Computed torque control of a prismatic-input delta parallel robot

Ahmed Fathy
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
ahmed.fathymorsy@student.guc.edu.eg

Mohammed Ashraf
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
mohammed.abdelrahim@student.guc.edu.eg

Ayman El-Badawy
Mechatronics Engineering Department
German University in Cairo
Cairo, Egypt
ayman.elbadawy@guc.edu.eg

***Abstract*—Delta parallel robots are regularly used in various industries, including but not limited to the pharmaceutical, food, automotive, and electronics industries. This nonlinear system possesses highly uncertain dynamics; therefore, controller design for delta robots tends to be quite challenging. Based on a simplified dynamic model which utilizes the principle of virtual work, a nonlinear computed torque controller is proposed for trajectory tracking control of a prismatic-input delta parallel robot. The simulation results demonstrate that the proposed controller can guarantee excellent trajectory tracking performance by considering nonlinear compensations. Lastly, simulation results are compared with a linear proportional-integral-derivative controller to show its effectiveness.**

***Index Terms*—Prismatic-input delta robot, computed torque control, trajectory tracking**

I. Introduction

Delta robots are prominently known for their fast operating speed, precision, and accuracy. From food packaging to surgical applications [1], delta robots have proven to be tremendously useful in a wide range of implementations. Invented in the 1980s by a team of researchers led by Reymond Clavel [2], the delta robot is comprised of several duplicate kinematic chains in parallel [3], [4]. The driving motors are usually attached to the fixed base, reducing the mass of the moving parts. This allows the manipulator to achieve high velocities and acceleration. Delta robots are mainly divided into 2, 3, and 4 degrees of freedom (DOF) delta robots. 2-DOF delta robots are the simplest parallel robots available and are only capable of translational motion in two directions of space [5]. 3-DOF delta robots can achieve translational motion in all three directions of space [6], [7], making them more applicable to real-life scenarios. 4-DOF delta robots [8] can also translate in three axes but use a redundant link to achieve faster speeds and acceleration. Delta robots are also available in two different input types: revolute, which follows Clavel's original design, and prismatic, which utilizes prismatic joints rather than revolute joints. The main advantages of prismatic delta robots are their robustness against external forces [9] and ease of modeling and solving their dynamics. To ensure adequate performance, a suitable control method which guarantees satisfactory trajectory tracking is required.

Control methods for parallel manipulators are classified into either kinematic control algorithms or dynamic control algorithms. Kinematic control strategies ignore the dynamic characteristics of the manipulator [10], whereas dynamic control strategies are based on the rigid body dynamics of the robot [11], [12]. Various kinematics-based controllers have been proposed. Reference [13] proposed a proportional-integral-derivative (PID) controller with tracking which can asymptotically track a desired input trajectory. An integrated controller consisting of a proportional-derivative (PD) controller and a saturated proportional-integral (PI) controller was designed, which was capable of converging both position and synchronization errors to zero [14]. Moreover, the effects on trajectory tracking between a standard PID controller and a PI synchronized type control algorithm were analyzed [15]. While the aforementioned controllers showed promising results, high-speed and accurate control of delta robots cannot be achieved by only considering the kinematics. Consequently, it is vital to study other control methods that consider the dynamics of delta robots.

Dynamic control for parallel robots is generally categorized into either computed torque control (CTC) [16] or dynamic feedforward control (DFC) [17]. The Lagrange-D'Alembert method was used to compute the inverse dynamics of a 2-DOF redundant parallel manipulator, based on which the performance of linear PD control and CTC was compared [18]. Reference [19] developed a neural network optimized CTC algorithm for a 2-DOF parallel manipulator titled Diamond 600, which proved to have favorable tracking ability and robustness. A dynamic controller was developed, which consisted of a kinematic control subsystem and a feedforward rigid-body dynamics-based control subsystem for a 2-DOF parallel robot [20]. This control method proved incredibly effective at eliminating the dynamic characteristics of the manipulator. Finally, a PD+ controller with an external state observer (ESOPD+) capable of achieving high accuracy and robust control for a 3-DOF revolute-input delta robot was proposed [21].

