Check for updates

Parallel Image-Based Visual Servoing/Force Control of a Collaborative Delta Robot

Minglei Zhu[1], Cong Huang[1], Zhiqiang Qiu[2], Wei Zheng[2] and Dawei Gong[1]*

[1] School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China, [2] Science and Technology on Thermal Energy and Power Laboratory, Wuhan Second Ship Design and Research Institute, Wuhan, China

In this paper, a parallel Image-based visual servoing/force controller is developed in order to solve the interaction problem between the collaborative robot and the environment so that the robot can track the position trajectory and the desired force at the same time. This control methodology is based on the image-based visual servoing (IBVS) dynamic computed torque control and couples the force control feedback in parallel. Simulations are performed on a collaborative Delta robot and two types of image features are tested to determine which one is better for this parallel IBVS/force controller. The results show the efficiency of this controller.

Keywords: image-based visual servoing/force control, collaborative robot, Delta robot, trajectory tracking, image moment visual servoing

1. INTRODUCTION

Recently, a large number of research studies focus on the study of collaborative robots in the domains of mechanical, sensing, planning, and control issues in order to improve the safety and dependability of robotic systems (Magrini and De Luca, 2016). When collaborative robots are performing tasks, such as picking and placing of objects, deburring, polishing, spraying, and high precision positioning assembly (Yang et al., 2008; Wu et al., 2017; Xu et al., 2017), the control of contact forces and the position are all necessary. When collaborative robots are interacted with the environment, the contact forces exist and should be controlled properly, otherwise, this force may damage the robot manipulator or the human operating the robot.

Compared with serial robots, parallel robots show better performance in terms of high speeds and accelerations, payload, stability, stiffness, and accuracy. Due to these advantages, parallel robots is widely applied as collaborative robots (Zabihifar and Yuschenko, 2018; Jeanneau et al., 2020; Fu et al., 2021). However, the force/position control of parallel robots has been rarely addressed since the structure of parallel robots is complex and the relationship between the input and output of the controller is highly non-linear (Merlet, 2006; Dai et al., 2021). When parallel robots are controlled with a classical model-based computed torque controller, the position accuracy relies on the model precision. Nevertheless, even a detailed robot model still can not guarantee high accuracy because of the errors from manufacturing, assembly, and external disturbances in practice. To overcome this problem, we need to find an alternative way to bypass the complex parallel robot architectures. Visual servoing is a type of an external sensor-based controller which takes one or several cameras as external sensors and closes the control loop with the vision

feedback (Chaumette and Hutchinson, 2006; Chaumette et al., 2016; Fu et al., 2021; Zhu et al., 2022). With this controller, the pose of the robot can be directly estimated from the camera and we can bypass the complex calculation of the direct kinematic model. The visual servoing can be divided into three categories: image-based visual servoing (IBVS), position-based visual servoing (PBVS), and 2.5D visual servoing. Image-based visual servoing regulates the error directly in the image plane, and the position-based visual servoing takes the pose of the camera with respect to some reference coordinates frame to define the feature to be controlled and the entire control algorithm is realized in the Cartesian space. 2.5D visual servoing is a combination of IBVS and PBVS. Compared with PBVS and 2.5D visuals servoing, image-based visual servoing is realized in the image plane, so that it will not cause the loss of tracking target during the servoing. In addition, IBVS is more robust to the camera calibration errors and the noise from the camera observation.

Force control is important in the process when collaborative robots are operating tasks requiring interaction with the environment, such as contour-following operations, assembly of mechanical components, and the use of any mechanical tool. Several types of force control schemes have been proposed and they can be divided into two main categories: direct force control and indirect force control (Siciliano and Villani, 1999). Indirect force control includes the admittance control (Mason et al., 1981) and impedance control (Hogan, 1985). These two control methods are based on creating a model between the interaction forces and the pose of the manipulator and replacing the direct control of the contact force with the position control of the robot manipulator. The parameters of the impedance model or the admittance model, such as the stiffness, damping, and inertia, should be aware in advance and the performance of the controller is affected by the accuracy of these parameters. Direct force control includes the hybrid position/force control (Raibert and Craig, 1981) and parallel position/force control (Chiaverini and Sciavicco, 1993). The hybrid position/force controller is designed with two individual controllers: position controller and force controller. They are developed for each decoupled subspace and connected with a selection matrix. When hybrid position/force control is applied, the geometric description of the interaction environment ought to be available in advance. In addition, for one dedicated DOF, we cannot control the contact force and the position simultaneously. The parallel position/force controller combines the position control loop and the force control loop in parallel. It can reach the goal of controlling position and force at the same time (Chiaverini and Sciavicco, 1993) without the selection matrix. The dynamic parameters of the interaction model and the geometric description of the environment are not indispensable when we apply this method.

Therefore, in the current research, a novel controller: a parallel IBVS/force controller is created. The IBVS dynamic controller is applied to perform the position control and the force feedback is coupled in parallel. Two controlled variables, contact forces and the image features, are combined together to reach the goal of controlling the force and position directly in the image space at the same time. Considering the fact that the accuracy of position

control directly affects the precision of contact force control, in this paper, two types of image features, normalized image points and image moments (Tahri and Chaumette, 2005; Dallej et al., 2006; Andreff et al., 2007), are selected as the vision feedback, and the comparison of controller performance will be tested in order to determine which one is better for the parallel IBVS/force control of parallel robots.

To the best of our knowledge, this is the first time that the parallel IBVS/force control is proposed and applied to the control of a collaborative Delta robot. The compensation of the robot vision dynamic non-linearities are considered and both position and force can be controlled in all directions. In addition, two types of IBVS with different image features, normalized image points and image moments, are tested to determine which one is more suitable for this controller.

This paper is organized as follows. In Section 2, the IBVS dynamic model is created and a simple recall on the normalized image point visual servoing and image moment visual servoing are given. In Section 3, the parallel IBVS/force control method is proposed and presented in detail, in addition, the stability analysis is given. Then, in Section 4, the way of creating the dynamic model of the Delta robot is presented. Simulations and the analysis of results are given in Section 5. In the end, we draw the conclusion in Section 6.

2. IMAGE-BASED VISUAL SERVOING DYNAMIC MODEL

For image-based visual servoing, the vector of image features obtained from the camera observation is assigned to $\mathbf{s}$. With the well-known interaction matrix $\mathbf{L}_a$ (Chaumette and Hutchinson, 2006), the IBVS kinematic model can be written as follows:

$$\dot{\mathbf{s}} = \mathbf{L}_a \boldsymbol{\tau} \qquad (1)$$

where $\boldsymbol{\tau}$ is the camera-object kinematics screw.

When the dynamics of robots are considered, the first order of visual servoing is no longer sufficient because the velocity of the image features could not correspond to the acceleration of the robot. Then, we needed to find the relationship between the image features acceleration and the robot kinematics, and dynamics with a second-order model so that the computed torque controllers can be obtained directly in image space. This second-order visual servoing interaction model can be obtained by differentiation of Equation (1):

$$\ddot{\mathbf{s}} = \mathbf{L}_a \frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\tau} + \dot{\mathbf{L}}_a \boldsymbol{\tau} \qquad (2)$$

In addition, the time derivation of the kinematic screw is as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{\tau} = \mathbf{a} + \begin{bmatrix} v \times w \\ 0 \end{bmatrix} \qquad (3)$$

