*Research Article*

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

*International Journal of Advanced Robotic Systems*
November-December 2017: 1–14

DOI: 10.1177/1729881417738738
journals.sagepub.com/home/arx
SAGE

# Experimental and simulation studies of motion control of a Delta robot using a model-based approach

**Yong-Lin Kuo and Peng-Yu Huang**

**Abstract**

Robot manipulators are mostly used to complete positioning tasks, and the conventional approach utilizes position-based control scheme, which converts a robot control problems to several motor control problems, where encoder signals are used to fulfill feedback control. This control scheme might not provide excellent positioning accuracy due to the lack of the dynamics of robot manipulators. This article presents a simple model-based control scheme to achieve the positioning control and applies the scheme to a Delta robot, which can perform the motion with three translational degrees of freedom. The proposed scheme first evaluates the applied torques of motors based on robot's kinematics and dynamics using the encoders, and then the evaluated torques are sent to controllers to fulfill feedback control. The entire control loop is little similar to torque control, but torque or current sensors are not used. To demonstrate the performances of the proposed scheme, not only the numerical simulations are performed but also the experimental work is conducted. Both results are compared with those by applying the position-based control. The results show that the proposed model-based control scheme provides better positioning accuracy.

**Keywords**

Delta robot, motion control, model-based control

Date received: 5 January 2017; accepted: 19 August 2017

Topic: Special Issue – Theoretical and Experimental Technologies for Advanced and Basic Machines
Topic Editor: Pedro Ponce
Associate Editor: Arturo Molina

## Introduction

Parallel robot manipulators have broadly applied in industry due to their advantages such as high speed, high stiffness, high accuracy, low moving inertia, and so on. The Delta robot is one of the most common parallel robot manipulators, because it is usually applied to operate a pick-and-place task under the requirements of high moving speed and high positioning accuracy.[1,2]

Over the past decades, there are numerous researchers studied the motion control of a Delta robot. Castañeda et al. designed an adaptive controller, which aims to achieve the trajectory tracking for a Delta robot with uncertainties. Besides, the adaptive controller is incorporated with an observer due to the lack of the measurements of the joint velocities of the Delta robot.[3] Ramírez-Neria et al. investigated a robust trajectory tracking, where the controller is based on a linear feedback control and is capable of actively rejecting disturbance using a linear disturbance observer.[4] Guglielmetti and Longchamp developed a Delta robot model, where the state variables are the joint angles

Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan

**Corresponding author:**
Yong-Lin Kuo, Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, No. 43, Sec. 4, Keelung Road, Taipei 10607, Taiwan.
Email: yl_kuo@yahoo.com

and the end-effector coordinates. Besides, the applied torques can be evaluated for the purpose of achieving a linear feedback control.[5] Lu et al. developed a type-2 fuzzy logic controller for the trajectory tracking control of the Delta robot. The controller is designed based on a type-1 fuzzy logic controller, whose membership functions are blurred by three methods to generate the type-2 fuzzy sets.[6] Dumlu and Erenturk applied the classical proportional-integral-derivative (PID) control and the fractional-order PID control to a Delta robot so as to improve the trajectory tracking performances. The controller parameters are obtained using the pattern search algorithm and mathematical methods for the classical PID control and the fractional-order PID controller, respectively.[7] Lin et al. implemented a smoothing robust control method, manifold deformation design scheme, to guarantee the smoothly and robustly dynamic behavior as designed.[8] This method has the capabilities of dynamics prediction and disturbance estimation and then outputs the control efforts to deform the dynamic manifold of the controlled plant into the desired manifold. Zhang et al. investigated the conveyor tracking of a Delta robot, and the controller design is based on a computer vision system and encoder signals so as to compute the joint torques.[9] Rachedi et al. applied the $H^{\infty}$ controller to a Delta robot, where the controller is multi-input multi-output centralized feedforward control based on the computed joint torques.[10]

Generally speaking, there are two types of control schemes for robot manipulators, position- and model-based control. The position-based control divides a manipulator as many separated active joint systems, and each active joint is considered as position control of a motor.[11–13] This type of control scheme might not always produce high positioning accuracy due to the lack of the dynamics of manipulators. However, the model-based control incorporates the dynamics in the controllers. Many researchers studied various model-based controls and applied them to robot manipulators. Diaz-Rodriguez et al. applied a passivity-based controller to a three degrees of freedom (3-DOFs) prismatic-revolute-spherical parallel manipulator, where a manipulator is simplified as a reduced model.[14] Zhang et al. proposed an ankle assessment and rehabilitation robot, which can perform a real-time ankle-assessment three-dimensional (3-D) robotic training, where the control is an open-loop controller.[15] Besides, an experiment was performed on a healthy person. Meng et al. investigated a trajectory tracking of a 6-DOF parallel robot, where a model-based controller was proposed based on the off-line multibody dynamics of the robot. This approach can save the online computations.[16] Gao et al. proposed a three-axis parallel mechanism, which is used to test the reliability of multiaxis CNC machine tools. The mechanism exerts specific load spectrums on the spindle, and the dynamic model is derived by employing the virtual work principle. Besides, the controller is designed based on the established model.[17] Zhang et al. developed a trajectory tracking controller and applied it to a 2-DOF translational parallel manipulator, which is driven by linear motors.[18] The controller is designed based on a dynamic model, where a cascade PID/PI controller and a velocity feedforward controller are incorporated.

To obtain precise position control without implementing additional sensors, the computed torque control is developed based on inverse dynamics of robots. Codourey presented the control of a Delta robot by implementing the computed torque approach, where the torque is obtained by applying the dynamic equations, which is derived based on the virtual work principle.[19,20] The derived equations lead to complicated formulations, because the mass matrix in the equations is determined using the Jacobian matrices for each component in the robot. Therefore, the author proposed an approach to simplify the mass matrix, which does not explicitly appear in the formulations. Yang et al. investigated the motor-mechanism dynamics of a 2-DOF parallel manipulator, and the computed torque control combined with a PID controller was applied to the manipulator, where the controller parameters are determined using the three-layer neural network algorithm based on the errors of the angular displacements and velocities.[21] Conventionally, the computed torque control is usually applied with a linear PD controller. Shang and Cong developed a nonlinear computed torque control to combine with a nonlinear PD controller and applied the control to a planar parallel manipulator, whose stability is proven using the Lyapunov theorem.[22] Shang et al. proposed an adaptive computed torque controller for the trajectory tracking of a 2-DOF parallel manipulator with redundant actuation, and the dynamic model includes the frictions of active joints, which are determined by the adaptive laws.[23] Masarati applied it to redundant manipulators using general-purpose multibody formulations and software tools, where the control scheme consists of inverse kinematics to compute positions, velocities, and accelerations and inverse dynamics problem to compute feedforward generalized driving forces.[24]

In reviewing literature, the computed torque control relies on inverse dynamics combined with a tracking control of angular displacements and velocities, and some researchers applied the control law with feedforward control. This article aims to provide a simple model-based control scheme, where the computed torques are obtained based on kinematics and dynamics of manipulators without applying feedforward control. The control scheme is little similar to a torque control, but torque or current sensor is not implemented. Thus, encoders are the only sensors used to compute the applied torques. The proposed approach is applied to a Delta robot, whose motion has three translating DOFs. This article is organized as follows. "Kinematics and dynamics of a Delta robot" section introduces the kinematics and the dynamics of the Delta robot. "Model-based tracking control" section proposes a model-based control scheme for the robot. "Experimental setup" section presents the experimental setup and validation. "Experimental and simulation results" section demonstrates the

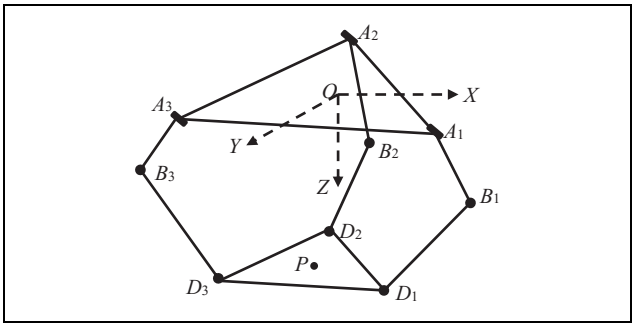


**Figure 1.** Schematic diagram of a Delta robot.

experimental and simulation results. "Conclusions" section summarizes some significant conclusions.

## Kinematics and dynamics of a Delta robot

A Delta robot consists of three kinematic chains, and each chain consists of two links ($A_iB_i$ and $B_iD_i$, where $i = 1, 2, 3$, see Figure 1). Each chain links a fixed base (triangle $A_1A_2A_3$) and a moving platform (or called an end effector, triangle $D_1D_2D_3$), where three motors are mounted at points $A_i$ to drive links $A_iB_i$. A coordinate system is defined on the fixed base, and the origin is located at its center, where the $X$-axis is arbitrarily defined on the base, the $Z$-axis is downward and perpendicular to the base, and the $Y$-axis is orthogonal to the $X$- and $Z$-axes. Thus, the motion of the end effector is driven by the three motors through the three kinematic chains, and the end effector has the 3-DOF translating motions. Based on the geometry of the Delta robot, each point $A_i$ has 1-DOF rotating motions, and each points $B_i$ and $D_i$ have 2-DOF rotating motions.