In this paper, a Computed Torque Control (CTC) algorithm is proposed for a 3-DOF prismatic-input delta robot to be employed in targeted drug delivery applications using micro and nanocarriers. The proposed controller offers highly accurate trajectory tracking control in rapid-speed motion. Besides, compared to PID control, CTC has a faster response with

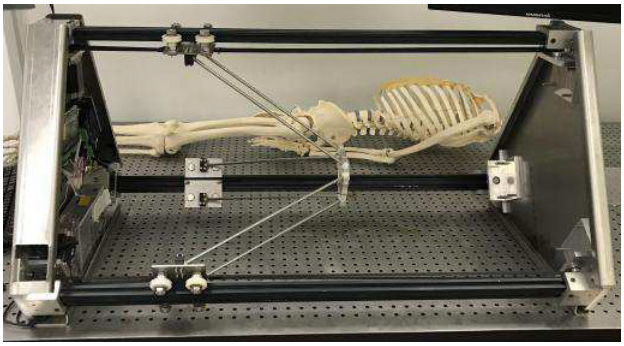

Fig. 1. Real-life delta robot

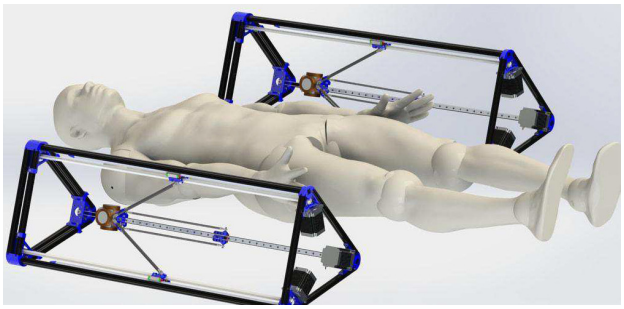

Fig. 2. Illustration of the setup

smaller gain parameters.

The rest of the paper is divided as follows: The "Application" section covers the intended application for this robot. In the "Dynamic analysis" section, the geometrical and structural characteristics of the robot are analyzed and a rigid body dynamics model is established based on the principle of virtual work. In "Controller design", based on the dynamic model, the CTC algorithm is proposed for the prismatic-input delta robot. A helical trajectory is given and tested in the "Simulation" section. On top of that, MATLAB's Simscape Multibody is employed to verify the effectiveness of the proposed control method. Lastly, conclusions are reached in the last section.

II. Application

Fig. 1 shows a real-life delta robot that includes a moving platform, a fixed frame, and three identical kinematic chains that are connected to the moving platform and fixed base by universal joints and prismatic joints, respectively. Each kinematic chain contains a slider (active limb) and a parallelogram link structure made of two jointed links (passive limb), which are connected to each other by a universal joint. The three driving motors and their respective drivers are installed on the fixed base. Combining the different motor torques, the robot can achieve purely translational motion in the *X*, *Y*, and *Z* axes.

While delta robots have numerous widespread applications, a unique implementation is targeted drug delivery. Fig. 2 [22] shows an illustration of the setup, which includes two identical delta robots with omnimagnets [23] attached to the moving platform. The patient can lie on the operation table and is injected with a nanocarrier that contains the drug. The magnetic field from the omnimagnets combined with the motion from the delta robots manipulates the movement of the nanocarrier in the bloodstream to deliver the drug to the required area. This has a multitude of advantages, such as giving adequately reduced dosages and having fewer side effects [24], [25].

III. Dynamic analysis

To facilitate the analysis, a kinematic diagram of the mechanism is pictured in Fig. 3. At the center of the fixed base