where $\mathbf{a}$ is the spatial relative camera-object acceleration screw. Equation (3) is obtained in the condition that the camera frame is not inertial with respect to the observed object.
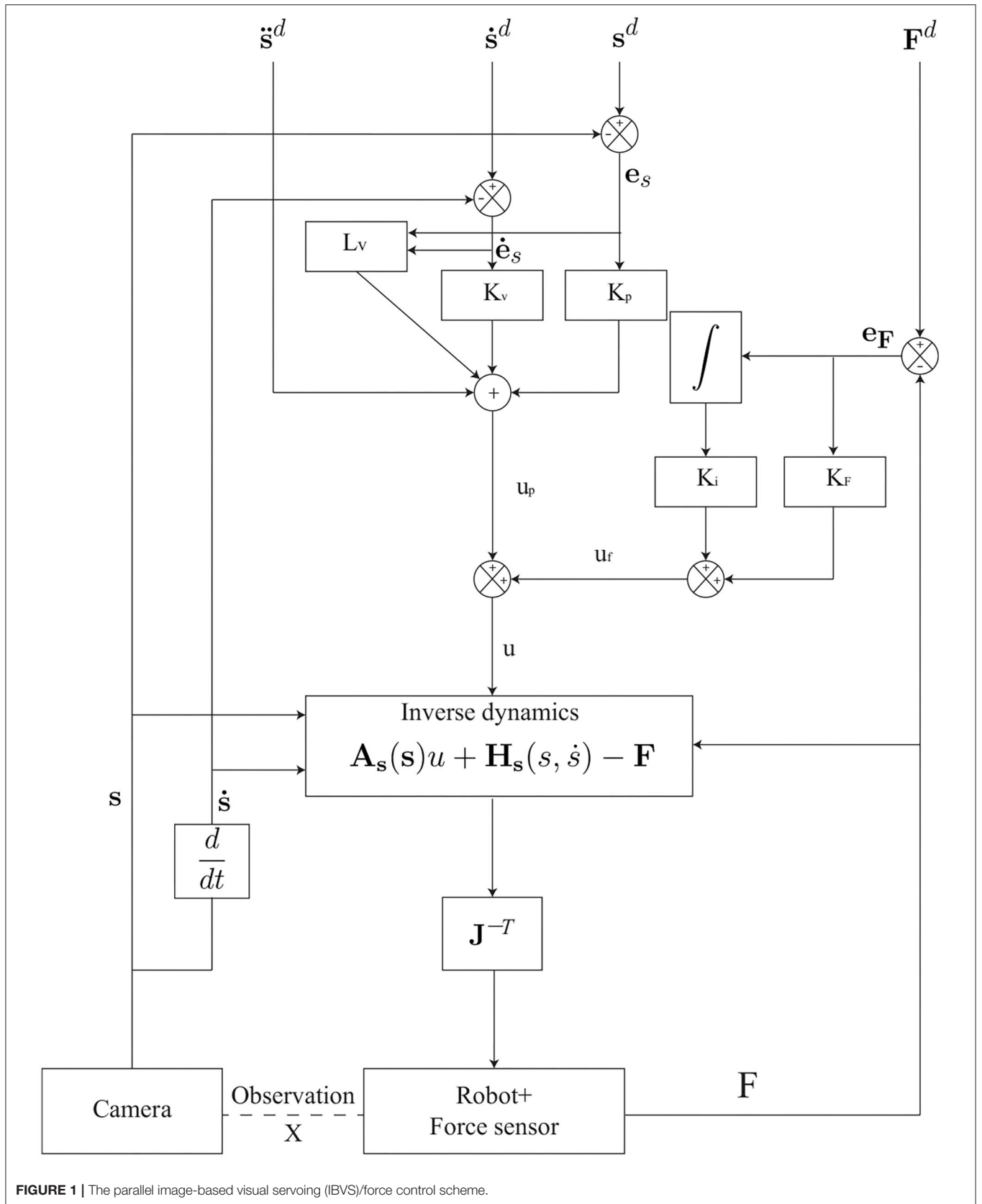
FIGURE 1 | The parallel image-based visual servoing (IBVS)/force control scheme.
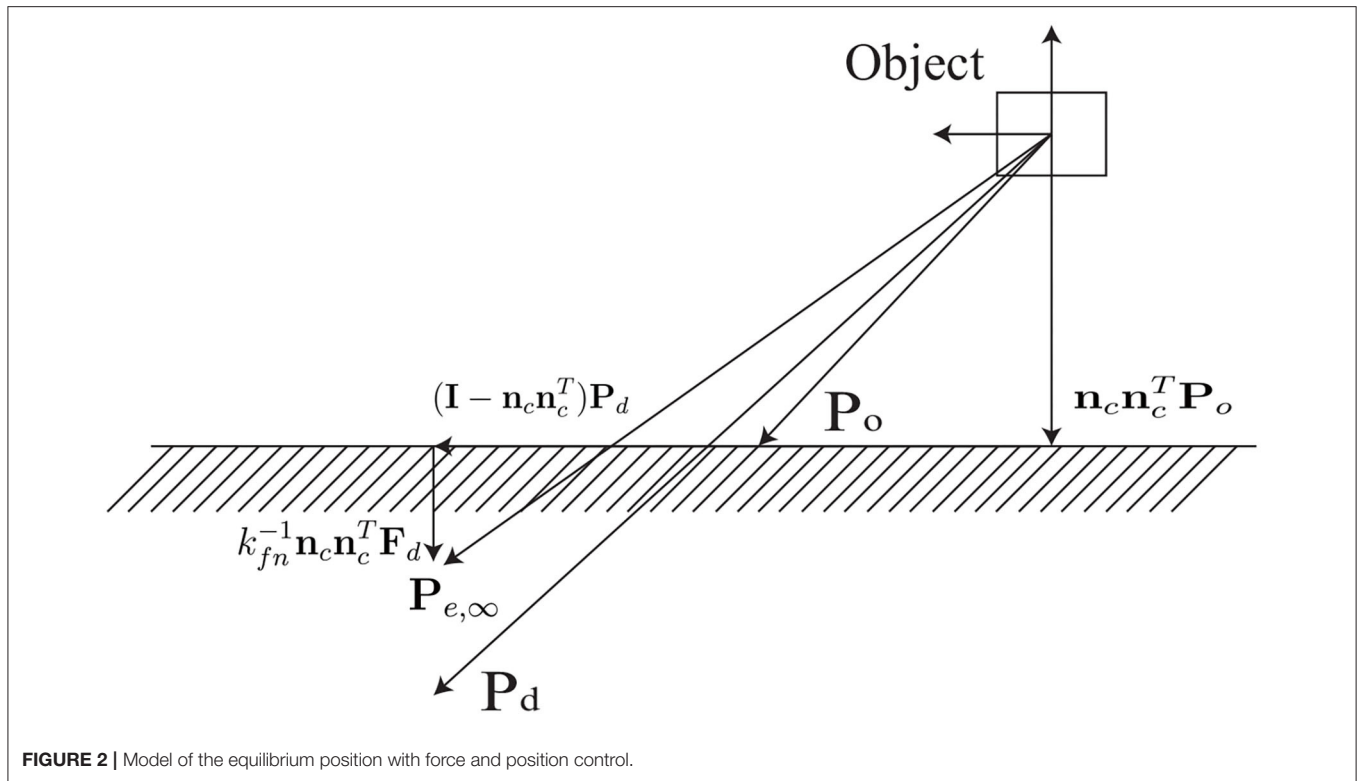
FIGURE 2 | Model of the equilibrium position with force and position control.

Then (2) can be transformed to

$$\ddot{\mathbf{s}} = \mathbf{L}_s \mathbf{a} + \mathbf{L}_s \begin{bmatrix} v \times w \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}^T \mathbf{H}_1 \\ \cdots \\ \boldsymbol{\tau}^T \mathbf{H}_n \end{bmatrix} \boldsymbol{\tau} \qquad (4)$$

where $\mathbf{H}_i \doteq (\frac{\partial l_i}{\partial \mathbf{s}})\mathbf{L}_s + (\frac{\partial l_i}{\partial z})\mathbf{L}_z$, and the depth z between the camera and the object has the following property $\dot{z} = \mathbf{L}_z \boldsymbol{\tau}$.

After some simple manipulations, we can rewrite Equation (2) as follows:

$$\ddot{\mathbf{s}} = \mathbf{L}_a \mathbf{a} + \mathbf{L}_v \qquad (5)$$

$\mathbf{L}_v$ corresponds to the differentiation of the interaction matrix and the Coriolis acceleration. It can be written in the form:

$$\mathbf{L}_v = \begin{bmatrix} \boldsymbol{\tau}^T \Omega_x \boldsymbol{\tau} \\ \boldsymbol{\tau}^T \Omega_y \boldsymbol{\tau} \end{bmatrix} \qquad (6)$$

$\mathbf{L}_v$ is a function of $\mathbf{s}$, $\boldsymbol{\tau}$, and the depth $\mathbf{Z}$ between the object and the camera and the expression of Ω_x and Ω_y can be found in Mohebbi (2012).

The most common image feature of visual servoing is the normalized image points. For a given 3D point P in space and its coordinates $[X_p, Y_p, Z_p]^T$ with respect to the camera frame, we can get the normalized coordinate $\mathbf{s} = [x, y]^T = [X/Z, Y/Z]^T$. Z is the so-called vision depth between the object and the camera. The interaction matrix related to

the image point feature $\mathbf{s}$, $\mathbf{L}_s$ takes the following well-known form (Chaumette and Hutchinson, 2006):

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -x^2 - 1 & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & y^2 + 1 & -xy & -x \end{bmatrix} \qquad (7)$$

while the interaction matrix of a set of normalized image points can be obtained by vertically stacking the individual matrices.

With the development of image processing technology, another image feature can be applied and has been proven to be effective in image-based visual servoing is planar image moments (Chaumette, 2004). The target $\mathcal{T}$ to be observed can be a dense object defined by a set of closed contours or a discrete set of image points (Tahri and Chaumette, 2005). The 2D image moments m_{ij} of order $i + j$ are defined by:

$$m_{ij} = \iint_{\mathcal{T}} x^i y^i dx dy \qquad (8)$$

where x and y are the coordinates in the camera frame of any point M belonging to the object $\mathcal{T}$. Several independent image moments can be calculated with this definition: the coordinates x_g, y_g of the center of gravity of the target, the area a of the target, the orientation α, and the invariant image moments c_1, c_2 (see definition in Tahri and Chaumette, 2005). The expression of the interaction matrix $\mathbf{L}_{m_{ij}}$ of any image moment of order $i+j$, m_{ij} has been provided in Chaumette (2004). The differentiation of the interaction matrix and the Coriolis acceleration term $\mathbf{L}_v$ related to the image moment m_{ij} can be found in Fusco (2020).
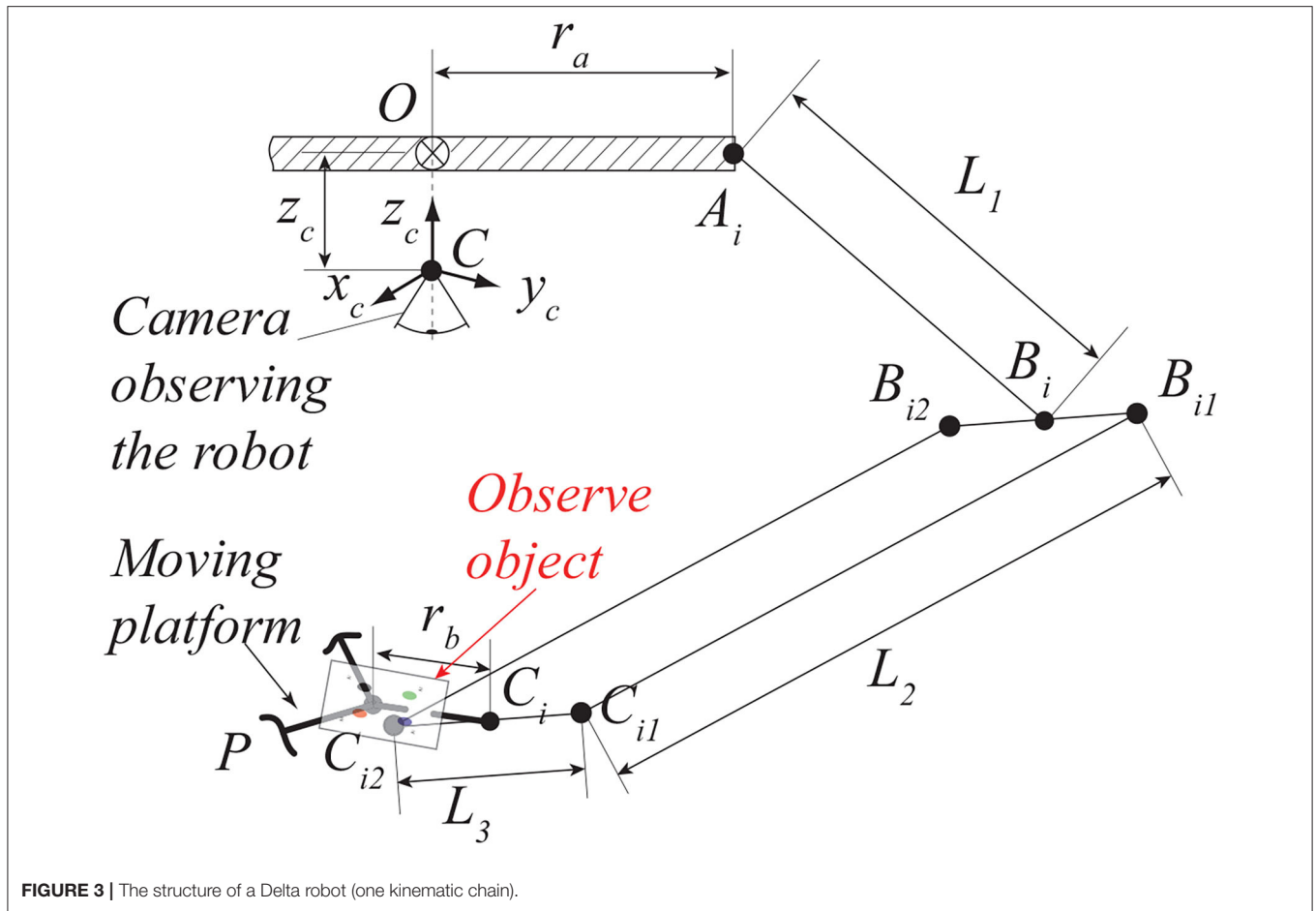
FIGURE 3 | The structure of a Delta robot (one kinematic chain).