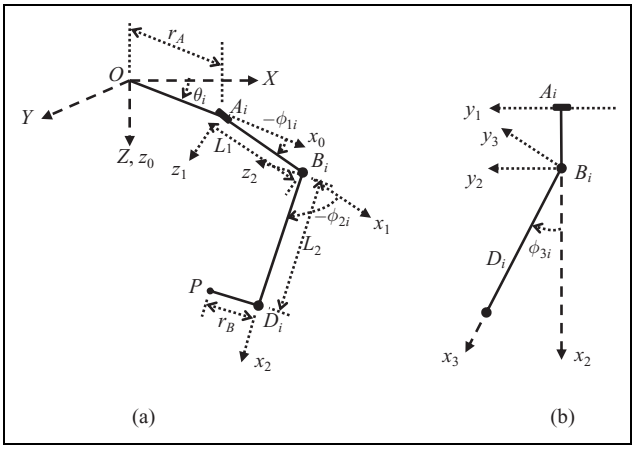


**Figure 2.** Schematic diagram of a kinematic chain (a) from a 3-D view and (b) on the $x$–$y$ plane.

### Kinematics

The size of each kinematic chain is assumed to be identical as shown in Figure 2, where $L_1$, $L_2$, $r_A$, and $r_B$ are the lengths of $B_iD_i$, $A_iB_i$, $OA_i$, and $D_iP$, respectively; $\phi_{1i}$ is the rotating angle of link $A_iB_i$ with respect to the fixed base; $\phi_{2i}$ and $\phi_{3i}$ are the rotating angles of link $B_iD_i$ with respect to link $A_iB_i$. Based on the geometry of the Delta robot, the position of point $P$ at end effector with respect to the $XYZ$ coordinate is written as

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \begin{bmatrix} \cos\theta_i(r_A - r_B + L_1\cos\varphi_{1i} + L_2\cos(\varphi_{1i}+\varphi_{2i})\cos\varphi_{3i}) - L_2\sin\theta_i\sin\varphi_{3i} \\ \sin\theta_i(r_A - r_B + L_1\cos\varphi_{1i} + L_2\cos(\varphi_{1i}+\varphi_{2i})\cos\varphi_{3i}) + L_2\sin\theta_i\cos\varphi_{3i} \\ L_1\sin\varphi_{1i} + L_2\sin(\varphi_{1i}+\varphi_{2i})\cos\varphi_{3i}) \end{bmatrix}, (i = 1, 2, 3) \quad (1)$$

Equation (1) represents the position of point $P$ through one of the three kinematic chains. By eliminating the passive joint angles $\varphi_{2i}$ and $\varphi_{3i}$, equation (1) can be simplified as

$$(X_P - X_i)^2 + (Y_P - Y_i)^2 + (Z_P - Z_i)^2 = L_2^2, \ (i = 1, 2, 3) \quad (2)$$

where

$$X_i = (r + L_1\cos\varphi_{1i})\cos\theta_i, \ \ Y_i = (r + L_1\cos\varphi_{1i})\sin\theta_i,$$
$$Z_i = -L_1\sin\varphi_{1i} \quad \text{and} \quad r = r_A - r_B.$$

*Direct kinematics.* The direct kinematics determines the position $(X_P, Y_P, Z_P)$ of the end effector of a Delta robot using the given active joint angles $\varphi_{1i}$. Equation (1) represents three spherical equations, whose center and radius are $(X_i, Y_i, Z_i)$ and $L_2$, respectively, and the equations are functions of the position $(X_P, Y_P, Z_P)$. Therefore, the intersection of the three spheres is the solution of equation (1). Besides, there are four possibilities after solving the equations (for more details see the study by Tsai[25]).

*Inverse kinematics.* The indirect kinematics determines the active joint angles $\varphi_{1i}$ based on a given position $(X_P, Y_P, Z_P)$ of the end effector. By expanding equation (2), the equation can be reduced as

$$l_i\cos\varphi_{1i} + m_i\sin\varphi_{1i} = n_i, \ (i = 1, 2, 3) \quad (3)$$

where

$$l_i = 2rL_1 - 2L_1X_P\cos\theta_i - 2L_1Y_P\sin\theta_i, \ m_i = 2L_1Z_P, \ \text{and}$$
$$n_i = 2rX_P\cos\theta_i - 2rY_P\sin\theta_i + X_P^2 + Y_P^2 + Z_P^2 + L_1^2 - L_2^2 + r^2 \quad (4)$$

Equation (3) refers to the functions of the active angles $\varphi_{1i}$, which can be determined by a single equation of equation (3), and four possibilities can be obtained after solving the equation.[26]

*Workspace.* The workspace of the Delta robot is a 3-D region which can be attained by the end effector of the robot. Based on equation (3) and trigonometry, the equation must satisfy the condition as

$$\left|\frac{n_i}{\sqrt{l_i^2+m_i^2}}\right| \le 1 \tag{5}$$

Note that equation (5) is a function of the position $(X_P, Y_P, Z_P)$ of the end effector only. Therefore, substituting any position $(X_P, Y_P, Z_P)$ can check if it is within the workspace of the robot or not.

*Singularities.* The singularities of the Delta robot interpret the relationships between the joint rates of the Delta robot and the velocities of the end effector. There are two types of singularities, direct singularities and inverse singularities, and they are determined by examining the singularities of Jacobian matrices. Taking the time derivative of equation (2) leads to

$$J_P\dot{\vec{v}} = J_\phi\ddot{\vec{\phi}} \tag{6}$$

where

$$\dot{\vec{v}} = \begin{bmatrix}\dot{X}_p & \dot{Y}_p & \dot{Z}_p\end{bmatrix}^T, \ddot{\vec{\varphi}} = \begin{bmatrix}\ddot{\varphi}_{1i} & \ddot{\varphi}_{2i} & \ddot{\varphi}_{3i}\end{bmatrix}^T, J_P = \begin{bmatrix}J_{1x} & J_{1y} & J_{1z}\\ J_{2x} & J_{2y} & J_{2z}\\ J_{3x} & J_{3y} & J_{3z}\end{bmatrix}, J_{ix} = \sin\varphi_{3i}\cos(\varphi_{1i}+\varphi_{2i})\cos\theta_i + \cos\varphi_{3i}\sin\theta_i,$$
$$J_{iy} = -\sin\varphi_{3i}\cos(\varphi_{1i}+\varphi_{2i})\cos\theta_i + \cos\varphi_{3i}\cos\theta_i, J_{iz} = \sin\varphi_{3i}\sin(\varphi_{1i}+\varphi_{2i}), \text{ and}$$
$$J_\varphi = \text{diag}(\sin\varphi_{21}\sin\varphi_{31}, \sin\varphi_{22}\sin\varphi_{32}, \sin\varphi_{23}\sin\varphi_{33}). \tag{7}$$

The direct singularities are determined by calculating the roots of the determinant of $J_P$. Although it is complicated to find them, a solutions is $\varphi_{3i} = 0, \pi$ or $\varphi_{1i}+\varphi_{2i} = 0, \pi$.[17] Similarly, the inverse singularities are determined by evaluating the roots of the determinant of $J_\varphi$, and the solutions are $\varphi_{2i} = 0, \pi$ or $\varphi_{3i} = 0, \pi$.

## Dynamics

To investigate the dynamics of the robot, one assumes that there are no frictions on all joints and no flexibility on all links. The dynamic equations of the robot will be formulated by applying the Lagrange's equations. The kinetic and potential energies of the robot are first written as

$$T_m = \frac{1}{2}m_P(\dot{X}_P^2+\dot{Y}_P^2+\dot{Z}_P^2) + \frac{1}{6}m_1L_1^2\sum_i\dot{\phi}_{1i}^2 + \frac{1}{2}m_2\sum_i(\dot{X}_P^2+\dot{Y}_P^2+\dot{Z}_P^2+L_1^2\dot{\phi}_{1i}^2) \tag{8}$$

$$V_m = m_PgZ_P + \frac{1}{2}m_1gL_1\sum_i\sin\phi_{1i} + m_2g\sum_i(Z_P+L_1\sin\phi_{1i}) \tag{9}$$

where $m_P$, $m_1$, and $m_2$, respectively, refer to the masses of the end effector, the links $A_iB_i$, and the links $B_iD_i$.

Since equations (8) and (9) are the functions of six variables $X_P$, $Y_P$, $Z_P$, and $\phi_{1i}$, the Lagrangian should incorporate with three constraint equations as equation (2), which leads to the dynamic equations as

$$(m_P+3m_2)\ddot{X}_P - 2\sum_i\lambda_i(X_P + r\cos\theta_i - L_1\cos\varphi_{1i}\cos\theta_i) = F_{PX} \tag{10}$$

$$(m_P+3m_2)\ddot{Y}_P - 2\sum_i\lambda_i(Y_P + r\sin\theta_i - L_1\cos\varphi_{1i}\sin\theta_i) = F_{PY} \tag{11}$$

$$(m_P+3m_2)\ddot{Z}_P - 2\sum_i\lambda_i(Z_P - L_1\sin\varphi_{1i}) + (m_P+3m_2)g = F_{PZ} \tag{12}$$

$$\left(\frac{1}{3}m_1+m_2\right)L_1^2\ddot{\varphi}_{1i} + \left(\frac{1}{2}m_1+m_2\right)gL_1\cos\varphi_{1i} - 2\lambda_iL_1[(X_P\cos\theta_i + Y_P\sin\theta_i + r)\sin\varphi_{1i} - Z_P\cos\varphi_{1i}] = \tau_i,\ (i=1,2,3) \tag{13}$$

where $\lambda_i$ are the Lagrange multipliers; $F_{PX}$, $F_{PY}$, and $F_{PZ}$ are the applied external forces at the point $P$; $\tau_i$ are the applied external torques at the active joints. Note that equations (10) to (13) are functions of nine variables $X_P$, $Y_P$, $Z_P$, $\phi_{1i}$, and $\lambda_i$. Therefore, the equations will be solved with the constraint equations as equation (2). These equations are differential-algebraic equations.

An alternative application of the equations is used to determine the applied torques $\tau_i$, and the procedure is explained as follows.

1. The kinematic equations shown in equation (2) are solved, and then the active angles $\varphi_{1i}$ and the coordinates $X_P$, $Y_P$, and $Z_P$ can be obtained. Moreover, the first- and second-time derivatives can be calculated.
2. The Lagrange multipliers $\lambda_i$ can be obtained by solving equations (10) to (12), where the applied forces $F_{PX}$, $F_{PY}$, and $F_{PZ}$ are given.
3. The applied torques $\tau_i$ can be calculated using equation (13).