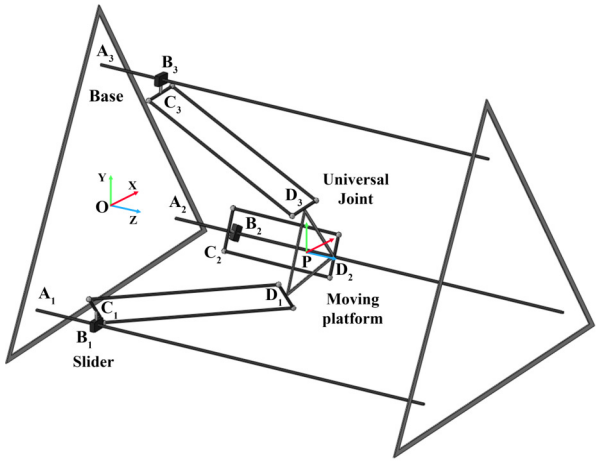

Fig. 3. Kinematic diagram of the delta robot

$A_1A_2A_3$, the origin O of the global coordinate system *O-XYZ* is defined with the *XY* plane on the base and the *Z*-axis perpendicular to it. A local frame *P-XYZ* is located at the center of the moving end platform with the *XY* plane on the platform and the *Z*-axis orthogonal to it. To further simplify the analysis of the robot, some geometric parameters are illustrated as follows: $OA_i = a_i$, $A_iB_i = q_iu_i$, $B_iC_i = c_i$, $C_iD_i = Ll_i$, $OP = P$, $PD_i = d_i$, $i = 1, 2, 3$. a_i represents the position vector of point A_i, $q_i\boldsymbol{u}_i$ is the slider displacement and its unit vector, c_i denotes the carriage offset, $L\boldsymbol{l}_i$ symbolizes the length of the passive parallelogram link and its unit vector, d_i is the end platform offset, and P is the position vector of the moving platform with respect to O.

The passive limbs of the delta robot are usually manufactured from lightweight materials. Additionally, complete dynamical models of the delta robot are too complex for real-time control purposes. Therefore, a method where the inertia of the passive links is ignored and the mass is split at its two extremity points is utilized [26], [27]. Additionally, the dynamic model present in [27] was adapted to match the robot under study. Therefore, using the principle of virtual work, the dynamic model for the prismatic delta robot is given as

$$f^T\partial q - \boldsymbol{G}_s^T\partial q - \boldsymbol{F}_s^T\partial q - \boldsymbol{G}_p^T\partial P - \boldsymbol{F}_p^T\partial P = 0 \quad (1)$$

$$\boldsymbol{F}_s = \begin{bmatrix} \bar{m}_s\ddot{q}_1 \\ \bar{m}_s\ddot{q}_2 \\ \bar{m}_s\ddot{q}_3 \end{bmatrix}, \boldsymbol{F}_p = \begin{bmatrix} \bar{m}_p\ddot{x} \\ \bar{m}_p\ddot{y} \\ \bar{m}_p\ddot{z} \end{bmatrix} \tag{2}$$

$$\boldsymbol{G}_s = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{G}_p = \begin{bmatrix} 0 \\ -\bar{m}_p g \\ 0 \end{bmatrix}$$

$$\bar{m}_p = m_p + m_{load} + 6wm_l \quad , \quad \bar{m}_s = m_s + 2(1-w)m_l$$

The symbols involved and their meanings are mentioned as follows:

f	Prismatic joint actuator forces
∂q	Virtual slider linear displacement
∂P	Virtual moving platform linear displacement
$\boldsymbol{G}_s$	Gravitational forces on the sliders
$\boldsymbol{F}_s$	Inertial forces of the sliders
$\boldsymbol{G}_p$	Gravitational forces on the moving platform
$\boldsymbol{F}_p$	Inertial forces of the moving platform
$\ddot{q}$	Linear acceleration of the slider
$\ddot{x}, \ddot{y}, \ddot{z}$	Linear acceleration of the platform in *X*, *Y*, and *Z* axes
$\bar{m}_p$	Equivalent mass of platform
$\bar{m}_s$	Equivalent mass of slider
m_p	Mass of platform
m_{load}	Mass of load
m_s	Mass of slider
m_l	Mass of link
g	Gravitational acceleration
w	Mass proportion ratio

From the kinematic model, a relationship between the linear velocity of the sliders and moving platform is formulated as

$$\dot{q} = J_q\dot{P}, \quad J_q = \begin{bmatrix} J_{q1} \\ J_{q2} \\ J_{q3} \end{bmatrix}, \quad J_{qi} = \frac{\boldsymbol{l}_i^T}{\boldsymbol{l}_i^T\boldsymbol{u}_i}, \quad \dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{3}$$

where $\dot{q}$ is the linear velocity of the sliders, J_q is the Jacobian matrix, and $\dot{P}$ is the linear velocity of the moving platform.

