg3 are shown in Fig. 19(c), and measured angles in Leg1, Leg2 and Leg3 are shown in Fig. 19(d).

I have hence tested the fundamental motions. Since the general motion of a rigid body in space is described by a screw (Chasal's Theorem), I have also tested helical motion. I have hence validated that the error between the vision-based measurement and kinematics based on link length and encoder angles is of the order of a degree.

The result section (VI), sub-section A (kinematics simulation results), sub-section B (kinematics results validated using RecurDyn™ and sub-section C (dynamics analysis results) presented the simulation-based implementation. The sub-section D (real-time controlling of the delta manipulator) presented the experimental results using the real robot(s) (delta robot).

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

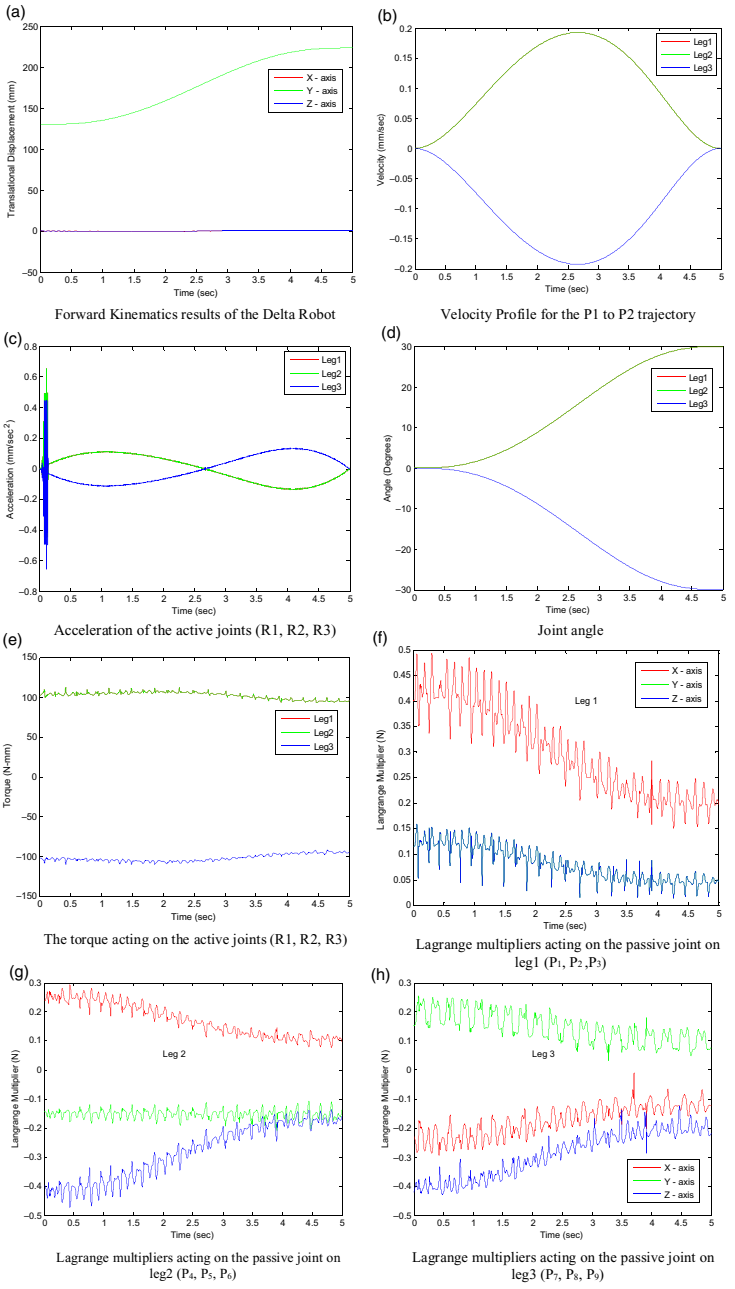
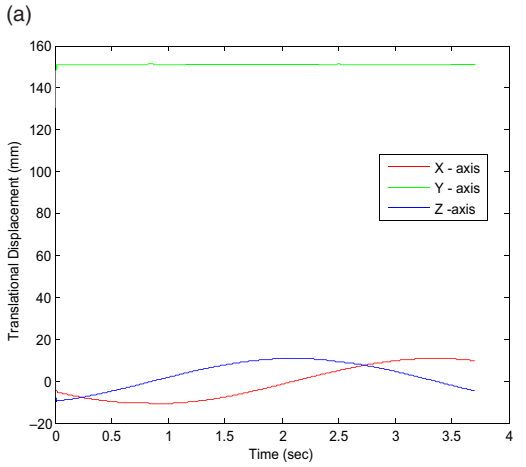
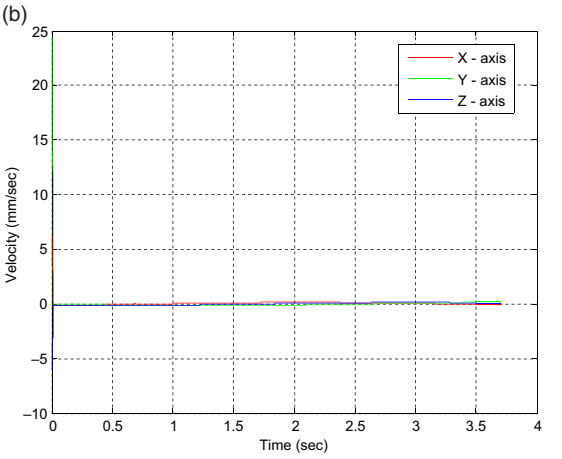