This procedure can be used for a torque control of the robot. Besides, the proposed control scheme applies this procedure to compute the desired torques and the real torques, which will be presented in the next section.

## Model-based tracking control

### Path planning

A Delta robot is usually applied to a pick-to-place operation. For a specific 3-D trajectory such as a straight line, a circle, a parabola, and so on, the path can be expressed as a parametric function $q(u)$, where $q$ refers to coordinates $x$, $y$, and $z$ and $u$ is a parameter to define the path from the initial position to the final position. To maintain zero oscillations when the picking and placing operations are performed, one intends to plan a path with zero oscillations at the initial and final positions of the path. Thus, the parameter u can be replaced by a quintic polynomial function of time:

$$u(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \qquad (14)$$

with the boundary conditions as $u(t_0) = u_0$, $\dot{u}(t_0) = 0$, $\ddot{u}(t_0) = 0$, $u(t_f) = u_f$, $\dot{u}(t_f) = 0$, and $\ddot{u}(t_f) = 0$, where $u_0$ and $u_f$ are two given parameters referring to the initial and final positions, and the coefficients $a_i$ can be determined by these boundary conditions.

By differentiating equation (14) with respect to time, the velocity and acceleration of any point on the path can be, respectively, obtained as

$$\dot{q}(u) = \dot{u}\frac{\mathrm{d}q}{\mathrm{d}u},\ \ddot{q}(u) = \ddot{u}\frac{\mathrm{d}q}{\mathrm{d}u} + \dot{u}^2\frac{\mathrm{d}^2 q}{\mathrm{d}u^2} \qquad (15)$$

or

$$\dot{u} = \dot{q}(u)/\frac{\mathrm{d}q}{\mathrm{d}u},\ \ddot{u} = (\ddot{q}(u) - \dot{u}^2\frac{\mathrm{d}^2 q}{\mathrm{d}u^2})/\frac{\mathrm{d}q}{\mathrm{d}u},\ \left(\frac{\mathrm{d}q}{\mathrm{d}u} \neq 0\right) \qquad (16)$$

Note that equation (16) is used to determine the boundary conditions so as to solve equation (14) for the coefficients $a_i$.

For the initial and final positions, substituting those boundary conditions into equation (15) leads to zero velocities, $\dot{q}(u_0)$ and $\dot{q}(u_f)$, as well as the accelerations, $\ddot{q}(u_0)$ and $\ddot{q}(u_f)$. Therefore, the results imply that there are no oscillations at both positions by following the aforementioned path planning approach.

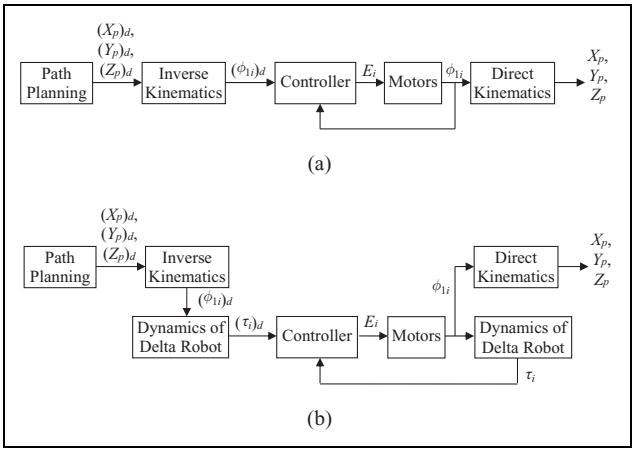


**Figure 3.** Block diagram of (a) a position-based control and (b) a model-based control.

### Tracking control

*Position-based control.* The position-based control converts a robot control problem to several motion control problems, and each motor is achieved by the position tracking control. Based on the aforementioned path planning, the desired coordinates of the end effector can be obtained, and then they can be used to determine the desired active joint angles by applying the inverse kinematics. A controller will be designed to achieve the tracking control of the active joint angles, which are generated from the motors mounted on the active joints, and the active joint angles are considered to be measurable using encoders. The active joint angles generated from the motors can be used to determine the coordinates of the end effector by applying the direct kinematics.

Actually, the entire control system is a semi-closed loop system, and only the dynamics of the motors and the kinematics of the Delta robot are included in the system. Figure 3(a) shows the block diagram of the control system, where $E_i$ are the voltages applied to the motors, and the subscript $d$ refers to reference signals. The first block is the path planning, which can generate the desired trajectories of the end effector expressed as the coordinate functions of time, $(X_p)_d$, $(Y_p)_d$, and $(Z_p)_d$. The second block is the inverse kinematics, which can convert the coordinates $(X_p)_d$, $(Y_p)_d$, and $(Z_p)_d$ to the desired active joint angles $(\varphi_{1i})_d$. The third block is a controller, which computes voltage signals $E_i$ of the motors based on the desired active joint angles $(\varphi_{1i})_d$ and the measured joint angles $\varphi_{1i}$. The last block is the direct kinematics, which can evaluates the actual coordinates $X_p$, $Y_p$, and $Z_p$. Note that the third block consists of a motor and a driver for an experimental work, and the block

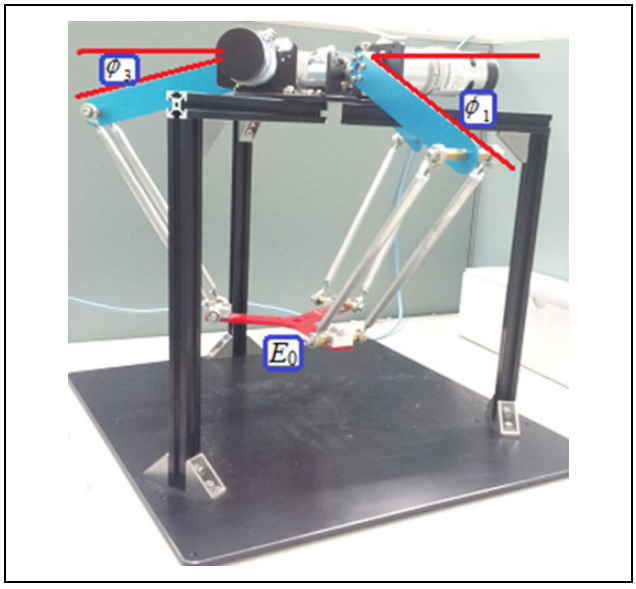


**Figure 4.** An experimental Delta robot.

is modeled as a second-order transfer function with a time delay, which is expressed as

$$G(s) = \frac{K_m e^{-T_{m2}s}}{s(T_{m1}s + 1)} \tag{17}$$

where $K_m$ is a gain, $T_{m1}$ is a time constant, and $T_{m2}$ is delay time. They are model parameters, which will be determined by applying the system identification to the block.

*Model-based control.* The aforementioned control system is broadly applied to various industrial robot manipulators. It can be seen that the dynamics of the Delta robot is not included in the control system, so the dynamic characteristics of the robot will not be shown in the system. To keep the simplicity of the entire system without applying any additional sensors and to incorporate the dynamics into the control system, a model-based loop system is proposed as shown in Figure 3(b). To compare it with Figure 3(a), the proposed control system still uses the encoders to obtain the active joint angles, and there are two additional blocks included in the block diagram, which are the dynamics of the Delta robot. The block converts the desired coordinates $(X_p)_d$, $(Y_p)_d$, and $(Z_p)_d$ to the desired motor torques $(\tau_i)_d$ based on equations (10) to (13). Similarly, the same block converts the actual coordinates $X_p$, $Y_p$, and $Z_p$ to the actual motor torques $\tau_i$ based on the same equations. It is worth noting that a central difference technique is used to calculate $\dot{\varphi}_{1i}$ and $\ddot{\varphi}_{1i}$ before applying the conversion procedures.

## Experimental setup

### Experimental delta robot

An experimental Delta robot is designed and is manufactured as shown in Figure 4. The robot is composed of a moving platform, a fixed base, and three kinematic chains. The moving platform is also called the end effector, which is the member connected by three passive links as shown in Figure 4. The fixed base is the top plane, which is used to fix motors. The kinematic chain consists of an upper arm and a lower arm, where the upper arm is a link connected by a motor, and the lower arm has four links, which form four-bar mechanisms as shown in Figure 4. The three kinematic chains connect the fixed base and the end effector. The actuators drive the three kinematic chains to yield the motion of the end effector. The geometric parameters and the component masses of the robot are shown in Table 1.

**Table 1.** Parameters of the experimental Delta robot.

| Parameters (symbols) | Values (units) |
|---|---|
| Radius of the fixed base ($r_A$) | 90 (mm) |
| Radius of the moving platform ($r_B$) | 105 (mm) |
| Length of the upper links ($L_2$) | 180 (mm) |
| Length of the lower links ($L_1$) | 230 (mm) |
| Mass of the upper links ($m_a$) | 362 (g) |
| Mass of the lower links ($m_b$) | 190 (g) |
| Mass of the moving platform ($m_p$) | 466 (g) |
| Mass of the bearing and the spring ($m_C$) | 34.8 (g) |

The actuators in the robot are DC brushed motors, and each upper arm is driven by a motor mounted on the fixed base. Thus, there are totally three motors on the robot. The type number of the motor is IG-42GM, manufactured by Sha Yang Ye, Inc, Taiwan. The motor also includes a set of planetary gears with the reduction ratio 1/49, which is used to reduce the rotating speed of the motor. According to the specifications provided by the manufacturer, the rated voltage, current, torque, and speed of the motor are 12 V, 5500 mA, 18 kg cm, and 119 r/min, respectively. Besides, each motor is attached by a rotary incremental encoder, which uses two Hall-effector sensors to measure the rotating angle of the motor. Since the motor can generate five pulses each round, the accuracy of the encoder can be evaluated as 0.367°.