Additionally, the relationship between the linear acceleration of the sliders and moving platform is given by

$$\ddot{q} = J_q\ddot{P} + \boldsymbol{V}H\hat{\boldsymbol{V}} \tag{4}$$

$$\ddot{P} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \boldsymbol{V} = \begin{bmatrix} \dot{P}^T & 0 & 0 \\ 0 & \dot{P}^T & 0 \\ 0 & 0 & \dot{P}^T \end{bmatrix}, H = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ 0 & 0 & H_3 \end{bmatrix}$$

$$H_i = \frac{1}{L(\boldsymbol{l}_i^T\boldsymbol{u}_i)}(I_3 - \boldsymbol{u}_iJ_{qi})^T(I_3 - \boldsymbol{u}_iJ_{qi}), \quad \hat{\boldsymbol{V}} = \begin{bmatrix} \dot{P} \\ \dot{P} \\ \dot{P} \end{bmatrix}$$

where $\ddot{P}$ is the linear acceleration of the moving platform, H is the Hessian matrix, and I_3 is a 3×3 identity matrix.

Substituting (2), (3), and (4) into (1), the generalized form of the prismatic delta robot dynamic model is derived as

$$f = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) \tag{5}$$

TABLE I
PRISMATIC-INPUT DELTA ROBOT PARAMETERS

Symbol	Value	Unit	Symbol	Value	Unit
a_i	0.280	m	m_l	0.080	Kg
c_i	0.030	m	m_{load}	0	Kg
d_i	0.05351	m	g	9.81	m/s^2
L	0.365	m	z_{min}	0.3411	m
m_p	0.180	Kg	z_{max}	0.8521	m
m_s	0.560	Kg	w	0.5	-

where M is the mass inertia matrix, C is the centrifugal and Coriolis force matrix, and G is the gravitational force vector, are given by

$$M = M_s + J_q^{-T}M_pJ_q^{-1}, \quad C = J_q^{-T}M_pJ_q^{-1}(-\boldsymbol{V}H\hat{\boldsymbol{V}})$$

$$G = G_s + J_q^{-T}G_p \tag{6}$$

where the diagonal matrices M_s and M_p are represented as $M_s = diag(\bar{m}_s, \bar{m}_s, \bar{m}_s)$ and $M_p = diag(\bar{m}_p, \bar{m}_p, \bar{m}_p)$.

IV. Controller Design

In this section, a CTC algorithm is proposed for the prismatic-input delta robot. The controller can accurately track the desired path by taking the dynamic properties of the robot into account to achieve better performance.

With the following control objective in mind,

$$\lim_{t\to\infty} \begin{bmatrix} \tilde{q}(t) & \dot{\tilde{q}}(t) \end{bmatrix}^T = \begin{bmatrix} 0 & 0 \end{bmatrix}^T \tag{7}$$

where $\tilde{q}(t) = q_d(t) - q(t)$ and $\dot{\tilde{q}}(t) = \dot{q}_d(t) - \dot{q}(t)$. $q_d(t)$ and $q(t)$ are the desired and actual joint displacements, respectively. $\dot{q}_d(t)$ and $\dot{q}(t)$ are the desired and actual joint velocities, respectively.

The proposed controller can be formulated as

$$f = M(q)[\ddot{q}_d + K_v.\dot{\tilde{q}} + K_p.\tilde{q}] + C(q,\dot{q}).\dot{q} + G(q) \tag{8}$$

where $\ddot{q}_d$ is the desired joint acceleration, K_p and K_v are symmetric positive definite matrices which contain the proportional and derivative gains, respectively.

V. Simulations

A. Simulation parameters

In order to verify the effectiveness of the proposed controller, a 3D model of the prismatic-input delta robot was built in SolidWorks. Consequently, the model is imported into MATLAB's Simscape Multibody. The delta robot parameters are given in Table I, where z_{min} and z_{max} are the minimum and maximum linear displacements for the moving platform in the *Z*-axis, respectively. The moving platform initial position is $(0, 0, z_{min})$.

As shown in Fig. 4, the robot was given a helical path along the *Z*-axis to follow represented in the frame *O-XYZ* as

$$x = r_0 sin(at) \quad y = r_0 cos(bt) \quad z = z_0 + vt \tag{9}$$

where r_0 is the helix radius in meters, t is the time variable in seconds, z_0 is the initial end effector displacement in the *Z*-axis, a and b are frequencies, and v is a scaling constant. The values for these parameters are shown in Table II.