Figure 14 *(a) Forward kinematics results of the delta robot [P1 to P2]. (b) Velocity profile for the P1 to P2 trajectory. (c) Acceleration of the active joints (R1, R2, R3). (d) Joint angle. (e) The torque acting on the active joints (R1, R2, R3). (f) Lagrange multipliers acting on the passive joint on leg1 (P_1, P_2, P_3). (g) Lagrange multipliers acting on the passive joint on leg2 (P_4, P_5, P_6). (h) Lagrange multipliers acting on the passive joint on leg3 (P_7, P_8, P_9).*

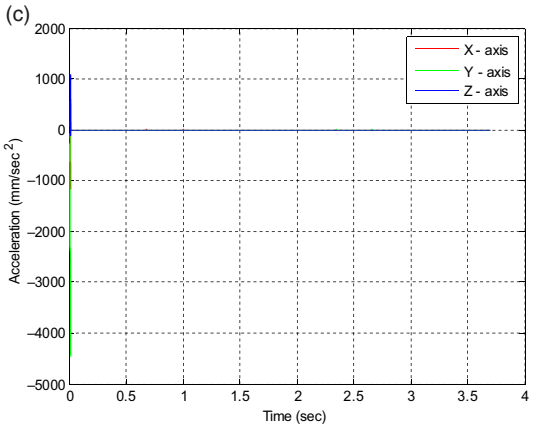
https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



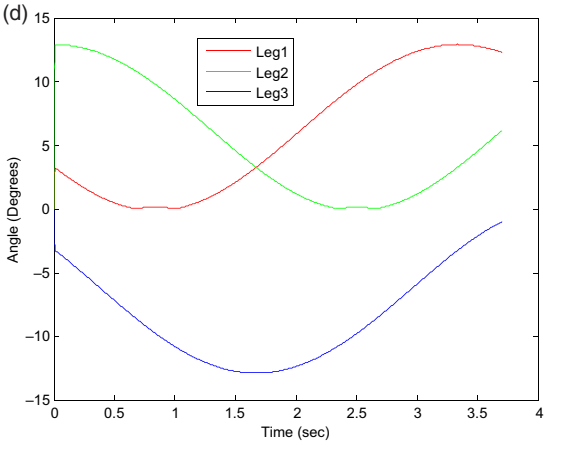
Forward Kinematics results of the Delta Robot