3. DEVELOPMENT OF THE PARALLEL IBVS/FORCE CONTROLLER

3.1. Design of the Parallel IBVS/Force Controller

The requirements of restricted motion tasks, which comprise simultaneous motion and force control, cannot be met by classical computed torque control applied in joint space. Therefore, we consider the general dynamic model of robots expressed in the task space.

$$\mathbf{M}_\chi(\chi)\ddot{\chi} + \mathbf{C}_\chi(\chi, \dot{\chi}) + \mathbf{G}_\chi(\chi) + \mathbf{F} + \mathbf{J}^T(\chi)f = \boldsymbol{\tau}_\chi \quad (9)$$

where χ is the set of Cartesian coordinates of the robots. $\mathbf{M}_\chi$ is the inertia matrix in task space, $\mathbf{C}_\chi(\chi, \dot{\chi})$ is the term Coriolis and centrifugal. $\mathbf{G}_\chi(\chi)$ represents the gravity, and f is the vector of friction forces. $\mathbf{F}$ is the contact force caused by the robot manipulator and the environment. $\boldsymbol{\tau}$ is the projection of the generalized torques at the joints in the task frame. $\mathbf{J}$ is the Jacobian matrix of the robot.

In the condition that the parameters of the robot model are perfectly known, then the parallel position/force control law can be written as follows:

$$\boldsymbol{\tau}_\chi = \hat{\mathbf{M}}_\chi(\chi)\boldsymbol{u} + \hat{\mathbf{C}}_\chi(\chi, \dot{\chi}) + \hat{\mathbf{G}}_\chi(\chi) + \hat{\mathbf{F}} + \mathbf{J}^T(\chi)\hat{f} \quad (10)$$

where $\hat{\ }$ represents the estimated value. We suppose that the position and force control loops are, respectively, a linear PD and PI controller, then we have:

$$\begin{aligned}
\boldsymbol{u} &= \boldsymbol{u}_p + \boldsymbol{u}_f \\
\boldsymbol{u}_p &= \ddot{\chi}^d + \mathbf{K}_v(\dot{\chi}^d - \dot{\chi}) + \mathbf{K}_p(\chi^d - \chi) \\
\boldsymbol{u}_f &= \mathbf{K}_f(\mathbf{F}^d - \mathbf{F}) + \mathbf{K}_i \int_0^t (\mathbf{F}^d - \mathbf{F})d\tau
\end{aligned} \quad (11)$$

where $()^d$ represents the desired value. $\mathbf{K}_v$, $\mathbf{K}_p$, $\mathbf{K}_f$, and $\mathbf{K}_i$ are positive diagonal matrices. From this control law, we see that because of the integral item, the position, and force in all degrees of freedom (DOF) with the force control loop taking precedence over the position control loop is considered.

The drawback of applying the model-based position controller (with joint sensor) has been detailed and presented. Moreover, the combination of a sensor-based force control loop and a model-based control loop leads to a control architecture that is not homogeneous. Then, we replace the model-based position controller with the image-based visual serving controller. The resultant control law applied to the robots in task space can be written as follows and the control scheme is illustrated
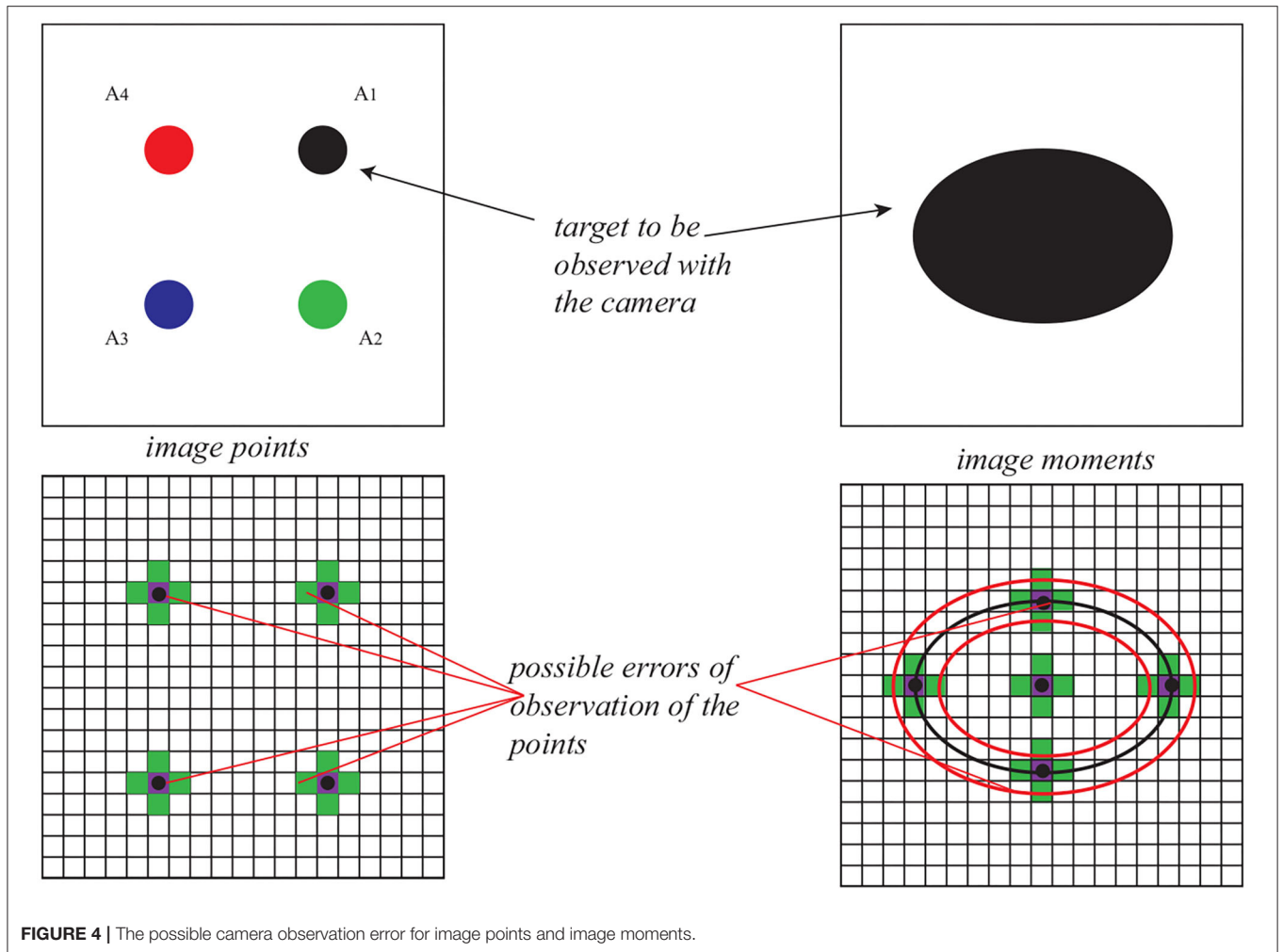
FIGURE 4 | The possible camera observation error for image points and image moments.

Delta robot

Parameters	Value
Radius of the base r_a (m)	0.1
Radius of the moving platform r_b m	0.05
Length of L_1 (m)	0.1
Length of L_2 (m)	0.25
Length of L_3 (m)	0.04
Mass of moving platform (kg)	1.94

in **Figure 1**.

$$
\boldsymbol{\tau}_\chi = \hat{\mathbf{M}}_\chi(\chi)[\hat{\mathbf{L}}_a^+(\ddot{\mathbf{s}}^d + \mathbf{K}_v \dot{\mathbf{e}}_\mathbf{s} + \mathbf{K}_p \mathbf{e}_\mathbf{s} - \mathbf{L}_v) + \mathbf{K}_f \mathbf{e}_f +
$$
$$
\mathbf{K}_i \int_0^t \mathbf{e}_f dt] + \hat{\mathbf{C}}_\chi(\chi, \dot{\chi}) + \hat{\mathbf{G}}_\chi(\chi) + \hat{\mathbf{F}} + \mathbf{J}^T(\chi)\hat{\boldsymbol{f}} \quad (12)
$$

where $\ddot{\mathbf{s}}^d$ is, respectively, the desired acceleration of the image features, $\mathbf{e}_\mathbf{s} = \mathbf{s}^d - \mathbf{s}$ is the error of image features, $\dot{\mathbf{e}}_\mathbf{s}$ is its time

derivative, and $\mathbf{e}_f = \mathbf{F}^d - \mathbf{F}$ is the error of the force applied on the robot. The rest are the same as we defined in Equation (10).