### Robot control system

The robot is driven by three motors, and each motor is controlled by a controller, which generates a signal through a control law (e.g. PID controller) processing the encoder signal. Thus, there are three motor controllers on the robot, and the entire robot control system can be treated as a semi-closed-loop system as illustrated in Figure 5. A real-time embedded system *CompactRIO*, manufactured by National Instruments, Inc., USA is used as a controller module in the robot control system. The embedded system has several components as follows:

1. An embedded processor for communication and signal processing (*eRIO* 9022), which is used to process all signals and to generate signals to the motor drivers.

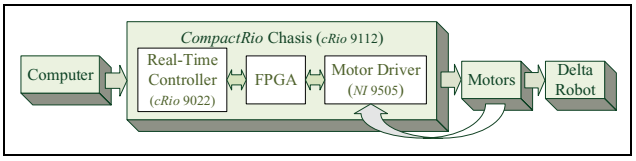


**Figure 5.** Hardware-in-the-loop diagram.

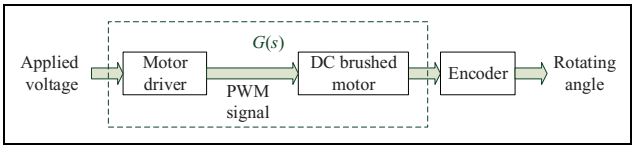


**Figure 6.** Block diagram of a motor and a driver.

2. Three servo motor driver modules (*NI* 9505), which are full H-bridge brushed DC servo drive modules to directly connect the motors and to decode the signals obtained from the encoders.
3. A reconfigurable chassis housing the user-programmable FPGA (*cRIO* 9112), which is a base to connect the processor *eRIO* 9022 and the motor driver module *NI* 9505.
4. A graphical software *LabVIEW* for real-time programming, where the kinematics and dynamics of the robot are programmed through a personal computer.
5. A personal computer, which is used to execute the codes developed by the software *LabVIEW*.

### System identification

To design a controller for the motors, it is necessary to have a mathematical model of the motor, so the system identification will be performed to determine the motor model. Since the motor is driven by a motor driver, the motor and the drive are considered as a system, and the block diagram is shown in Figure 6. The transfer function of the system model is considered as a second-order system with a time delay, which is represented as equation (17). To perform system identification for the system, one generates multiple signals as input signals, which consist of sinusoidal functions with different frequencies and amplitudes. The signals are delivered to the system, and the rotating angles are the output signals and can be measured by the encoders. The system identification of the model is completed using the *MATLAB System Identification Toolbox*, and Table 2 shows the estimated system parameters.

## Experimental and simulation results

This section demonstrates the tracking control using the proposed control approaches presented in "Model-based tracking control" section, where the reference signal of the control loop is a prescribed trajectory of the end effector. First, the PID controller will be used in the system control loop, where the controller parameters will be determined based on the motor models, which are identified in "System identification" section, using the Ziegler–Nichols tuning method and the software *Simulink* PID controller. Secondly, the numerical simulations will be performed. Thirdly, to demonstrate the proposed approaches presented in "Tracking control" section, the path planning is performed based on a quintic polynomial to design a trajectory for the end effector. Finally, the tracking control is completed based on the proposed control approaches and the path planning. The experimental and simulation results are presented, where the experimental work has been introduced in "Experimental Delta robot" section.

**Table 2.** Parameters of the system model consisting of the driver and the motor.

| Motor \ Parameters | $K_m$ (rad/V) | $T_{m1}$ (s) | $T_{m2}$ (s) |
|---|---|---|---|
| Motor 1 | 11.089 | $1.268 \times 10^{-3}$ | $1.143 \times 10^{-2}$ |
| Motor 2 | 10.990 | $1.364 \times 10^{-3}$ | $1.108 \times 10^{-2}$ |
| Motor 3 | 9.768 | $1.000 \times 10^{-3}$ | $1.037 \times 10^{-2}$ |

**Table 3.** PID controller parameters.

| Controls | Position-based control | | | Torque-based control | | |
|---|---|---|---|---|---|---|
| Controller parameters | $K_P$ | $K_I$ | $K_D$ | $K_P$ | $K_I$ | $K_D$ |
| Motor 1 | 0.79302 | 0.04091 | 0.38869 | 0.88009 | 0.10633 | 0.28262 |
| Motor 2 | 0.68921 | 0.03551 | 0.40911 | 0.73113 | 0.18891 | 0.27901 |
| Motor 3 | 0.66412 | 0.02984 | 0.39674 | 0.79409 | 0.09891 | 0.31971 |

### PID controller

The controller law applied to the Delta robot is a PID controller, and the procedure of the controller parameters is illustrated as follows:

1. A step function is selected as a reference command and is applied to the control system for a single motor.
2. The Ziegler–Nichols tuning method is applied to obtain a set of reference parameters. The Ziegler–Nichols tuning method first performs an experiment to reach the critical stability and then applies some empirical formula to obtain the controller parameters.
3. The obtained controller parameters are applied to numerical simulations of the robot based on the identified motor models presented in "System identification" section.
4. The software, *Simulink* PID controller, is utilized to fine-tune the parameter. Since the obtained controller parameters are based on a single-motor experiment, its performance might not be suitable for the robot.

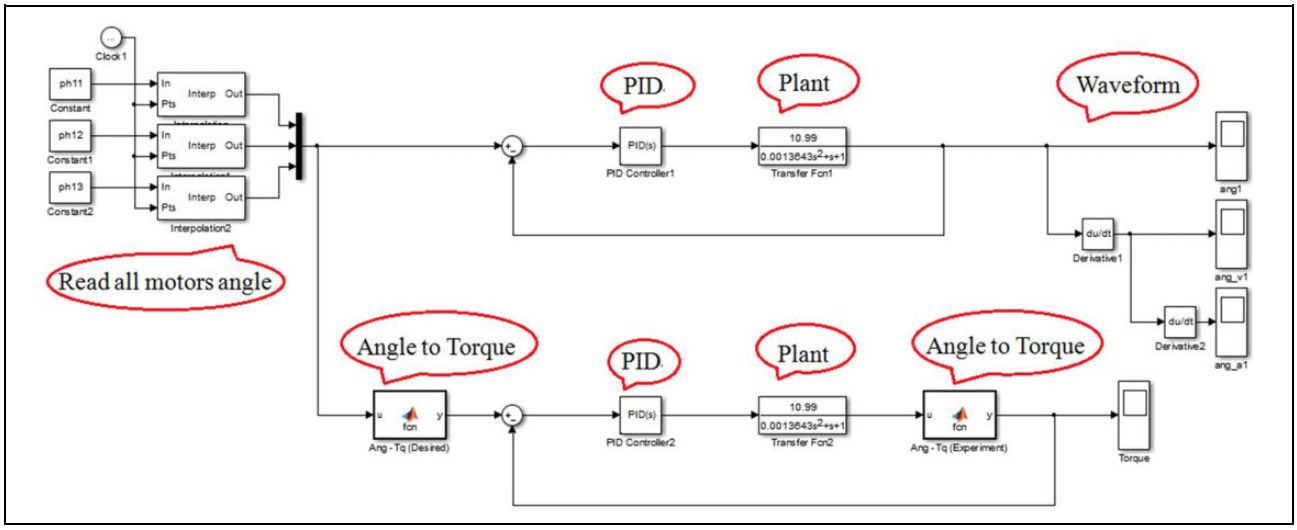


**Figure 7.** Block diagram of performing numerical simulation through *Simulink*.

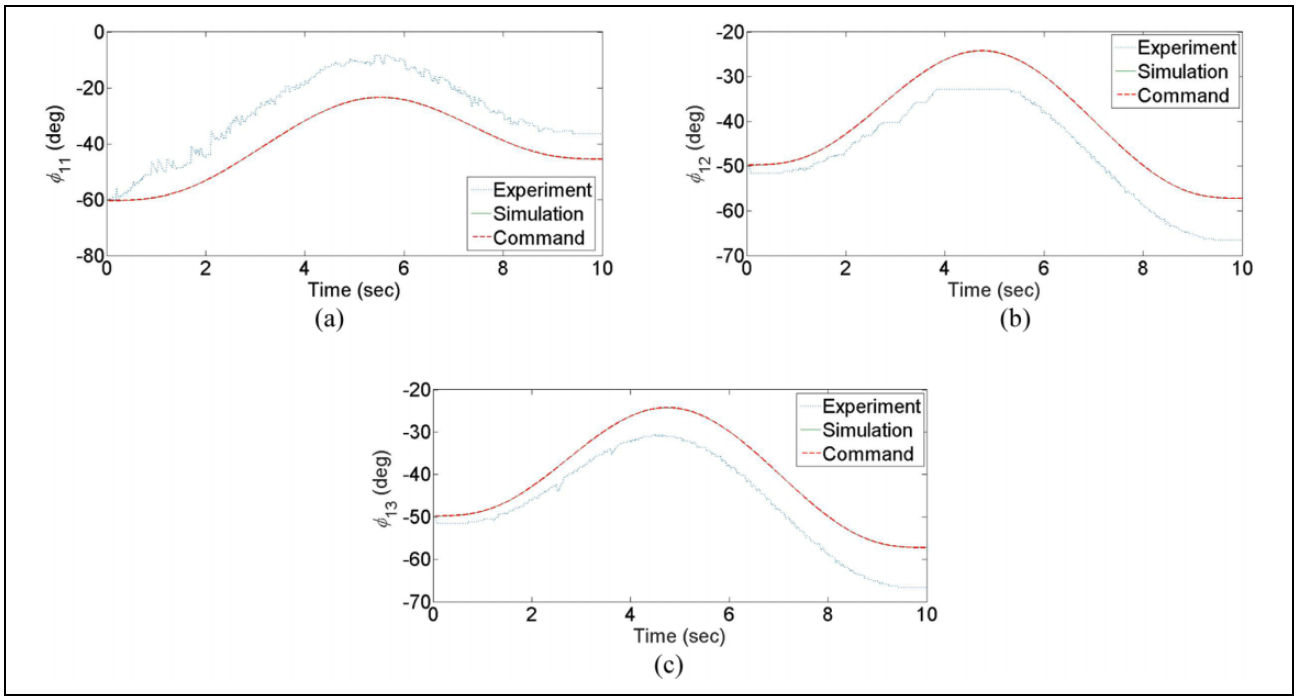


**Figure 8.** Active joint angles (a) $\phi_{11}$, (b) $\phi_{12}$, and (c) $\phi_{13}$ by the position-based control.

5. The procedure is repeated in order to obtain the controller parameters for the other two motors.

Since the input of the controller for the position- and torque-based control systems is different, the PID controller parameters are different, which are listed in Table 3.