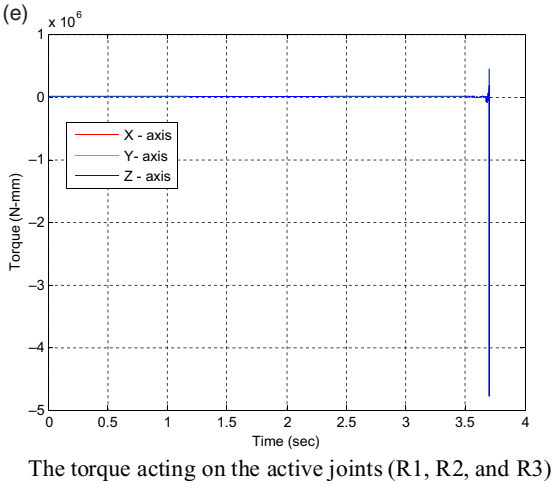
Velocity Profile for the P2 to P1 trajectory



Acceleration of the active joints (R1, R2, R3)



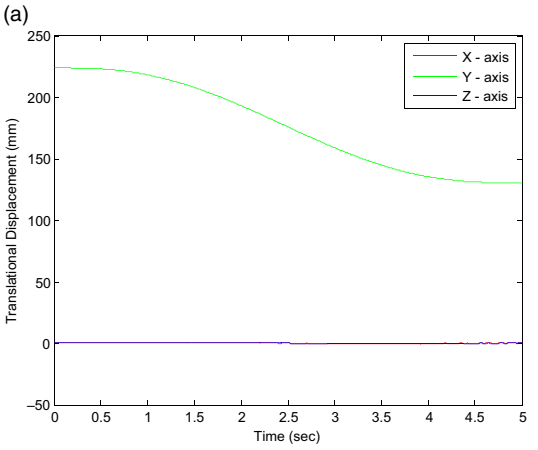
Joint angle



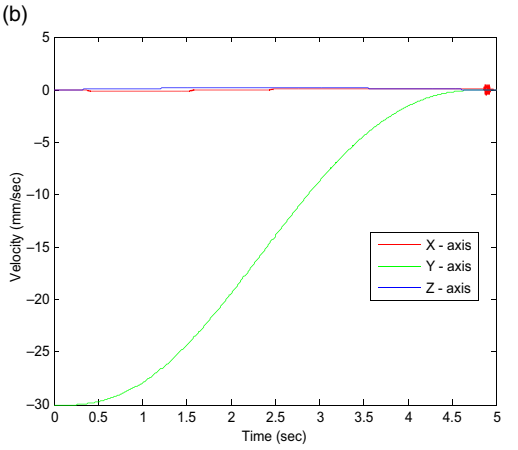
The torque acting on the active joints (R1, R2, and R3)

Figure 15 *(a) Forward kinematics results of the delta robot [P2 to P1]. (b) Velocity profile for the P2 to P1 trajectory. (c) Acceleration of the active joints (R1, R2, R3). (d) Joint angle. (e) The torque acting on the active joints (R1, R2, R3).*

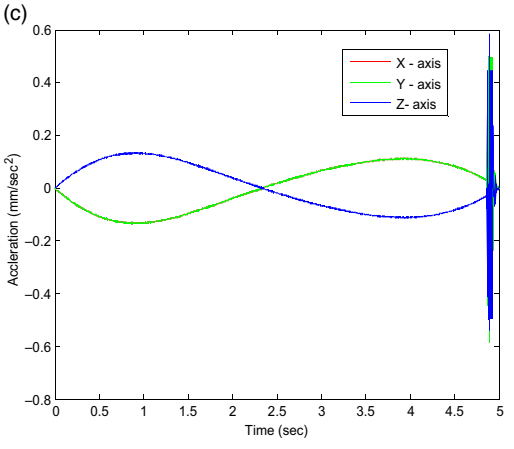
https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



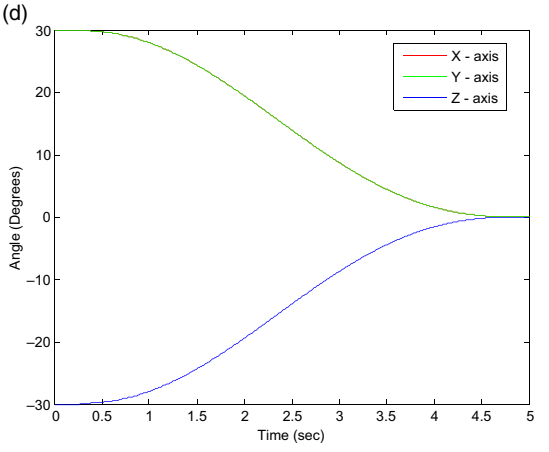
Forward Kinematics results of the Delta Robot