3.2. Stability Analysis

The parallel IBVS/force control (12) aims at composing the compliant displacement caused by the contact force with the desired position. In order to get a clear description of the behavior during interaction with parallel control, it is necessary to consider the contact environment. We suppose the interaction environment is a planar surface, the rotation matrix of the compliant frame $\sum_c$

$$
\mathbf{R}_c = [\mathbf{t}_{1c} \ \mathbf{t}_{2c} \ \mathbf{n}_c] \quad (13)
$$

where $\mathbf{n}_c$ is normal and $\mathbf{t}_{1c}$, $\mathbf{t}_{2c}$ are tangential to the plane. The contact model is supposed to be a spring model, and the contact force can be calculated with $\boldsymbol{f} = \mathbf{K}_e(\mathbf{P}_e - \mathbf{P}_o)$, $\mathbf{P}_o$ represents any position of the undeformed plane, and $\mathbf{P}_e$ is the equilibrium position. $\mathbf{K}_e$ denotes the contact stiffness matrix and can be obtained from

$$
\mathbf{K}_e = \mathbf{R}_c diag\{0, 0, k_{fn}\}\mathbf{R}_c^T = k_{f,n}\mathbf{n}_c\mathbf{n}_c^T \quad (14)
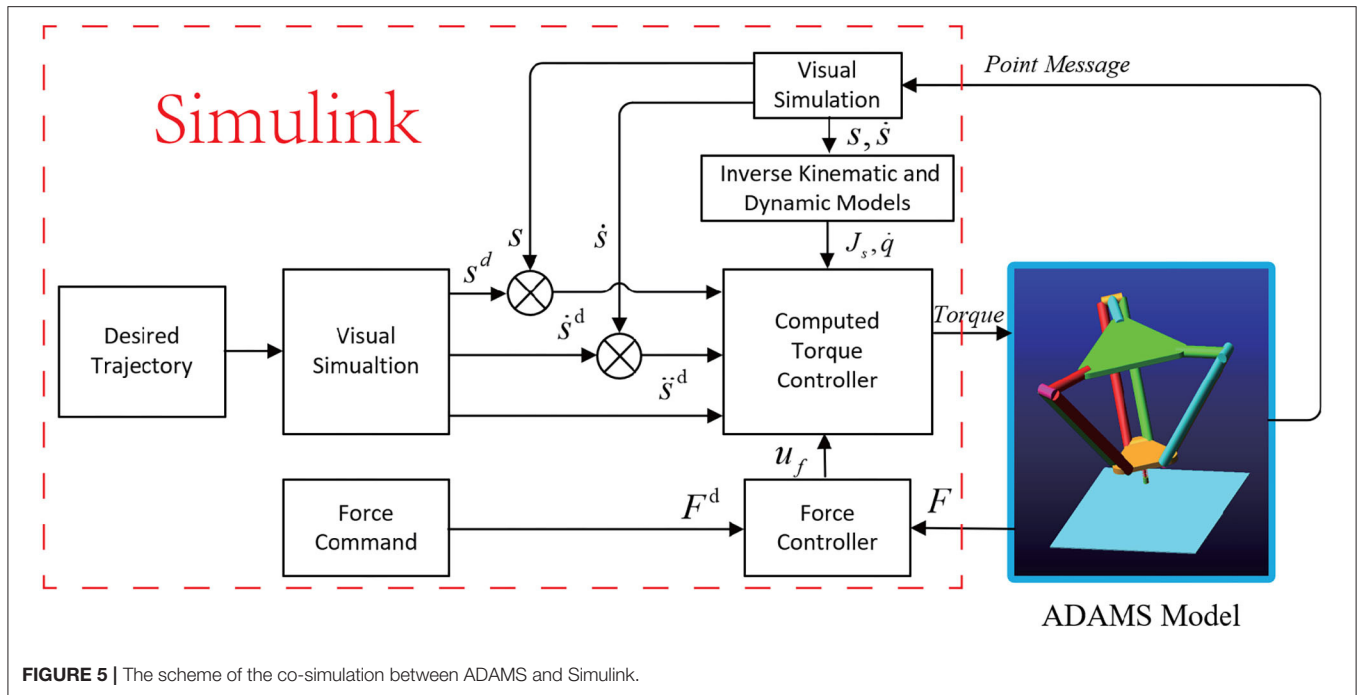$$

FIGURE 5 | The scheme of the co-simulation between ADAMS and Simulink.

TABLE 2 | Controller parameters.

K_v	K_p	K_i	K_f
5	0.1	600	15.7

with k_{fn} is positive.

The elastic spring model shows that the contact force is normal to the plane and the desired force $\mathbf{F}_d$ is aligned with $\mathbf{n}_c$.

The equilibrium position can be described as

$$\begin{aligned}
\mathbf{P}_{e,\infty} &= (\mathbf{I} - \mathbf{n}_c\mathbf{n}_c^T)\mathbf{P}_d + \mathbf{n}_c\mathbf{n}_c^T(k_{f,n}^{-1}\mathbf{F}_d + \mathbf{P}_o) \\
\mathbf{F}_\infty &= k_{f,n}\mathbf{n}_c\mathbf{n}_c^T(\mathbf{P}_{e,\infty} - \mathbf{P}_o) = \mathbf{F}_d
\end{aligned} \quad (15)$$

We can see from **Figure 2**, when a desired position $\mathbf{P}_d$ is given, the corresponding $\mathbf{P}_e$ can be obtained. $\mathbf{P}_e$ is different from $\mathbf{P}_d$ because of a vector normal to the contact plane and to guarantee $\mathbf{F}_\infty = \mathbf{F}_d$. Since the equilibrium model is described in the Cartesian space, with the help of (1) and (5), the motion control realized in task space can be projected to image feature space and the corresponding equilibrium position in vision system can be obtained with force and visual servoing.

For the system (12), achieved with the following linear behavior of the system error :

$$\hat{\mathbf{L}}_s^+(\ddot{\mathbf{e}}_s + \mathbf{K}_v\dot{\mathbf{e}}_s + \mathbf{K}_p\mathbf{e_s}) + \mathbf{K}_f\mathbf{e}_f + \mathbf{K}_i\int_0^t \mathbf{e}_f dt = 0 \quad (16)$$

where $\mathbf{K}_v = K_v\mathbf{I}$, $\mathbf{K}_p = K_p\mathbf{I}$, $\mathbf{K}_v = K_f\mathbf{I}$, $\mathbf{K}_i = K_i\mathbf{I}$.

We project (16) to $\sum_c$ and get

$$\hat{\mathbf{L}}_s^+(\ddot{\mathbf{e}}_{st} + \mathbf{K}_v\dot{\mathbf{e}}_{\mathbf{st}} + \mathbf{K}_p\mathbf{e_{st}}) = 0 \quad (17)$$

$$\hat{\mathbf{L}}_s^+(\ddot{\mathbf{e}}_{ns} + \mathbf{K}_v\dot{\mathbf{e}}_{\mathbf{ns}} + \mathbf{K}_p\mathbf{e_{ns}}) + \mathbf{K}_f\mathbf{e}_{nf} + \mathbf{K}_i\int_0^t \mathbf{e}_{nf} dt = 0 \quad (18)$$

The Equation (17) gives the dynamic of the components of the position error on the contact plane. Its stability can be obtained for any $K_v, K_p > 0$. The Equation (18) gives the components of the position error and the force error in normal direction. Considering the contact model (14), we have:

$$\hat{\mathbf{L}}_s^+(\mathbf{e_{ns}}) = k_{f,n}^{-1}\mathbf{e}_{nf} - \mathbf{d}_n \quad (19)$$

in which

$$\mathbf{d}_n = k_{f,n}^{-1}\mathbf{F}_{dn} - \mathbf{P}_{dn} + \mathbf{P}_{on} \quad (20)$$

Then the Equation (18) can be transformed to

$$k_{n,f}^{-1}\ddot{\mathbf{e}}_{nf} + k_{n,f}^{-1}\mathbf{K}_v\dot{\mathbf{e}}_{nf} + k_{n,f}^{-1}\mathbf{K}_p\mathbf{e}_{nf} + \mathbf{K}_f\mathbf{e}_{nf} + \mathbf{K}_i\int_0^t \mathbf{e}_{nf} dt = \phi_n \quad (21)$$

where

$$\phi_n = \ddot{\mathbf{d}}_n + \mathbf{K}_v\dot{\mathbf{d}}_n + \mathbf{K}_p\mathbf{d}_n \quad (22)$$