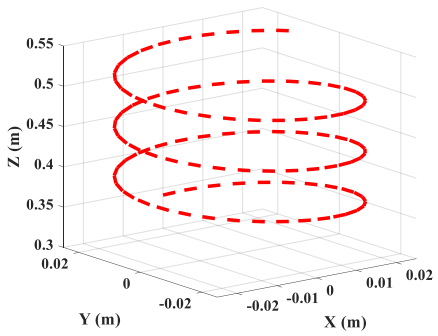

Fig. 4. Desired trajectory

TABLE II
HELICAL TRAJECTORY PARAMETERS

Variable	Value	Variable	Value
r_0	0.025	b	2
z_0	0.3411 m	v	0.02
a	2	t	$0 \rightarrow 10$ s

B. Results

To verify the success of the proposed method, it is compared with a linear PID control algorithm. For CTC, the proportional gain was chosen as $K_p = diag(1500, 1500, 1500)$ and the derivative gain was selected as $K_v = diag(200, 200, 200)$. For PID, the proportional, integral, and derivative gains were set as $K_p = diag(2500, 2500, 2500)$, $K_i = diag(200, 200, 200)$, and $K_d = diag(350, 350, 350)$, respectively.

As illustrated in Figs. 5 and 6, CTC has a lower displacement error overall than PID and achieves better and faster trajectory tracking control accuracy of the moving platform despite the PID controller having much larger gains. Fig. 5 shows that the performance of PID control is significantly lower than CTC as it does not consider the dynamic characteristics of the robot. An instance in Fig. 6, precisely the Error X graph, shows that PID performs better than CTC. This is likely due to the fact that CTC is heavily dependent on the accuracy of the dynamic model. The performance of CTC is negatively impacted by the simplistic nature of the model established in this paper. Nevertheless, it prominently outperforms PID control. Moreover, due to the variations in velocity and acceleration, the mass proportion ratio w should not be constant and can have a great influence on the results of the CTC algorithm.

To better assess the performance of the CTC method, the root mean square error (RMSE) is adopted from [21] to compare the results of CTC with PID control. The RMSE results of the PID and CTC control algorithms are presented in Table III. It is quite evident that the RMSE of the CTC algorithm is smaller than that of PID throughout all three axes. This shows that considering the dynamic characteristics of the manipulator yields superior control performance.

Taking a look at the input forces to the system in Fig. 7, it is observed that the inputs are smooth throughout, meaning there

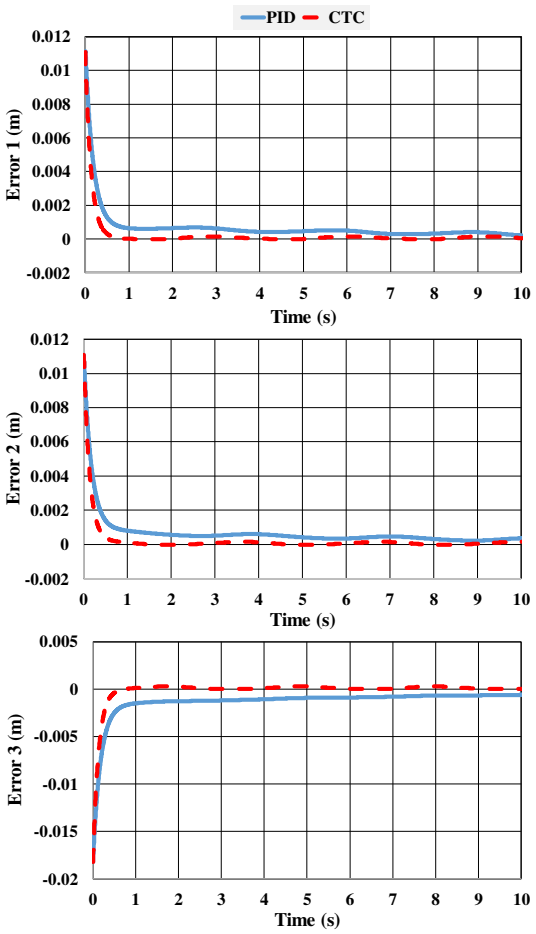

Fig. 5. Distance error of the three prismatic joints, q_1, q_2, and q_3

TABLE III
RMSE COMPARISON BETWEEN PID AND CTC (IN METERS)

Controller	Error X	Error Y	Error Z
PID	4.3934e-05	0.0023	1.5499e-04
CTC	6.0694e-05	0.0017	8.5907e-05

is little to no chattering, vibration, or jerky movement of the delta robot, which is ideal for real-life tests on the hardware.