## Numerical simulations

One utilizes the software *Simulink* to perform the numerical simulations based on the block diagrams proposed in Figure 3, and the *Simulink* block diagram is shown in Figure 7. Note that the diagram includes both the position- and torque-based control systems. The procedure to perform the simulations is listed as follows:

1. read all command motor angles,
2. simultaneously perform both the position- and torque-based control,
3. deliver the command and output motor angles to the PID controller,
4. achieve the feedback control, and
5. show the output motor angle signals.

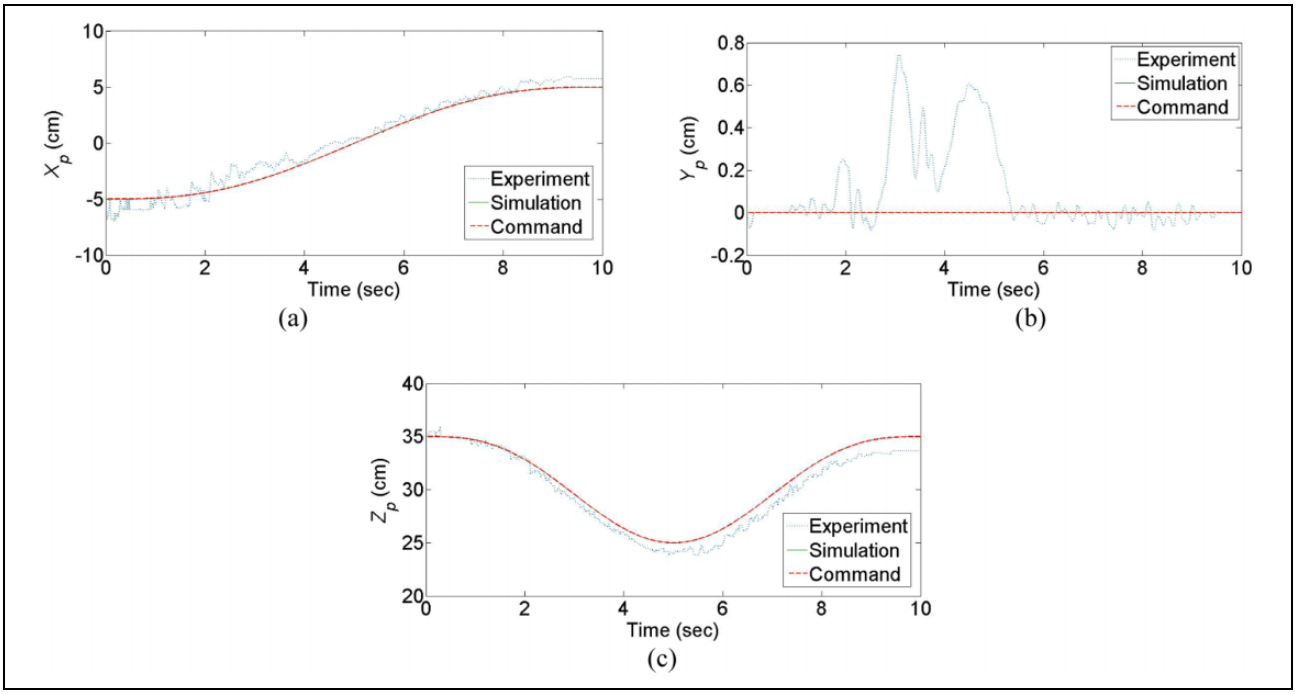


**Figure 9.** Coordinates of the end effector (a) $X_p$, (b) $Y_p$, (c) and $Z_p$ by the position-based control.

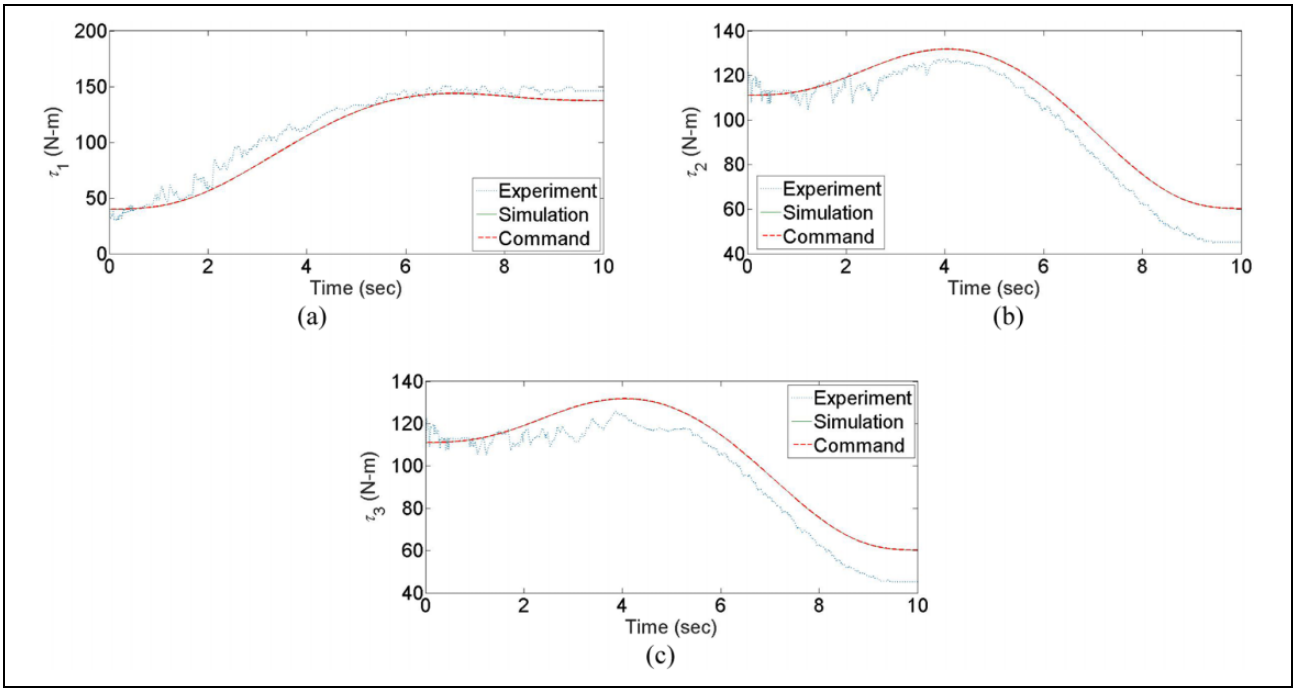


**Figure 10.** Applied torques of (a) motor 1, (b) motor 2, and (c) motor 3 by the position-based control.

The simulation results are combined with the experimental results as shown in Figures 8 to 17.

## Path planning

In this section, one intends to mimic the pick-and-place task of the Delta robot. The coordinates of the initial and final positions are selected as $(X_0, Y_0, Z_0) = (-5, 0, 35)$ and $(X_f, Y_f, Z_f) = (5, 0, 35)$ cm, respectively. In addition, the highest point is specified at $(X_m, Y_m, Z_m) = (0, 0, 25)$ cm. Thus, these three points form a parabolic curve on the $X$–$Z$ plane, which can be expressed as $Z = aX^2 + bX^2 + c$, where $a = 0.4$, $b = 0$, and $c = 25$. To express the equation as parametric forms, one simply defines $X = u$ and

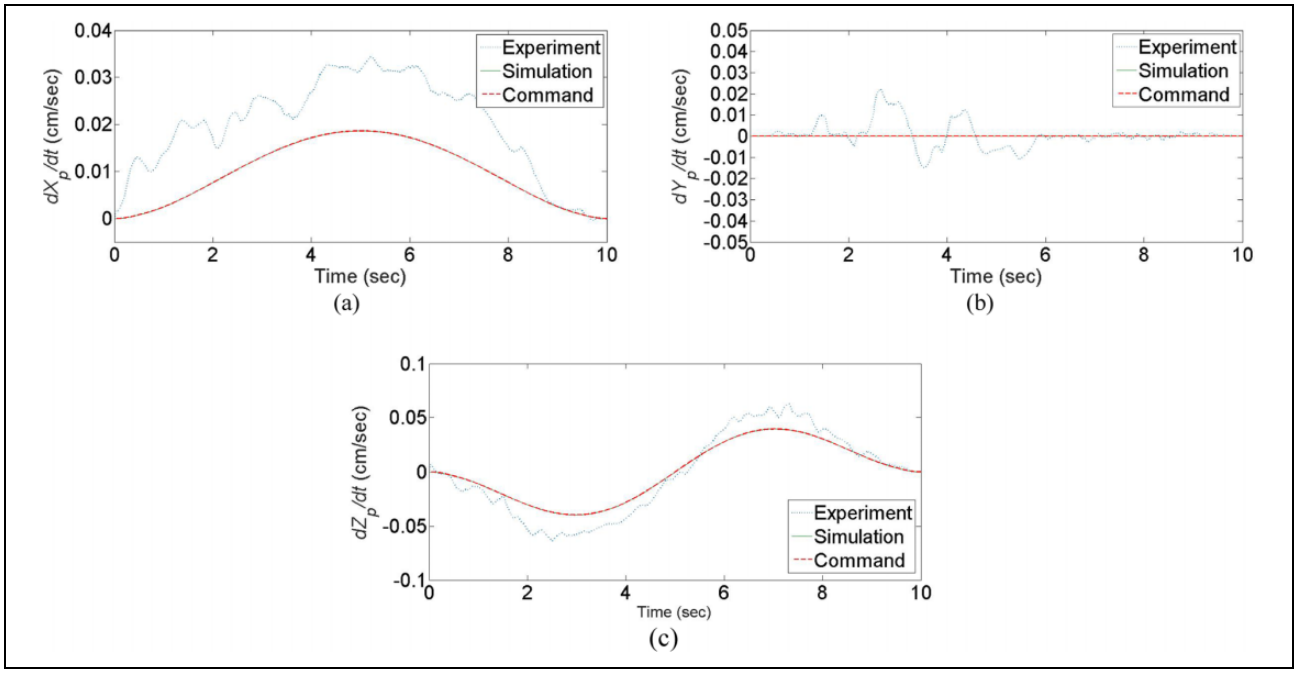


**Figure 11.** Velocities of the end effector (a) $dX_p/dt$, (b) $dY_p/dt$, and (c) $dZ_p/dt$ by the position-based control.