Equation (22) is a third-order system and it can be reached only if the gains satisfy the following condition:
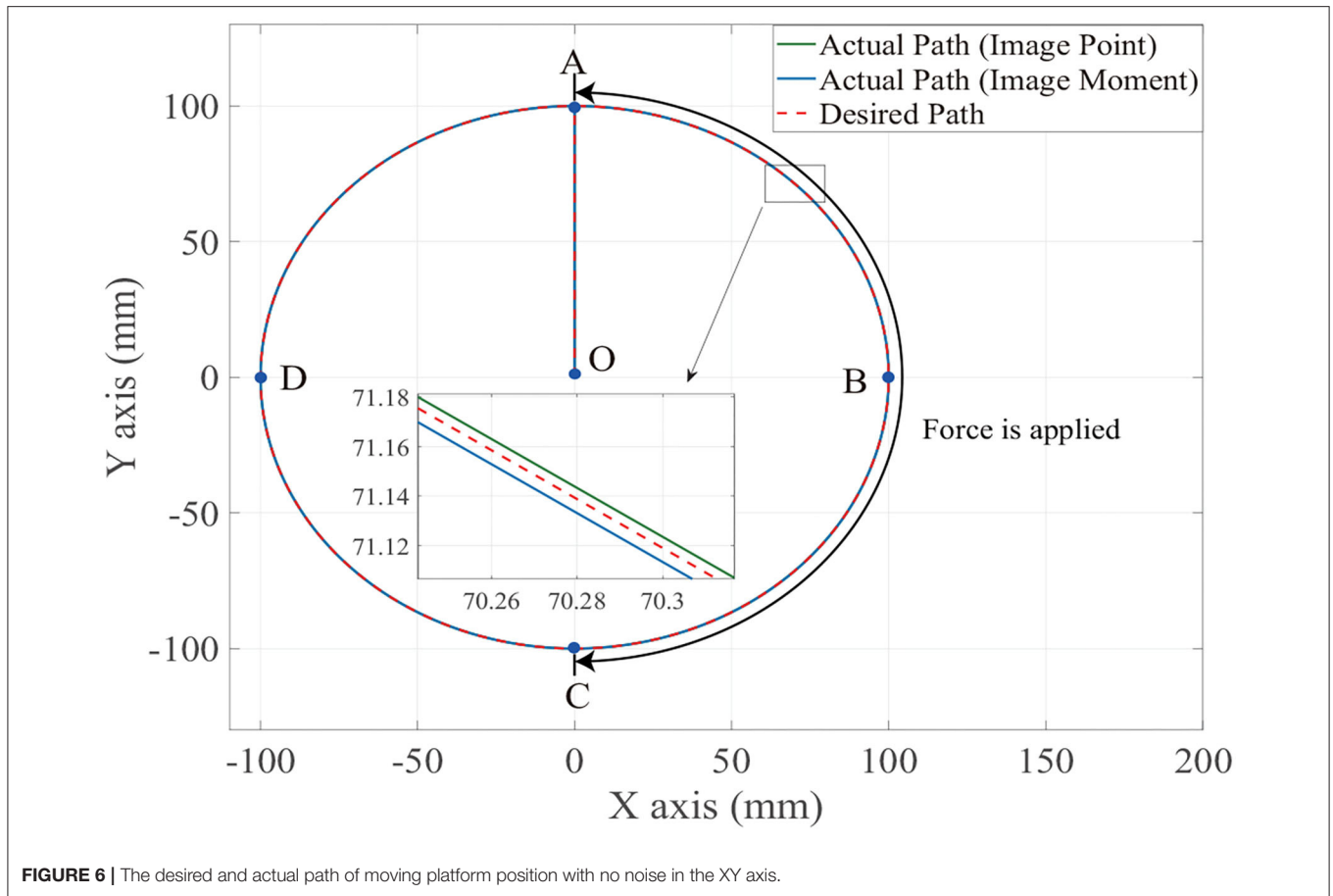
$$K_i < K_v(k_{n,f}^{-1}K_p + K_f) \quad (23)$$

FIGURE 6 | The desired and actual path of moving platform position with no noise in the XY axis.

4. DESCRIPTION OF THE COLLABORATIVE MACHINE: DELTA ROBOT

Delta robot is a well-known parallel robot with three degrees of freedom. The moving platform can only translate along the three axes of the space with respect to the fixed base. The symmetrical architecture of the Delta robot is given in **Figure 3**. The moving platform is connected to the base with three identical kinematic chains. Each chain is actuated by the revolute motor located at $A_i(i = 1, 2, 3)$ in the base. A spatial parallelogram $B_{i1}B_{i2}C_{i2}C_{i1}$ $(i = 1, 2, 3)$ is used to connect the active link A_iB_i $(i = 1, 2, 3)$ and the platform. The triangles $A_1A_2A_3$ and $C_1C_2C_3$ are equilateral, the links A_iB_i $(i = 1, 2, 3)$ are moving in vertical planes containing OA_i, the lengths for each link belonging to the different legs are the same: $\|\overrightarrow{A_iB_i}\| = L_1$, $\|\overrightarrow{B_iC_i}\| = L_2$, $\|\overrightarrow{B_{i1}B_{i2}}\| = \|\overrightarrow{C_{i1}C_{i2}}\| = L_3$ for $(i = 1, 2, 3)$.

Since the Delta robot can only translate along the three axis in space, we only consider the terms along the end-effector DOF, and the Coriolis and centrifugal terms are not taken into account. The dynamics of the moving platform projected to the task space are then reduced to:

$$\mathbf{F}_p = \mathbf{M}\ddot{\chi} - \mathbf{G}_p + \mathbf{F} \tag{24}$$

where $\mathbf{M}$ is the mass matrix of the platform and $\mathbf{G}_p$ is the gravity effect of the moving platform of the Delta robot. $\mathbf{F}$ is the external force applied to the platform because of the interaction with the environment.

To fully control the three DOF of the Delta robot and avoid the global minima at the same time (Zhu et al., 2020), the target to be observed is designed to be four points in one plane (for image points) and an ellipse (for image moments). Then, the image feature $\mathbf{s}$ is defined as $\mathbf{s} = [x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4]^T$ for image points visual servoing and $\mathbf{s} = [x_g, y_g, a]^T$ for image moment visual servoing (x_g, y_g the center of the gravity of the target and a the area of the target; see **Figure 4**). The interaction matrix $\mathbf{L}_s$ corresponding to the 4 image points can be obtained by stacking (7). For image moment visual servoing, the expression of the corresponding interaction matrix is as follows:
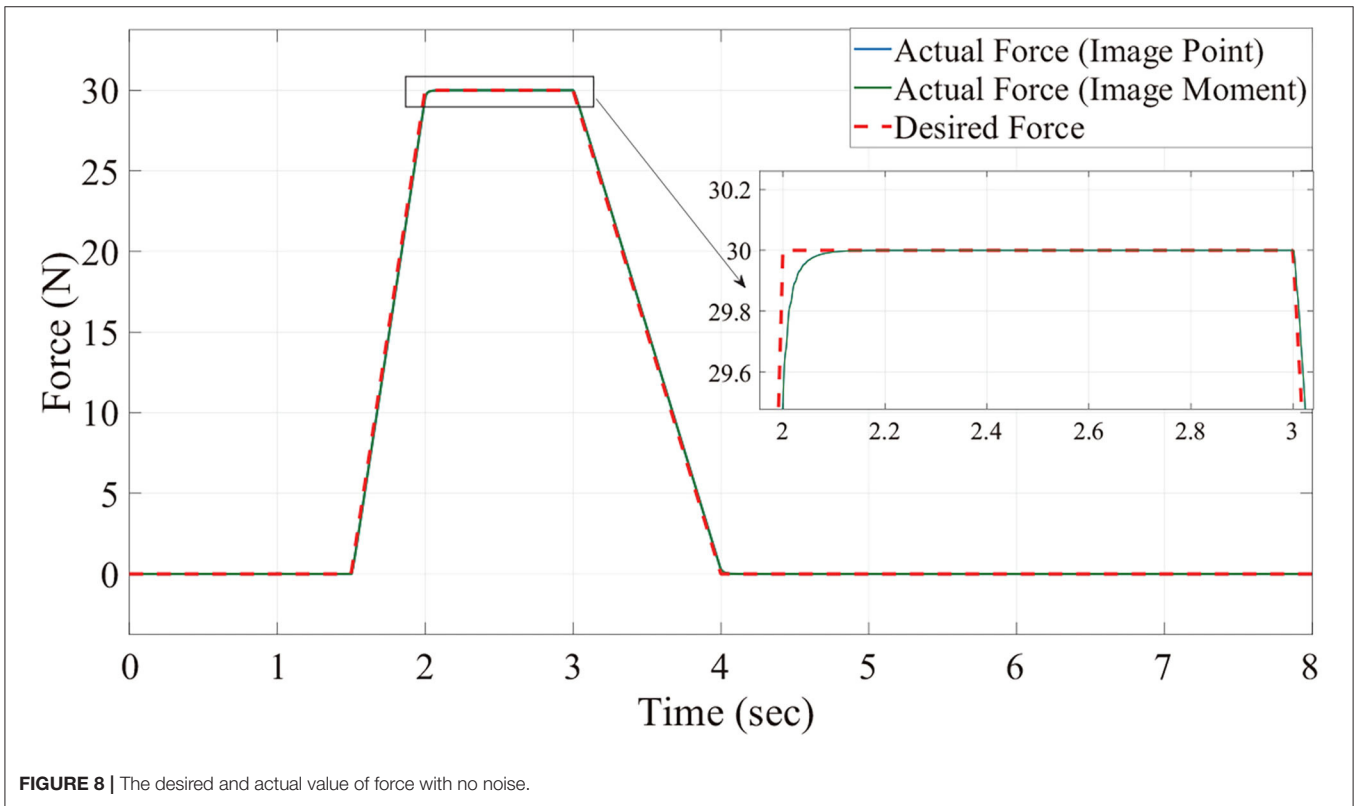
$$\mathbf{L}_s = \begin{bmatrix} -C & 0 & C\,x_g \\ 0 & -C & C\,y_g \\ 0 & 0 & 2\,a\,C \end{bmatrix} \tag{25}$$