VI. CONCLUSION

In this paper, a CTC algorithm is proposed for trajectory tracking control of a 3-DOF prismatic-input delta robot. By introducing a mass proportion ratio, a dynamic model is derived based on the principle of virtual work. The controller takes into account the dynamic properties of the robot to achieve accurate performance. Simulations in MATLAB Simscape Multibody verified the effectiveness of the proposed controller. Compared to a PID control algorithm, CTC provided superior performance and accuracy.

s

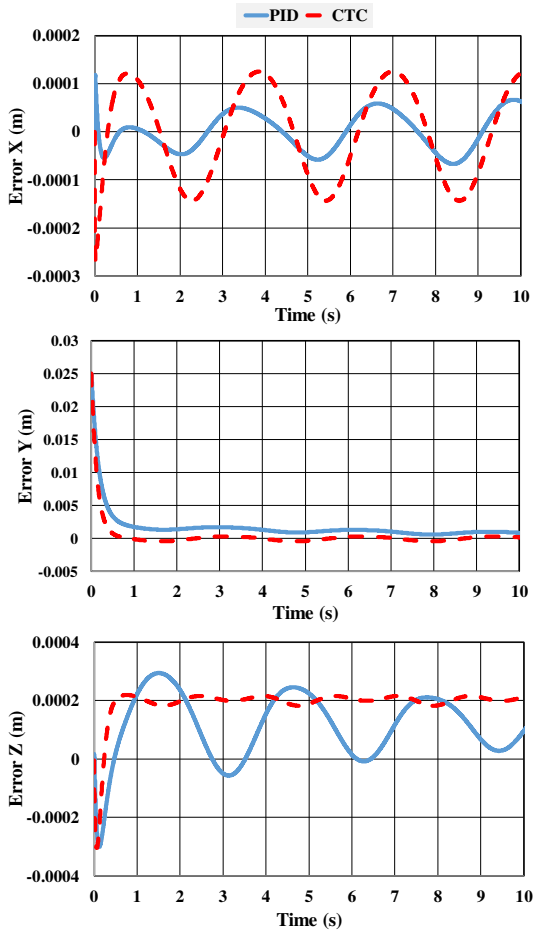

Fig. 6. Trajectory error in X, Y and Z axes

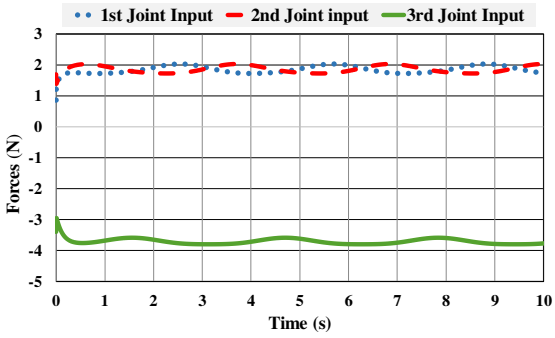

Fig. 7. Force inputs to the delta robot

REFERENCES

[1] Saadettin Aksoy and Emrah Ozan. Robots and their applications. *International Research Journal of Engineering and Technology (IRJET)*, 07:2001–2009, 02 2020.

[2] Ilian Bonev. Delta parallel robot — the story of success, 01 2001.

[3] J Kroneis, P Müller, and S Liu. Simplified modelling and parameter identification of parallel robots with redundancies. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 223(1):95–116, 2009.

[4] B. Zi, B.Y. Duan, J.L. Du, and H. Bao. Dynamic modeling and active control of a cable-suspended parallel robot. *Mechatronics*, 18(1):1–12, 2008.

[5] Tian Huang, Songtao Liu, Jiangping Mei, and Derek G. Chetwynd. Optimal design of a 2-dof pick-and-place parallel robot using dynamic performance indices and angular constraints. *Mechanism and Machine Theory*, 70:246–253, 2013.

[6] med amine Laribi, Lotfi Romdhane, and S. Zeghloul. Analysis and dimensional synthesis of the delta robot for a prescribed workspace. *Mechanism and Machine Theory*, 42:859–870, 07 2007.