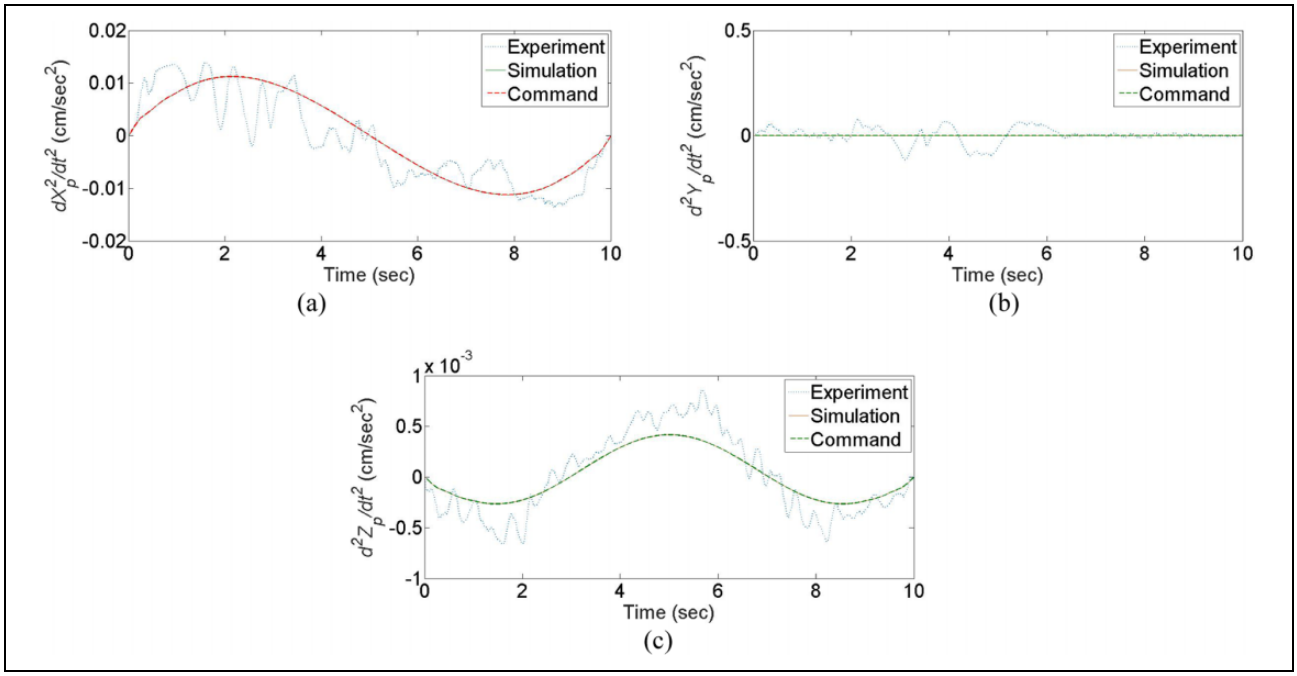


**Figure 12.** Accelerations of the end effector (a) $d^2X_p/dt^2$, (b) $d^2Y_p/dt^2$, and (c) $d^2Z_p/dt^2$ by the position-based control.

$Z = au^2 + bu + c$. Furthermore, to ensure the zero oscillations at the initial and final positions, by following the approach presented in "Path planning" section, the boundary conditions are given as $u(t_0) = X_0$, $\dot{u}(t_0) = 0$, $\ddot{u}(t_0) = 0$, $u(t_f) = X_f$, $\dot{u}(t_f) = 0$, and $\ddot{u}(t_f) = 0$. The initial and final time is set to be $t_0 = 0$ and $t_f = 10$ s, respectively, so the pick-and-place task will be completed in 10 s. Therefore, the coefficients $a_i$ in equation (14) can be determined as $a_i = [-5, 0, 0, 0.1, -0.015, 0.0006]$. Note that the planned path should be ensured within the workspace of the Delta robot and there are no singularities on the path.

## Tacking control

Two control schemes, the position- and model-based controls, are applied to the Delta robot.