where $c = \frac{1}{Z}$, and Z being the depth between the camera and the object.
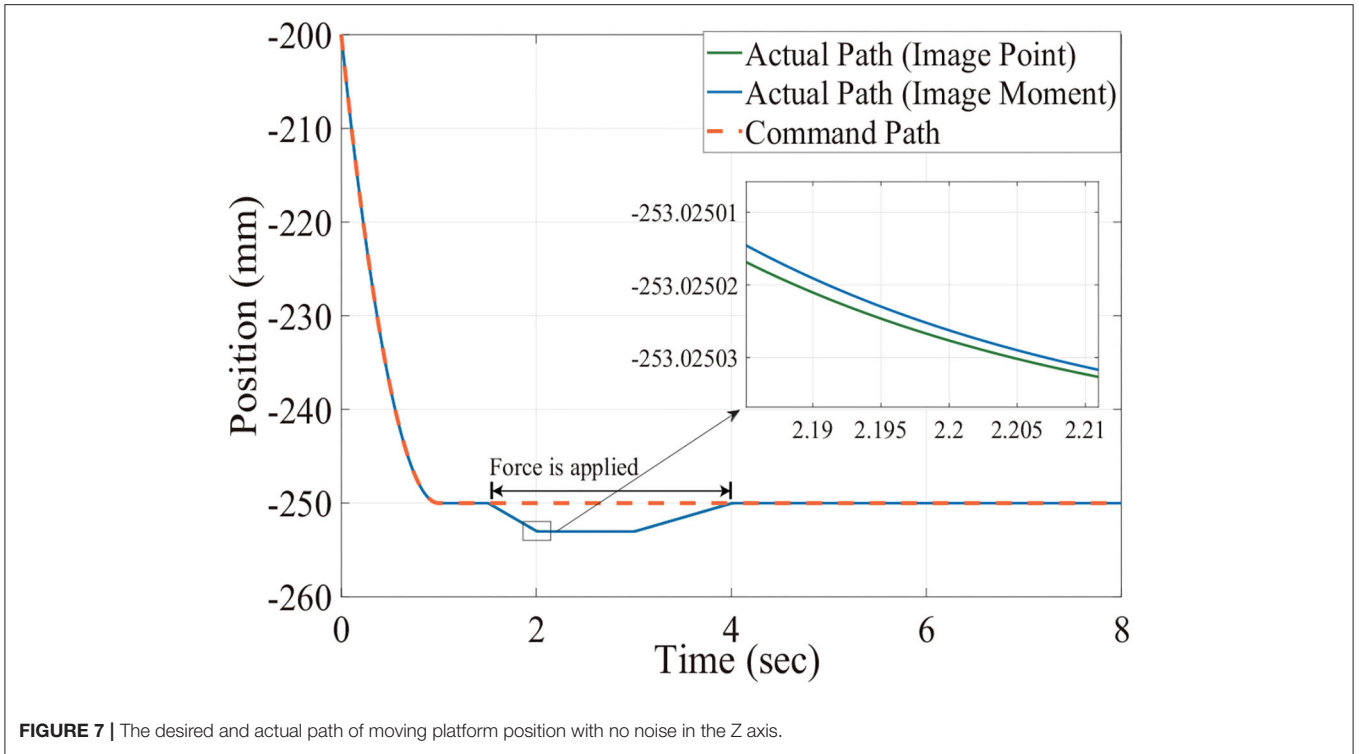
FIGURE 7 | The desired and actual path of moving platform position with no noise in the Z axis.

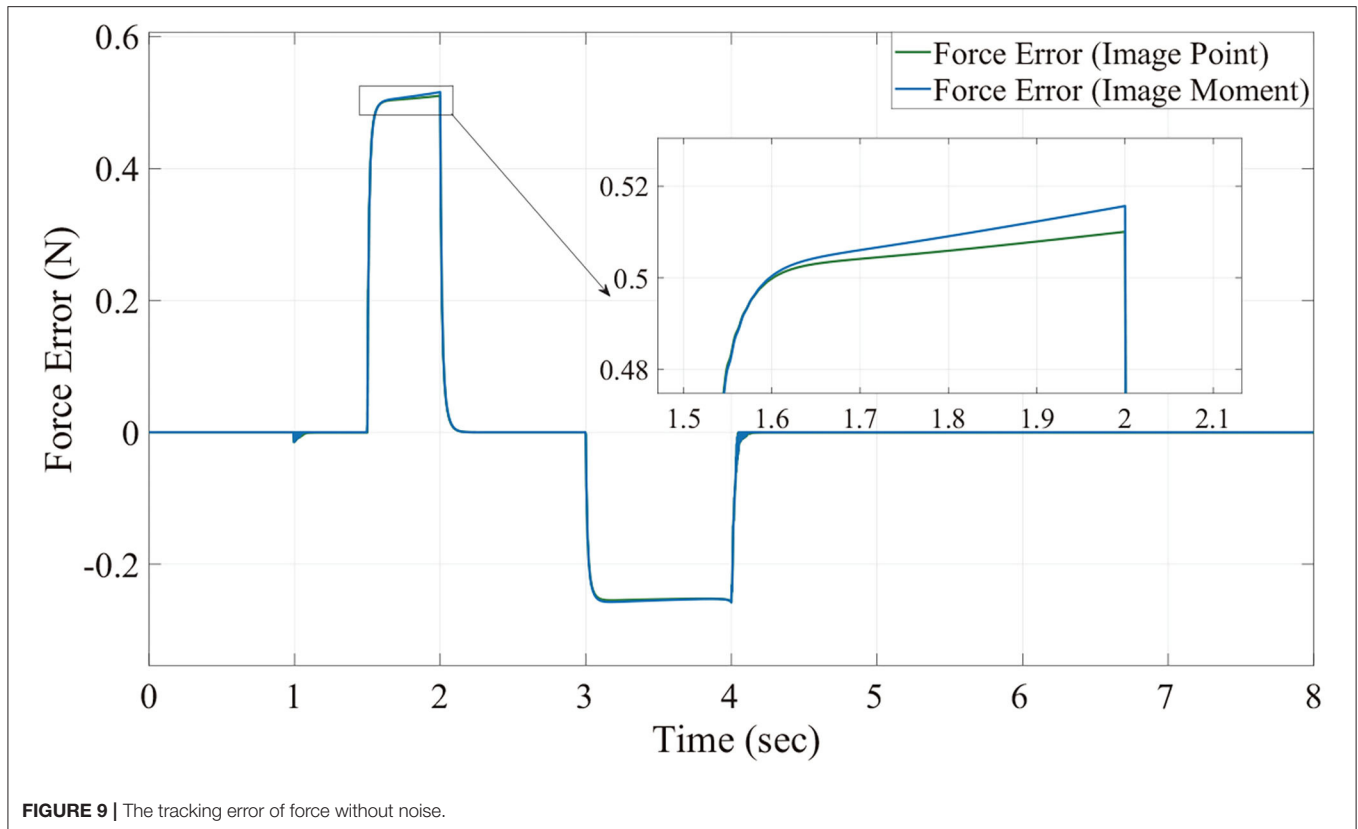**FIGURE 8 |** The desired and actual value of force with no noise.

FIGURE 9 | The tracking error of force without noise.

5. SIMULATION

5.1. Specification of the Simulation Environment

To test the parallel IBVS/force controller performance, co-simulations are performed in a connected Simulink-ADAMS environment. The Delta robot mechanical model is created in the software ADAMS with the geometric and dynamic parameters in **Table 1**, and the pinhole model of the camera and the controller is developed with Matlab/Simulink. To simulate the manufacturing and assembly error in reality, an uncertainty of 10% is added to the inertia parameters of the moving platform of the Delta robot and a 50 μm error is added to the geometric parameters as it was done in Bellakehal et al. (2011). The planar target is assumed to be attached to the moving platform. The contact surface of the target object is parallel to the base of the Delta robot. Then, there exerts a perpendicular force onto the surface of the moving platform when it is contacted with the target object surface. The interaction surface is defined as a spring and the model can be written as follows:

$$\mathbf{F} = \mathbf{K}_e \Delta \chi \qquad (26)$$

The stiffness of the spring model takes the value of 10^4 N/m.

During this simulation, the force sensor is fixed to the upper surface of the end-effector of the Delta robot and its resolution is $[\frac{1}{160}\text{N}, \frac{1}{160}\text{N}, \frac{1}{160}\text{N}]$. A noise of $\pm\frac{1}{160}$ N is added to the force

measurement feedback in order to test the robustness of the parallel IBVS/force controller.

5.2. Vision System

In this simulation, a single *eye-to-hand* pinhole camera model is considered. This camera is fixed along the z_0 axis of the world frame (see **Figure 3**) so that the camera can observe the target in a symmetrical way. The resolution of the camera is $1,920 \times 1,200$ pixels and 1 pixel/mm focal length.

For the target to be observed, in image point visual servoing, the coordinates of the four points A_1, A_2, A_3, A_4 are, respectively, (0.3, 0.3,0), (0.3, −0.3,0), (−0.3, −0.3,0), (−0.3, 0.3,0) m, with respect to the moving platform frame. In image moment visual servoing, the target is a circumcircle of A_1, A_2, A_3, A_4.

It has been proven in Zhu et al. (2020) that the positioning error of visual servoing comes from the camera observation error. Therefore, in this case, the value of the camera observation noise is set to be ± 0.1 pix, which is a typical noise for cameras roughly calibrated (Bellakehal et al., 2011, see **Figure 4**). In Zhu et al. (2020), it has been detaily presented on how to add noise when observing an ellipse using image moment visual servoing (**Figure 4**).