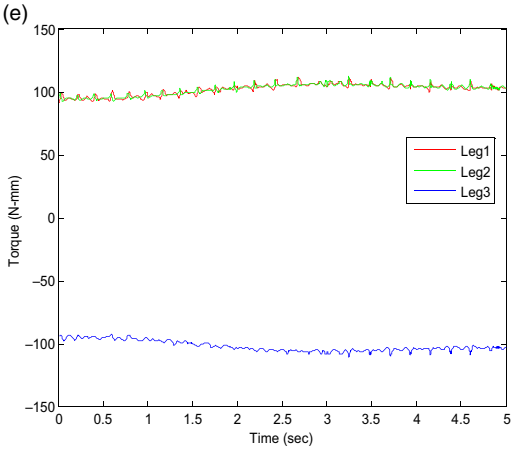
Velocity Profile for the P2 to P1 trajectory



Acceleration of the active joints (R1, R2, and R3)



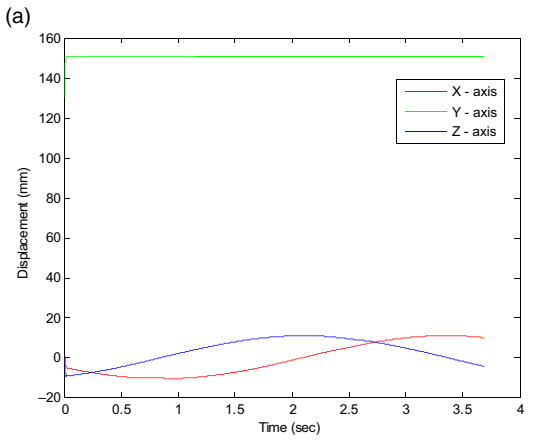
Joint angle



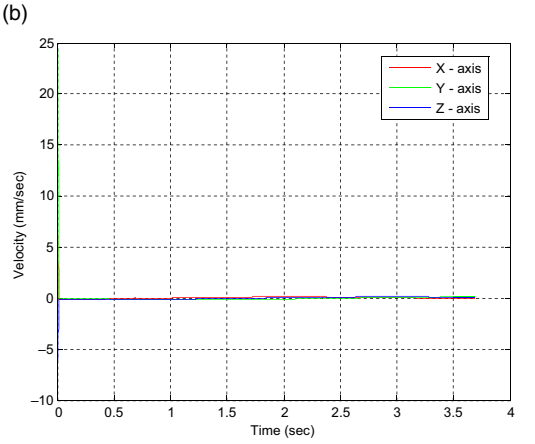
The torque acting on the active joints (R1, R2, R3)

Figure 16 *(a) Forward kinematics results of the delta robot [circular trajectory]. (b) Velocity profile for the P2 to P1 trajectory. (c) Acceleration of the active joints (R1, R2, R3). (d) Joint angle. (e) The torque acting on the active joints (R1, R2, R3).*

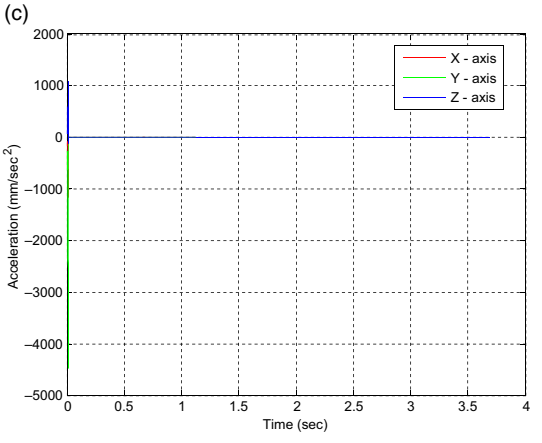
https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