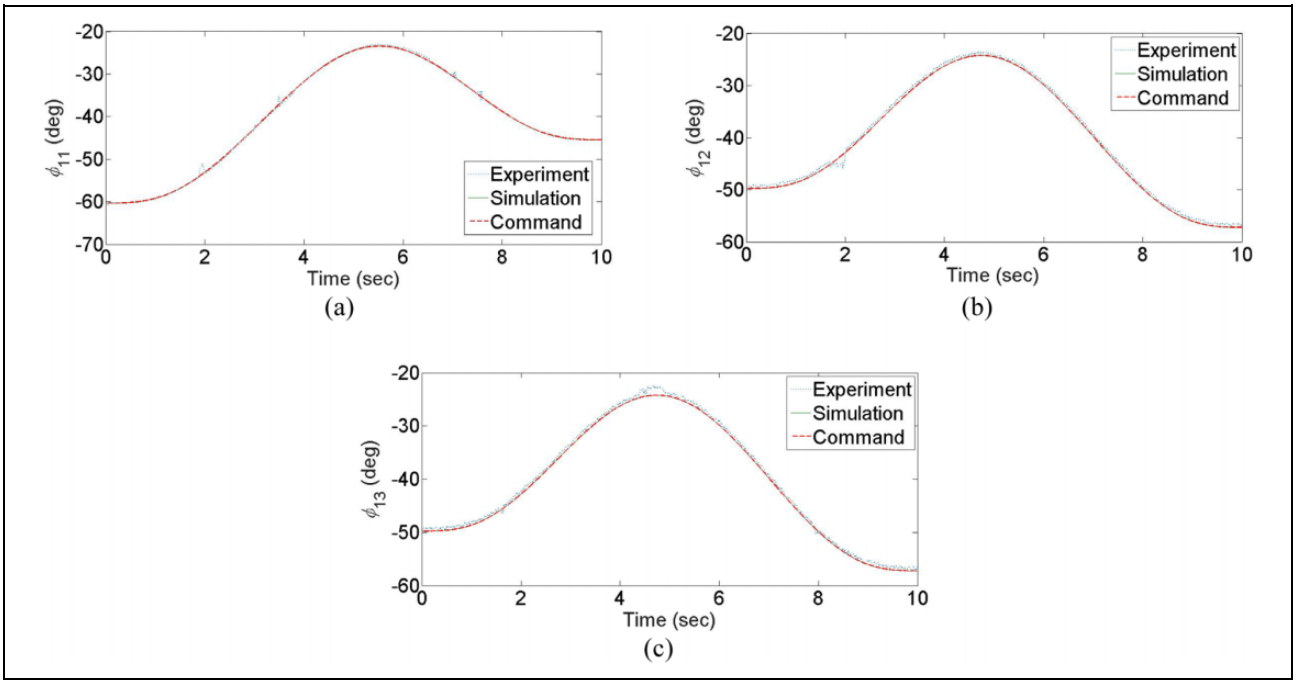


**Figure 13.** Active joint angles (a) $\phi_{11}$, (b) $\phi_{12}$, and (c) $\phi_{13}$ by the torque-based control.

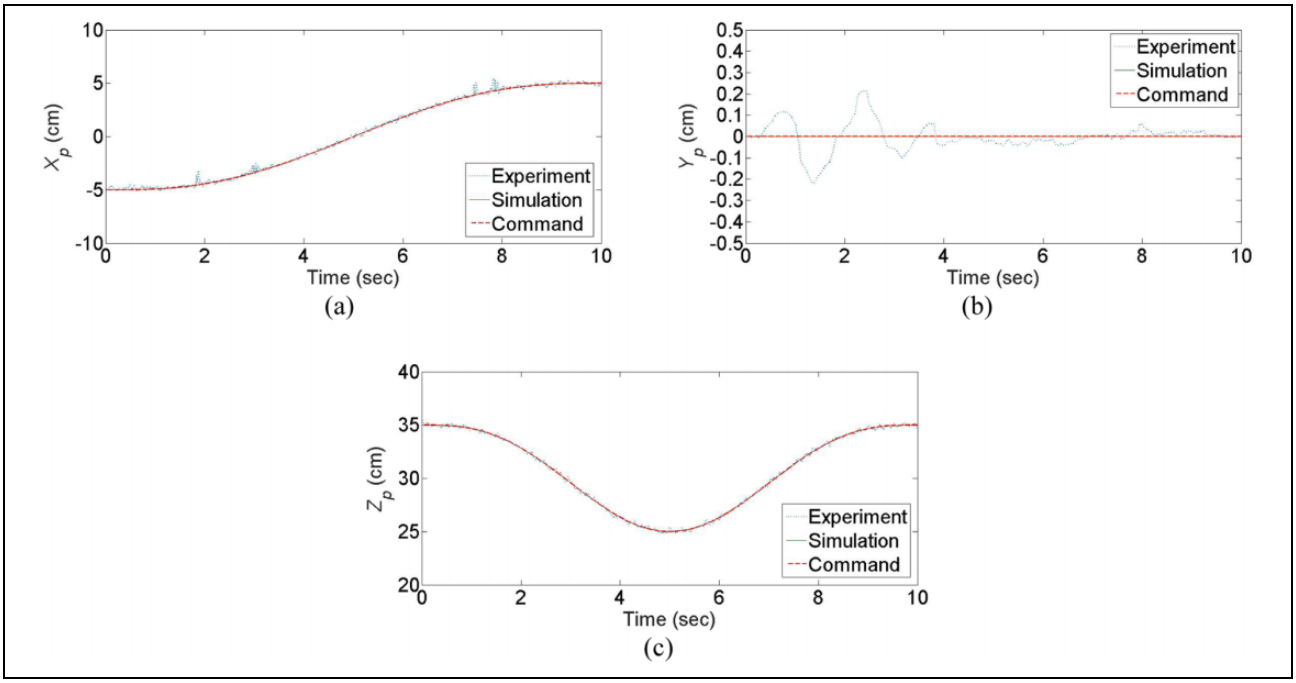


**Figure 14.** Coordinates of the end effector (a) $X_p$, (b) $Y_p$, (c) and $Z_p$ by the torque-based control.

*Position-based control.* This subsection demonstrates the position-based control introduced in "Position-based control" section, and the control is applied to the Delta robot. Figure 8 shows the time responses of the active joint angles for the desired, simulation, and experimental trajectories. The results show that the simulation and desired trajectories are coincident, but the experimental trajectory is much different from them. There are some factors yielding the inaccuracy of the experimental results. First, the initial position might not be very accurate, because there are no additional sensors to ensure the initial position as desired. As observed, there are the errors about 2° at the initial position. Secondly, the dynamics of the Delta robot is not incorporated in the control system, so the inertial and gravitational effects must affect the positioning accuracy. Finally, the gearbox of the motor has backlash, and the accuracy of the encoders has

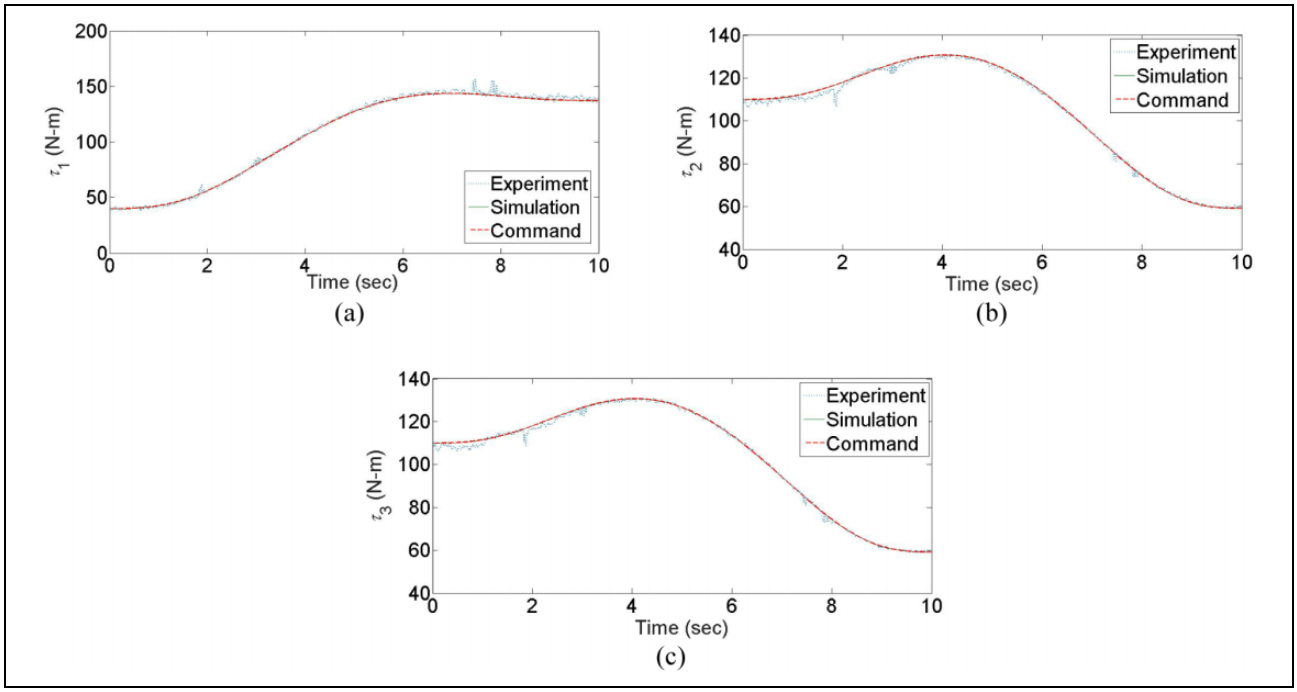


**Figure 15.** Applied torques of (a) motor 1, (b) motor 2, and (c) motor 3 by the torque-based control.

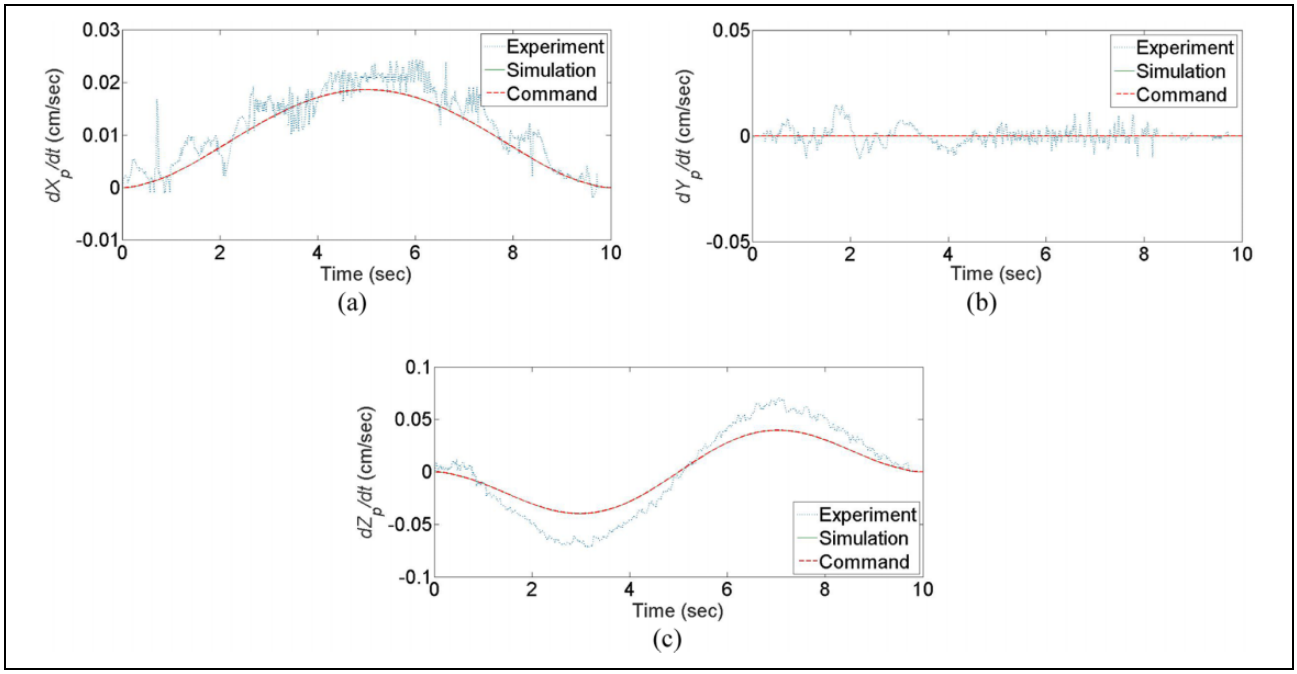


**Figure 16.** Velocities of the end effector (a) $dX_p/dt$, (b) $dY_p/dt$, and (c) $dZ_p/dt$ by the torque-based control.

only 0.367° based on the specifications of the encoders. This causes small oscillations during the entire trajectory. Thus, those factors produce the errors about 10° at the finial position. Figure 9 shows the time responses of the positions of the end effector for the desired, simulation, and experimental trajectories. The position of the end effector is obtained by performing the direct kinematics. Due to the errors of the active joint angles, the errors of the positions are around 2 cm. Figure 10 shows the time responses of the applied torques at the active joint angles. The applied torques are determined by applying the equations of the kinematics and dynamics. Similarly, there are some errors generated from the errors of the active joint angles.