5.3. Simulation Results

In this section, the parallel IBVS/force control scheme is applied to the control of a collaborative Delta robot to track the position trajectory and force command. For the controller based on image points, we extracted the coordinates of the image points. For
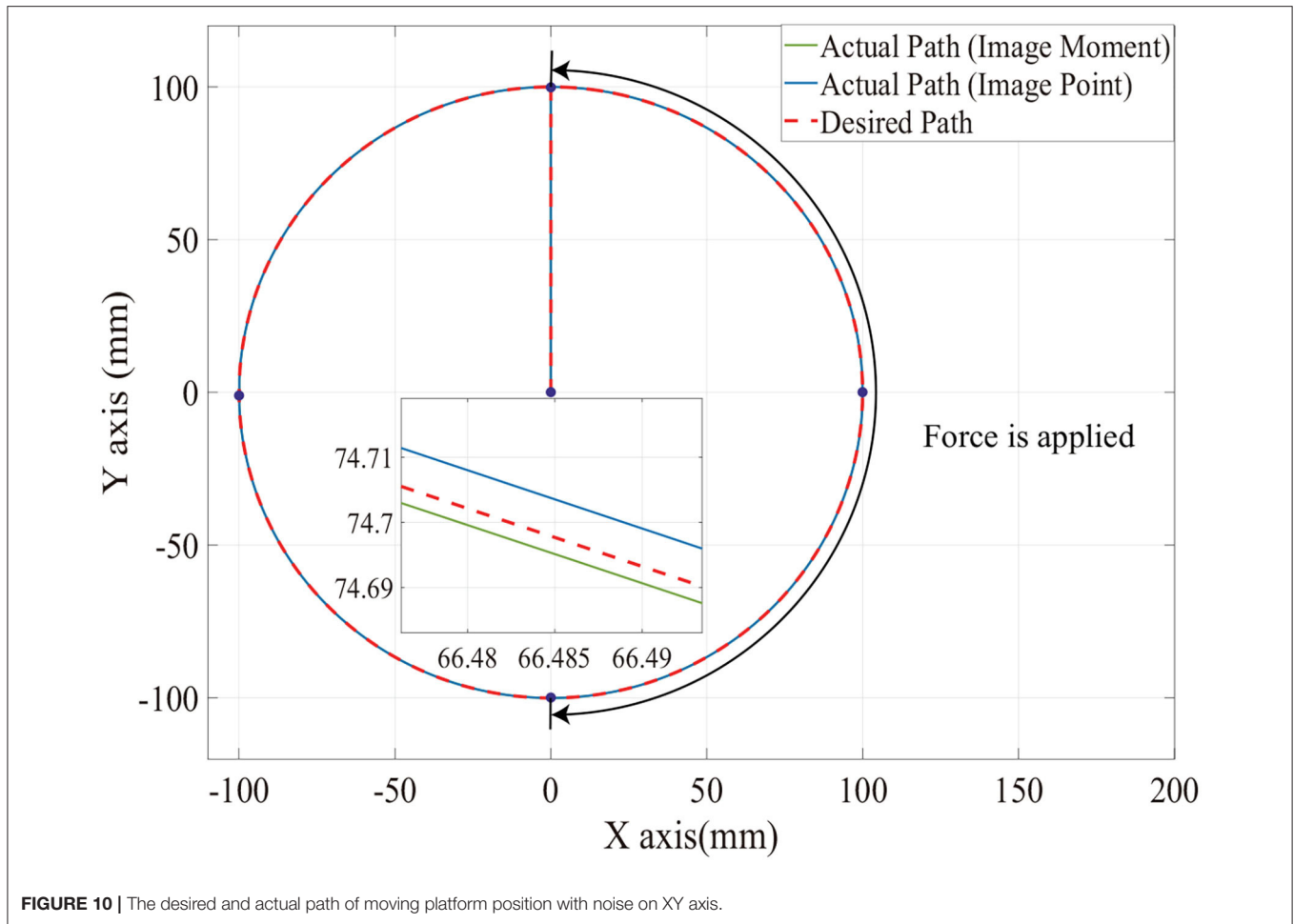
FIGURE 10 | The desired and actual path of moving platform position with noise on XY axis.

the controller based on the image moment, we extracted the coordinates of the centroid point of the observed ellipse and the two points at the extremity of its minimal and maximal radii. Based on these extracted data, we rebuild the image seen by the camera and extract the image feature, use them and the contact force to control the collaborative robot. The scheme of the co-simulation is illustrated in **Figure 5**.

First we define a position trajectory and a force trajectory. The Delta robot is driven with the corresponding dedicated controller to track these trajectories. The simulations with no noise added to the image plane have been performed to show the effectiveness of the IBVS/force control system. Then, the noise is added to the camera observation and the feedback of the force sensor to show the robustness of the IBVS/force control system. In addition, a comparison between the IBVS/force controller based on image points and image moments has been presented.

The parameters of the controller are given in **Table 2** where K_p, K_v, K_f, and K_i are the coefficients of the parallel IBVS/force control in **Figure 1**.

Figures 6, **7** show the position trajectory in x-y plane and z axis, and **Figure 8** gives the force command. The moving platform of the Delta robot is driven to move to point A

[(0, 100, −250) with respect to the global frame] in 1 s. Then, it is controlled to move clockwise along a circular trajectory in the x-y plane and follow the force command at the same time.

We can see from **Figures 6**, **7**, for both image point visual servoing and image moment visuals servoing, the position errors are within the order of 10^{-2} mm. We see that a residual constant position error exists along the force controlled direction (z axis) after 1.5 s (**Figure 7**). This error depends on the exerted force and the stiffness of the interaction environment. From the results illustrated in **Figures 6**, **7**, it is obvious that the two kinds of parallel IBVS/force control systems are able to control the position and follow the force trajectory simultaneously. **Figures 8**, **9** show the force track error and the differences between image point visual servoing/force control and image moment visual servoing/force control are not big enough to draw the conclusion which one is better.