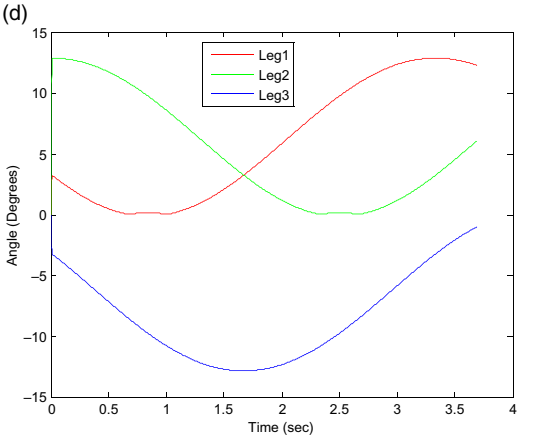
Forward Kinematics results of the Delta Robot



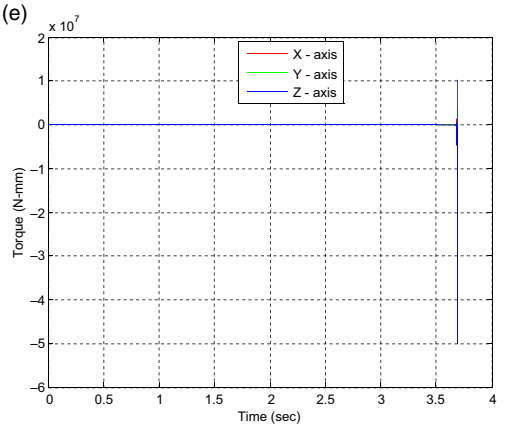
Velocity Profile for the P2 to P1 trajectory



Acceleration of the active joints (R1, R2, R3)



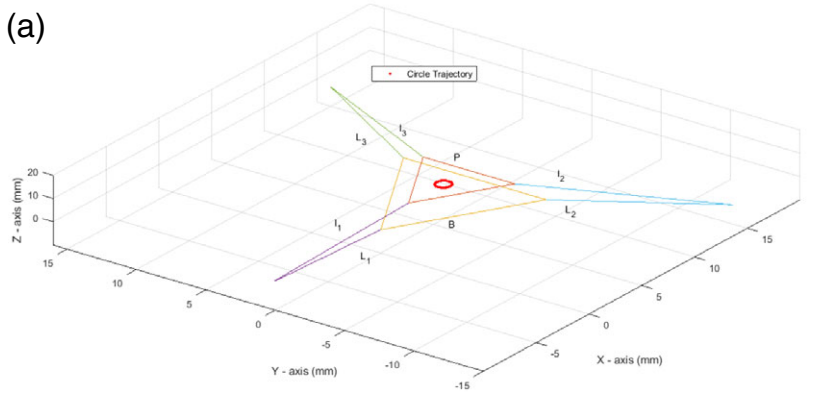
Joint angle



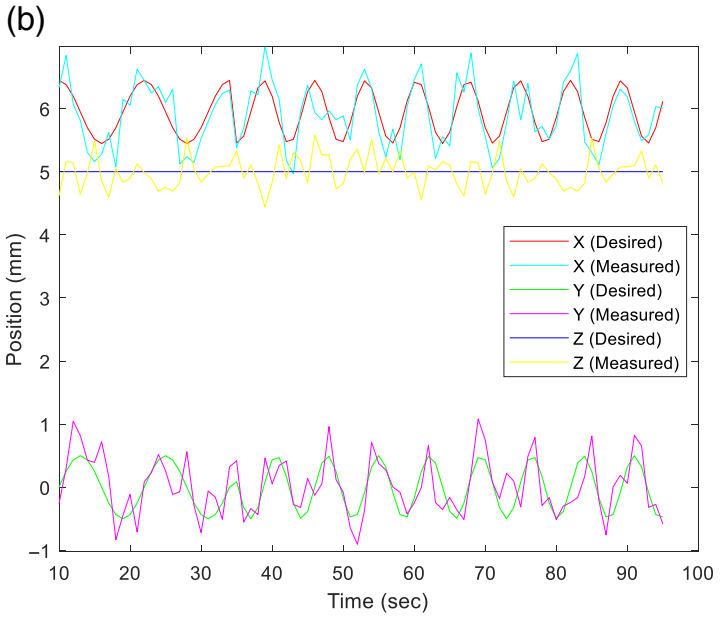
The torque acting on the active joints (R1, R2, R3)