[7] Luis Angel Castañeda, Alberto Luviano-Juárez, and Isaac Chairez. Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Transactions on Control Systems Technology*, 23(4):1387–1398, 2015.

[8] Erol Ozgur, Nicolas Bouton, Nicolas Andreff, and Philippe Martinet. Dynamic control of the quattro robot by the leg edges. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2731 – 2736, 06 2011.

[9] Vitor Borrelli. Kinematic and dynamic analysis of a machine for additive manufacturing, 7 2018.

[10] G.K. Mann and B.W. Surgenor. Model-free intelligent control of a 6-dof stewart-gough based parallel manipulator. In *Proceedings of the International Conference on Control Applications*, volume 1, pages 495–500 vol.1, 2002.

[11] Weiwei Shang and Shuang Cong. Nonlinear adaptive task space control for a 2dof redundantly actuated parallel manipulator. *Nonlinear Dynamics*, 59:61–72, 01 2010.

[12] Ayman El-Badawy and Khaled Youssef. On modeling and simulation of 6 degrees of freedom stewart platform mechanism using multibody dynamics approach. In *Proceedings of the ECCOMAS Thematic Conference Multibody Dynamics*, 07 2013.

[13] N. C. Ruiz-Hidalgo, A. Blanco-Ortega, A. Abúndez-Pliego, J. Colín-Ocampo, and M. Arias-Montiel. Design and control of a novel 3-dof parallel robot. In *2016 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pages 66–71, 2016.

[14] Yuxin Su, Dong Sun, Lu Ren, and James Mills. Integration of saturated pi synchronous control and pd feedback for control of parallel manipulators. *Robotics, IEEE Transactions on*, 22:202 – 207, 03 2006.

[15] Lu Ren, James K. Mills, and Dong Sun. Experimental comparison of control approaches on trajectory tracking control of a 3-dof parallel robot. *IEEE Transactions on Control Systems Technology*, 15(5):982–988, 2007.

[16] L.S. Guo and Q. Zhang. Adaptive trajectory control of a two dof closed-chain robot. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 1, pages 658–663 vol.1, 2001.

[17] M. Honegger, R. Brega, and G. Schweiter. Application of a nonlinear adaptive controller to a 6 dof parallel manipulator. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1930–1935 vol.2, 2000.

[18] Hui Cheng, Yiu-Kuen Yiu, and Zexiang Li. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on Mechatronics*, 8(4):483–491, 2003.

[19] Zhiyong Yang, Jiang Wu, and Jiangping Mei. Technical note. *Mechatronics*, 17(7):381–390, 2007.

[20] Qi Hao, Liwen Guan, Liping Wang, and Hua Shao. Dynamic feedforward control of the 2-dofs parallel manipulator of a hybrid machine tool. In *IEEE ICCA 2010*, pages 528–533, 2010.

[21] Mingkun Wu, Jiangping Mei, Jinlu Ni, and Weizhong Hu. Trajectory tracking control of delta parallel robot based on disturbance observer. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 235(7):1193–1203, 2021.

[22] H.M. Elhefnawi. Modeling and motion control of the delta robot, 6 2018.

[23] Amr Hassan and Ayman El-Badawy. Novel omnimagnet actuation method for a cubesat nano-satellite. *Aerospace Science and Technology*, 117:106913, 06 2021.

[24] Payam Abasian, Sonya Ghanavati, Saeed Rahebi, Saied Nouri Khorasani, and Shahla Khalili. Polymeric nanocarriers in targeted drug delivery systems: A review. *Polymers for Advanced Technologies*, 31, 08 2020.

[25] Christine Schmidt, Mariana Medina-Sánchez, Richard Edmondson, and Oliver Schmidt. Engineering microrobots for targeted cancer therapies from a medical perspective. *Nature Communications*, 11, 12 2020.

[26] Peng Xu, Bing Li, and Chi-Fai Chueng. Dynamic analysis of a linear delta robot in hybrid polishing machine based on the principle of virtual work. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 379–384, 2017.

[27] Yangmin Li and Qingsong Xu. Dynamic modeling and robust control of a 3-prc translational parallel kinematic machine. *Robotics and Computer-integrated Manufacturing*, 25:630–640, 2009.