To ensure the zero oscillations at the initial and final positions of the desired path, one applies the first-order central difference method to determine the velocities and

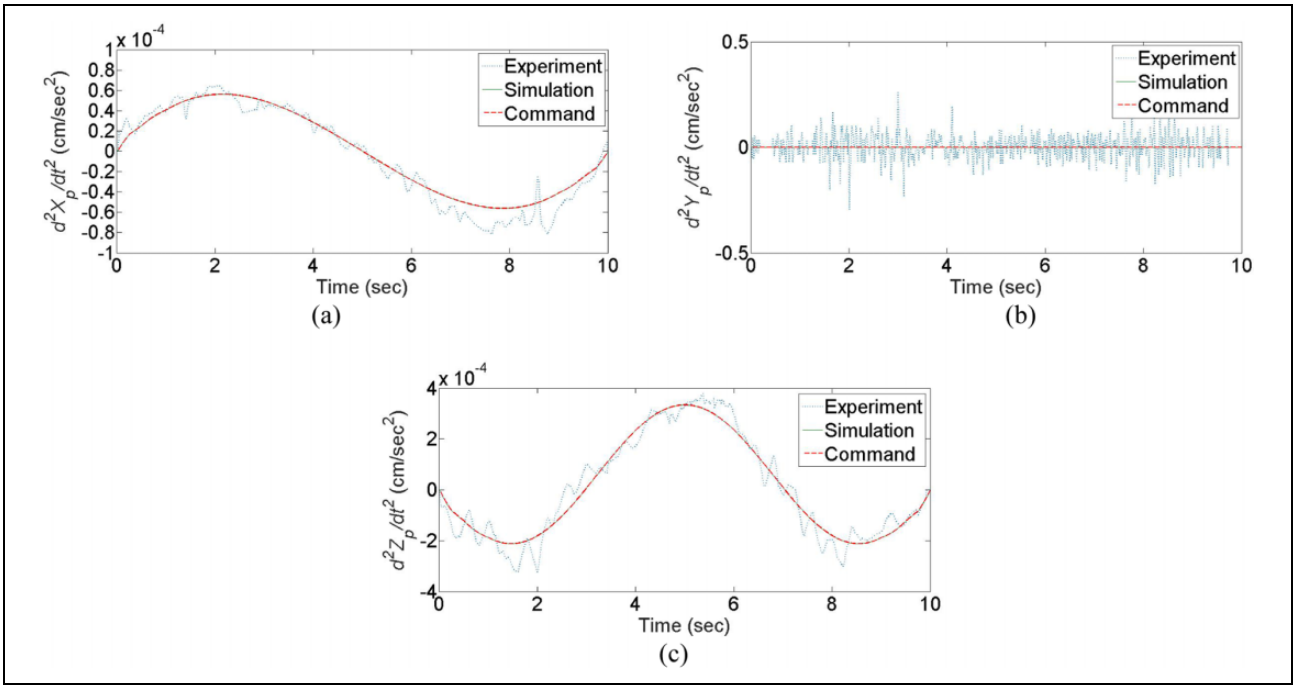


**Figure 17.** Accelerations of the end effector (a) $d^2X_p/dt^2$, (b) $d^2Y_p/dt^2$, and (c) $d^2Z_p/dt^2$ by the torque-based control.

accelerations of the end effector based on the positions. As expected, there will be some errors by applying the numerical method. Figures 11 and 12 show the time responses of the velocities and accelerations of the end effector. The results show that the initial and final velocities and the accelerations are zeros.

*Model-based control.* This subsection demonstrates the model-based control introduced in "Model-based control" section, and the control is also applied to the Delta robot. Figure 13 shows the time responses of the active joint angles for the desired, simulation, and experimental trajectories. The results show that the three trajectories are more coincident to compare with Figure 8. Figure 14 shows the time responses of the positions of the end effector for the desired, simulation, and experimental trajectories. The position of the end effector is obtained by performing the direct kinematics. The results also show that the three trajectories are more consistent to compare with Figure 9. This implies that the included dynamics in the control system can effectively reduce the positioning errors. Figure 15 shows the time responses of the applied torques at the active joint angles. The applied torques are determined by applying the equations of the kinematics and dynamics. Furthermore, Figures 16 and 17 show the time responses of the velocities and accelerations of the end effector. The results show that the initial and final velocities and the accelerations are zeros as expected.

## Discussion

To show the performance of the model-based control scheme over the conventional position-based control scheme, this section demonstrates an example, which shows both the numerical and experimental results obtained by both schemes. The example mimics the pick-and-place task of a Delta robot by designing a quintic polynomial with zero velocities and accelerations at the starting and end points. The results by applying the position-based control show that the maximum errors of the active joints are around 10°. The errors also lead to the maximum positioning errors of the end effector with around 3 cm. In contrast, the model-based control has the maximum errors of the active joints and the end effector with around 3° and 0.5 cm, respectively. Since both the control schemes are applied to the same experimental equipment, one can deduce that the model-based control has a better positioning accuracy.

## Conclusions

This article presents a simple model-based control scheme applied to a Delta robot, which has three translating DOFs and is driven by three rotating motors. Conventionally, the position-based control scheme is mostly applied to positioning control of robot manipulators. This control scheme converts the robot control problem to several separated motor control problems. The major disadvantage of the control scheme might not provide a good positioning accuracy due to the lack of the robot's dynamics in this control loop. In literature, one of the model-based controls is the computed torque control, which computes applied torques based on the inverse dynamics of robots and the errors of joint angles as well as angular velocities. The proposed model-based control scheme also computes the applied

torques based on both the kinematics and dynamics of robots, but the torques are utilized to calculate the errors between them and the desired torques, which is used to fulfill the feedback control. Thus, the proposed scheme is much simpler and can be treated as an indirect torque control without using torque and current sensors. This article demonstrates the simulation and experimental results by the mode-based control scheme, and the results are compared with those by the position-based control scheme. The results show that the model-based control scheme provides better positioning accuracy.

## Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

## References

1. Maya M, Castillo E, Lomelí A, et al. Workspace and payload-capacity of a new reconfigurable delta parallel robot. *Int J Adv Robot Syst* 2013; 10(1): 56.
2. Lin J, Luo CH and Lin KH. Design and implementation of a new DELTA parallel robot in robotics competitions. *Int J Adv Robot Syst* 2015; 12(10): 1–10.
3. Castañeda LA, Luviano-Juarez A and Chairez I. Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control. *IEEE Trans Control Syst Technol* 2015; 23(4): 1387–1398.
4. Ramírez-Neria M, Sira-Ramírez H, Luviano-Juárez A, et al. Active disturbance rejection control applied to a delta parallel robot in trajectory tracking tasks. *Asian J Control* 2015; 17(2): 636–647.
5. Guglielmetti P and Longchamp R. Task space control of the delta parallel robot. *Motion Control Int Autom* 2014; 337–342.
6. Lu XG, Liu M and Liu JX. Design and optimization of interval type-2 fuzzy logic controller for delta parallel robot trajectory control. *Int J Fuzzy Syst* 2017; 19: 1–17.
7. Dumlu A and Erenturk K. Trajectory tracking control for a 3-DOF parallel manipulator using fractional-order control. *IEEE Trans Indus Elect* 2014; 61(7): 3417–3426.
8. Lin HH, Ta YH and Liu CS. The implementation of smoothing robust control for a delta robot. In: *2013 Second international conference on robot, vision and signal processing (RVSP)*, Kitakyushu, Japan, 10–12 December 2013, pp. 208–213. IEEE.
9. Zhang GY, Liu GF, Guo XB, et al. Dynamic conveyor tracking control of a delta robot. In: Erian A. Armanios (ed) *Key Engineering Materials*, Vol. 679, Trans Tech Publications, 2016, pp. 43–48. Zurich, Switzerland: Trans Tech Publications.
10. Rachedi M, Bouri M and Hemici B. H$\infty$ feedback control for Parallel mechanism and application to delta robot. In: *2014 22nd Mediterranean conference of control and automation (MED)*, Palermo - ITALY, 16–19 June 2014, pp. 1476–1481.
11. Li Q and Ouyang PR. Integrated design and PD control of high-speed closed-loop mechanisms. *J Dyn Syst Meas Control* 2002; 124: 522–528.
12. Ghorbel FH, Chételat O, Gunawardana R, et al. Modeling and set point control of closed-chain mechanisms: theory and experiment. *IEEE Trans Control Syst Technol* 2000; 8(5): 801–815.
13. Ouyang PR, Zhang WJ and Wu FX. Nonlinear PD control for trajectory tracking with consideration of the design for control methodology. In: *2002 Proceedings ICRA'02 IEEE international conference on robotics and automation*, Vol. 4, Washington, D.C., USA, 11-15 May 2002, pp. 4126–4131.
14. Diaz-Rodriguez M, Valera A, Mata V, et al. Model-based control of a 3-DOF parallel robot based on identified relevant parameters. *IEEE/ASME Trans Mech* 2013; 18(6): 1737–1744.
15. Zhang M, Sheng B, Davies TC, et al. Model based open-loop posture control of a parallel ankle assessment and rehabilitation robot. In: *2015 IEEE international conference on advanced intelligent mechatronics (AIM)*, Busan, Korea, 7–11 July 2015, pp. 1241–1246.
16. Meng Q, Zhang T and Song JY. High-efficiency model-based control of 6-DOF parallel robot using position measurement only. In: *2013 25th Chinese control and decision conference (CCDC)*, Guiyang, China, 25–27 May 2013, pp. 2612–2615.
17. Gao SH, Fan R and Wang D. Dynamic modeling and model-based force control of a 3-DOF translational parallel robot. In: Xiao Zhi Hu and Alan Kin Tak Lau (eds) *Advanced Materials Research*, Vol. 1006, Trans Tech Publications, 2014, pp. 609–617. Zurich, Switzerland: Trans Tech Publications.
18. Zhang G, Wu J, Liu P, et al. Dynamic analysis and model-based feedforward control of a 2-DOF translational parallel manipulator driven by linear motors. *Int J Indus Robot* 2013; 40(6): 597–609.
19. Codourey A. Dynamic modeling of parallel robots for computed-torque control implementation. *Int J Robot Res* 1998; 17(12): 1325–1336.
20. Codourey A. Dynamic modelling and mass matrix evaluation of the DELTA parallel robot for axes decoupling control. *Proc IEEE/RSJ Int Conf* 1996; 3: 1211–1218.
21. Yang Z, Wu J and Mei J. Motor-mechanism dynamic model based neural network optimized computed torque control of a high speed parallel manipulator. *Mechatronics* 2007; 17(7): 381–390.
22. Shang W and Cong S. Nonlinear computed torque control for a high-speed planar parallel manipulator. *Mechatronics* 2009; 19(6): 987–992.
23. Shang WW, Cong S and Ge Y. Adaptive computed torque control for a parallel manipulator with redundant actuation. *Robotica* 2012; 30(03): 457–466.
24. Masarati P. Computed torque control of redundant manipulators using general-purpose software in real-time. *Multi Syst Dyn* 2014; 32(4): 403–428.
25. Tsai LW. *Robot analysis: The mechanics of serial and parallel manipulators*. New York, USA: John Wiley & Sons, 1999.
26. López M, Castillo E, García G, et al. Delta robot: inverse, direct, and intermediate Jacobians. *Proc Inst Mech Eng C J Mech Eng Sci* 2006; 220(1): 103–109.