Figure 17 *(a) Forward kinematics results of the delta robot [helical trajectory]. (b) Velocity profile for the P2 to P1 trajectory. (c) Acceleration of the active joints (R1, R2, R3). (d) Joint angle. (e) The torque acting on the active joints (R1, R2, R3).*

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press



A planar circular path is provided to the Delta robot



Desired and measured end-effector (circle trajectory) position

Figure 18 *(a) A planar circular path is provided to the delta robot. (b) Desired and measured end-effector (circle trajectory) position.*

7. Conclusion

In this paper, kinematic and dynamic analysis of the proposed architecture has been discussed. A description of the topological structure, geometrical model and the kinematics analysis of the three figured systems is presented. In kinematics analysis, inverse kinematics and forward kinematics solutions of the delta manipulator are implemented. The differences of the kinematics of the delta manipulator using forward kinematics based on encoders readings and the real-time vision-based system are termed as the error, and its characteristics have been investigated. The dynamic analysis was implemented using

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

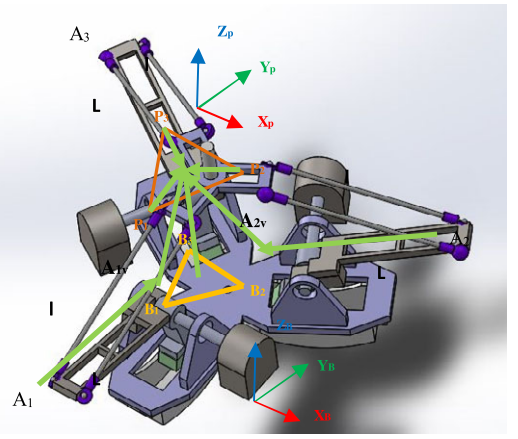
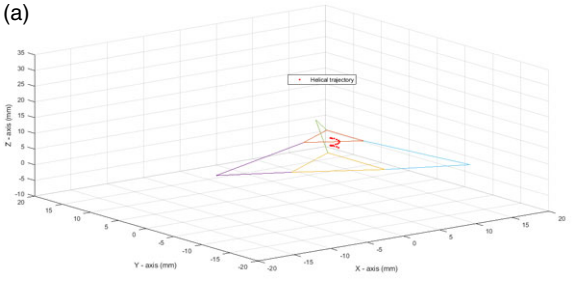
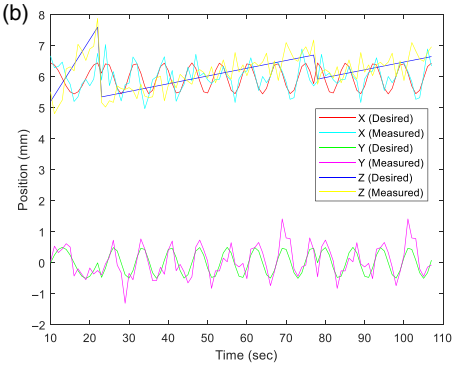


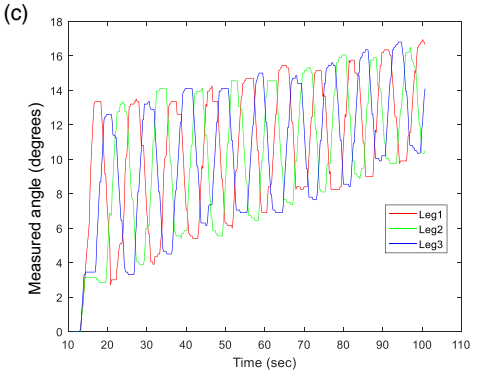
Figure 19. *Delta robot FPK diagram.*



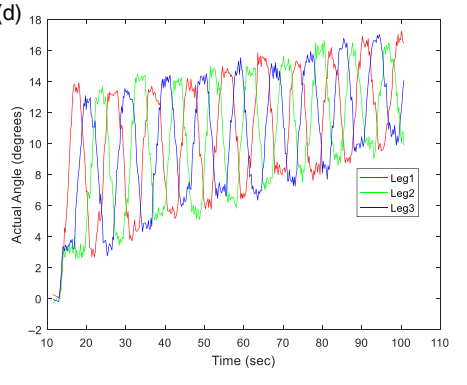
A helical trajectory path is provided to the Delta Robot



Desired and measured end-effector (helical trajectory) position



The desired angle for Leg1, Leg2 and Leg3 in helical profile



Measured angle for Leg1, Leg2 and Leg3 in helical profile

Figure 20 *(a) A helical trajectory path is provided to the delta robot. (b) Desired and measured end–effector (helical trajectory) position. (c) The desired angle for Leg1, Leg2, and Leg3 in helical profile. (d) Measured angle for Leg1, Leg2, and Leg3 in helical profile.*

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

the DeNOC approach. Various cases in real time and in the virtual environment, achieving the object's catching, have been discussed. In the future, we will implement the proposed framework to experiment with an irregular shaped object (like a semi filled bottle) with object-based classification techniques. It can be further extended by using the vision and electromechanical robot hardware in a real-time environment to demonstrate catching.

Acknowledgements. We hereby acknowledge the support of Dr. Anil Kumar Sharma for the help and Mechatronics Lab, IIT Delhi, for providing the facility and environment for carrying out the research activity.

Conflicts of Interest. None.

Financial Support. None.

Ethical Considerations. None.

Authors' Contributions. None.

References

[1] I.A. Bonev, "Delta Robot — The Story of Success," *Parallel MIC*, May 2001.
[2] P. J. Zsombor-Murray, *Descriptive Geometric Kinematic Analysis of Clavel's Delta Robot* (McGill University, Department of Mechanical. Engineering. Center for Intelligent Machine, Canada, 2004).
[3] L.-W. Tsai, *Position Analysis of Parallel Manipulator Robot Analysis – The Mechanics of Serial and Parallel Manipulators* (John Willey & Sons, Inc., New York, 1999).
[4] C. Alain, "Dynamic modeling of parallel robots for computed-torque control implementation," *Int. J. Rob. Res.* **17**(12), 1325–1336 (1998).
[5] M. A. Laribi, L. Romdhane and S. Zeghloul, "Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace," *Mech. Mach. Theory* **42**(7), 859–870 (2006).
[6] A. Kosinska, M. Galicki and K. Kedzior, *Designing and Optimization of Parameters of Delta-4 Parallel Manipulator for a Given Workspace* (Institute of Aeronautic and Applied Mechanics, Warsaw University of Technology, Poland, 2003).
[7] L.-W. Tsai, "Static Analysis of Parallel Manipulator," **In:** *Robot Analysis –The Mechanics of Serial and Parallel Manipulators* (A Wiley-Interscience Publication, New York, NY, 1999) pp. 1–519.
[8] J. Angeles and O. Ma, "Dynamic simulation of -axis serial robotic manipulators using a natural orthogonal complement," *Int. J. Rob. Res.* **7**(10), 32–47 (1988).
[9] J. Angeles and S. Lee, "The formulation of dynamical equations of holonomic mechanical system using a natural orthogonal complement," *ASME J. Appl. Mech.* **55**(5), 243–244 (1988).
[10] S.K. Saha, "Analytical expression for the inverted Inertia matrix of serial robots," *Int. J. Rob. Res.* **18**(1), 20–36 (1999).
[11] S.K. Saha, "Dynamics of serial multibody system using the decoupled natural orthogonal complement matrices," *ASME J. Appl. Mech.*, **66**, 986–996 (1999).
[12] S. K. Saha, "Recursive dynamic algorithms for serial, parallel, and closed-chain multibody systems," *Indo-US Workshop on Protein Kinematics & Protein Conformations* (IISc, Bangalore, Dec. 10–11, 2007).
[13] S.K. Saha, *Introduction to Robotics* (Tata McGraw-Hill, New Delhi, 2008).
[14] H. Shinno, H. Yoshioka and H. Sawano, "A newly developed long-range positioning table system with a sub-nanometer resolution," *CIRP Ann. Manuf. Technol.* **60**(1), 403–406 (2011).
[15] T. Atsumi, S. Nakamura, M. Furukawa, I. Naniwa and J. Xu, "Triple stage-actuator system of head-positioning control in hard disk drives," *IEEE Trans. Magnets.* **49**(6), 2738–2743 (2013).
[16] H. Asada and J. J. Slotine, Robot Analysis and Control (John Wiley & Sons, Inc., New York, NY, 1986).
[17] N. Hogan and E. Colgate, "Stability problem in contact tasks," **In:** *The Robotics Review* (Oussama Khatib, John J. Craig, and Tomas Lozano, eds.) (The MIT Press, Cambridge, MA, 1989) pp.339–348.
[18] J. McBean and Cynthia Breazeal, "Voice coil actuators for human-robot interaction," *International Conference on Intelligent Robots and Systems*, Sendai (2004).
[19] W. S. Philip, N. Nandha Kumar and A. K. Ramadorai, "Vision-based manipulation of non-rigid object," *Proceeding of IEEE International Conference on Robotics and Automation,* Minneapolis, MN (April 1996).
[20] S. Yonemoto and R. Taniguchi, "Vision-based 3D direct manipulation interface for smart interaction," *Proceedings of IEEE International Conference on Pattern Recognition*, Quebec City, QC (2002) pp. 655–658.
[21] Y. Wei, B. Goodwine, "Vision-based non-smooth kinematic stratified object manipulation," *Seventh International Conference on Control, Automation, Robotics and Vision (ICARCV02)*, Singapore (Dec 2002).
[22] C. H. Park and M. H. Ayanna, "Vision-based force guidance for improved human performance in a tele-operative manipulation system," *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA (2007).

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press

[23] A. Mkhitaryan and D. Burschka, "Vision-based haptic multisensory for manipulation of soft, fragile object," **In:** *Sensors* (Jin-Chern, ed.) (IEEE, Taipei, 2012) pp. 1–4.
[24] S. Y. Chen, J. Zhang, H. Zhang and N. M. Kwok, "Intelligent lighting control for vision-based robotic manipulation," *IEEE Trans. Ind. Electron.* **59**, 3254–3263 (2012).
[25] K. Sachin and M. Sudipto, "Vision-Based Kinematics Analysis of the Delta Robot for object Catching," *Robotica* **1**(31), 2661–2678 (2021).
[26] K. Sachin, K. Rajesh and M. Sudipto, "Color Invariant State Estimator to predict the object trajectory and catch Dexterous Multi-fingered Delta Robot," **In:** *Multimedia Tools and Applications*, 2020.
[27] K. Sachin and M. Sudipto, "Automatic single view monocular camera calibration based object manipulation using novel dexterous multi-fingered delta robot," *Neural Comput. Appl.* **7**(31), 2661–2678 (2017).
[28] C. John, *Introduction to Robotics: Mechanics and Control*, 4th Edition.
[29] V. Gupta, H. Chaudhary and S. Saha, "Dynamics and actuating torque optimization of planar robots," *J. Mech. Sci. Technol.* **1**, 2699–2704 (2015).
[30] A. M. Kushal, V. Bansal and S. Banerjee, "A simple method for interactive 3D reconstruction and camera calibration from a single view," *Proceedings: Indian Conference in Computer Vision, Graphics and Image Processing*, Mumbai (2002).
[31] A. Criminisi, I. Reid and A. Zisserman, "Single view metrology," *Int. J. Comput. Vision* **40**(2), 123–148 (2000).
[32] A. Criminisi, "Single- view metrology: Algorithms and applications," Invited Paper (Microsoft Research, One Microsoft Way, Redmond, WA, 2002).
[33] G. Wang, H. Tsui, H. Zhanyi and F. Wu, "Camera calibration and 3D reconstruction from a single view based on scene constraints," ELSEVIER, *Image Vision Comput.* **23**, 311–323 (2005).

Cite this article: S. Kansal and S. Mukherjee (2022). "Kinematic and dynamic analysis of a dexterous multi-fingered delta robot for object catching", Robotica **40**, 2878–2908. https://doi.org/10.1017/S0263574721002009

https://doi.org/10.1017/S0263574721002009 Published online by Cambridge University Press