Then, in order to test which image feature is more suitable for the parallel IBVS/force control in terms of robustness, the Gaussian noise is added to the camera observation and the feedback of the force sensor. The position trajectory and the force command are the same as we did above. The simulation results are illustrated in **Figures 10**, **12**, **13**.
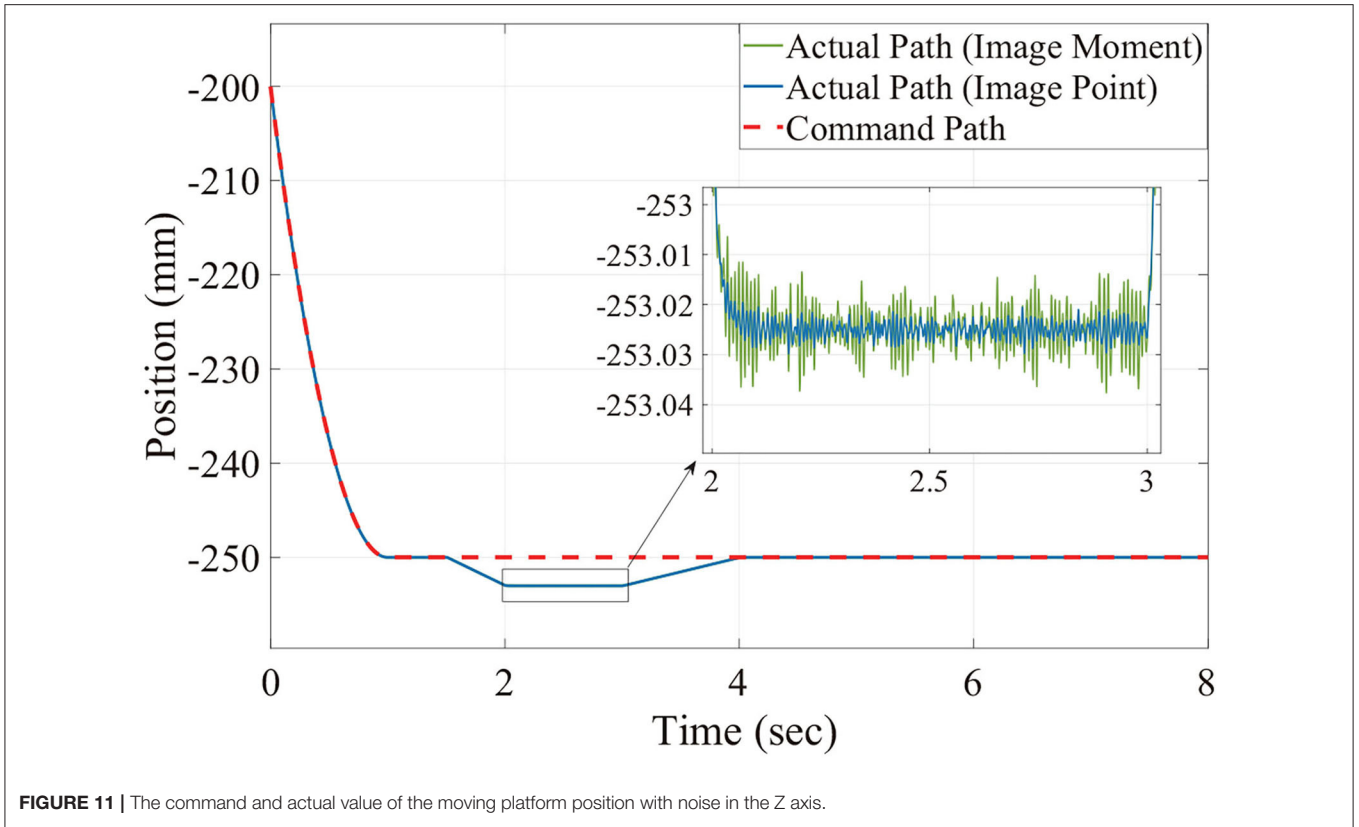
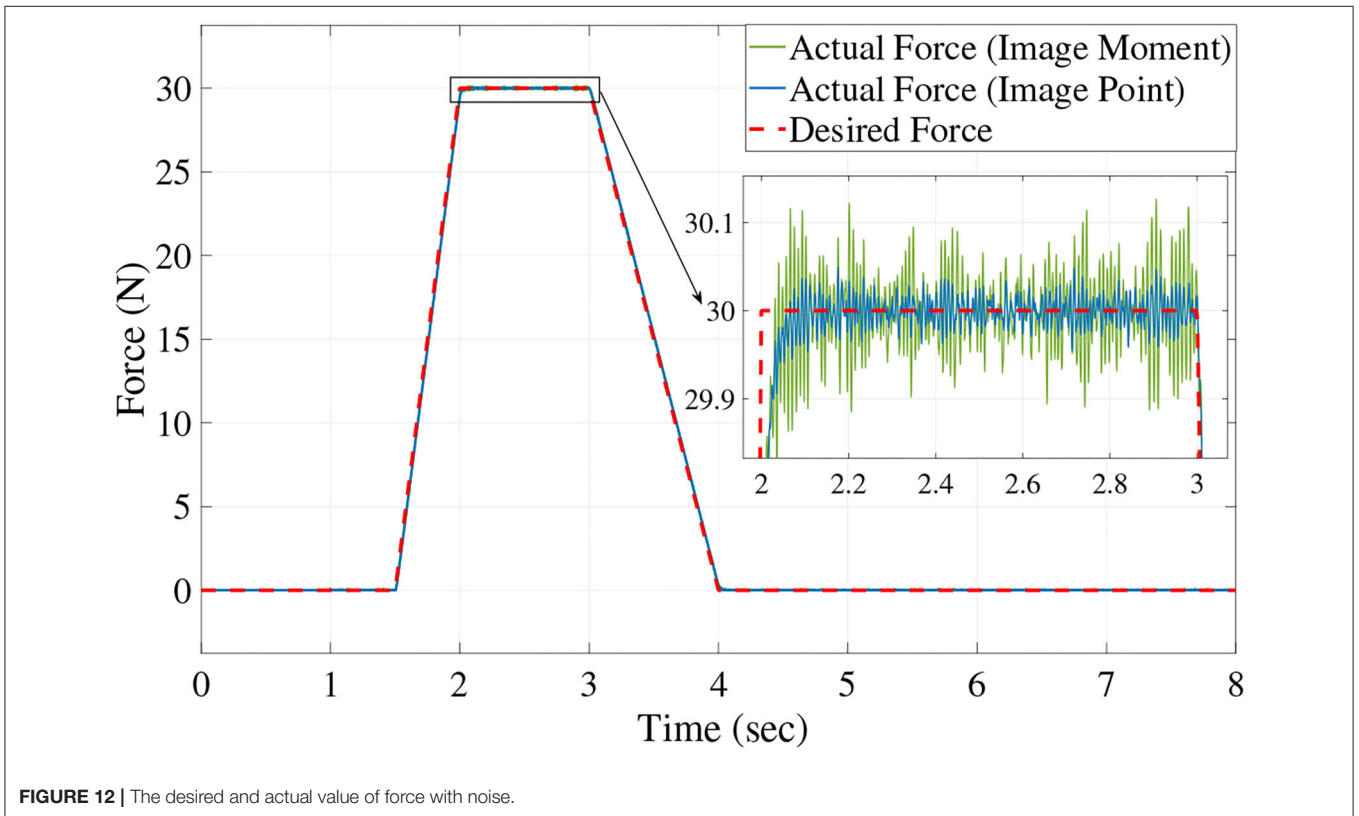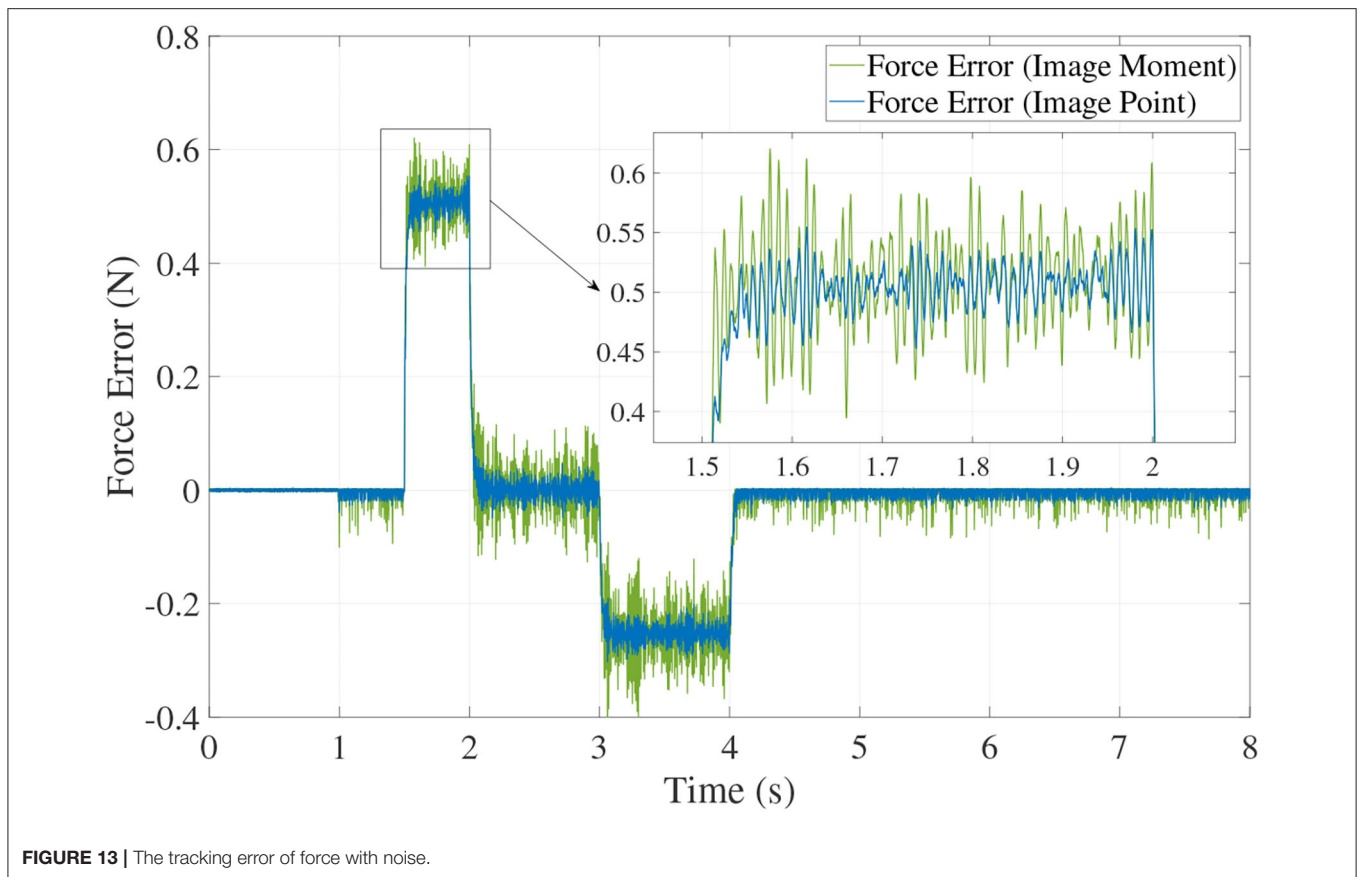**FIGURE 11 |** The command and actual value of the moving platform position with noise in the Z axis.

FIGURE 12 | The desired and actual value of force with noise.

FIGURE 13 | The tracking error of force with noise.

From **Figures 11**, **12**, **13**, we see that the robustness of the controller based on the image point is better than the controller based on image moment. When the same noise is added, the position track accuracy and force track accuracy of the image point visual servoing/force controller are two to three times better than those of the image moment visual servoing/force controller.

Compared with previous work by Callegari and Suardi (2003), when the model-based control scheme is used in the position/force control with parallel robots, the force error is more than 3 N while the desired contact force is 10 N (control precision is about 30%). Our simulation results show the overwhelming superiority of parallel IBVS/force upon the model-based position/force control scheme in the case of parallel manipulators. Moreover, compared with the work in Bellakehal et al. (2011), the parallel PBVS/force is applied to the position and force control of parallel manipulators. Based on the premise that the same uncertainty added to the geometric and dynamic parameters of the parallel robots and the noise added to the camera observation and force sensor, the positioning accuracy and orientation accuracy of this controller is 4–10 times better than those in Bellakehal et al. (2011). The force control precision of the parallel IBVS/force controller is about 0.42–1.3% and it is three times better than that of the force controller in Bellakehal et al. (2011), which is about 4%.

6. CONCLUSION

In this paper, a parallel image-based visual servoing/force controller is developed to control the motion of the collaborative robot and follow the contact force command at the same time. This controller has been proposed to generate the output torque signal based on the errors from the image plane and to control the robot by tracking the position trajectory and force command. Two kinds of image features, normalized image points and image moments, are selected to be the feedback of the vision system. The stability of this controller is proven. Simulations are performed on a collaborative Delta robot and the results show that this controller is able to control both contact forces and the motion of the end-effector of the robot at the same time. In addition, the results of the simulation with noise added to the camera observation and force sensor feedback prove that the robustness of the controller based on the image point is better than the controller based on image moment. This proposed parallel IBVS/force controller will be experimentally validated on the collaborative Delta robot prototype in the future.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

FUNDING

REFERENCES

Andreff, N., Dallej, T., and Martinet, P. (2007). Image-based visual servoing of a gough-stewart parallel manipulator using leg observations. *Int. J. Robot. Res.* 26, 677–687. doi: 10.1177/0278364907080426

Bellakehal, S., Andreff, N., Mezouar, Y., and Tadjine, M. (2011). Vision/force control of parallel robots. *Mech. Mach. Theory* 46, 1376–1395. doi: 10.1016/j.mechmachtheory.2011.05.010

Callegari, M., and Suardi, A. (2003). "On the force-controlled assembly operations of a new parallel kinematics manipulator," in *IEEE Mediterranean Conference on Control & Automation*, Rhodes.

Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Trans. Robot.* 20, 713–723. doi: 10.1109/TRO.2004.829463

Chaumette, F., and Hutchinson, S. (2006). Visual servo control. I. Basic approaches. *IEEE Robot. Autom. Mag.* 13, 82–90. doi: 10.1109/MRA.2006.250573

Chaumette, F., Hutchinson, S., and Corke, P. (2016). "Visual servoing," in *Springer Handbook of Robotics* (Springer), 841–866. doi: 10.1007/978-3-319-32552-1_34

Chiaverini, S., and Sciavicco, L. (1993). Parallel approach to force/position control of robotic manipulators. *IEEE Trans. Robot. Autom.* 9, 361–373. doi: 10.1109/70.246048

Dai, X., Song, S., Xu, W., Huang, Z., and Gong, D. (2021). Modal space neural network compensation control for gough-stewart robot with uncertain load - sciencedirect. *Neurocomputing* 449, 245–257. doi: 10.1016/j.neucom.2021.03.119

Dallej, T., Andreff, N., Mezouar, Y., and Martinet, P. (2006). "3D pose visual servoing relieves parallel robot control from joint sensing," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing), 4291–4296. doi: 10.1109/IROS.2006.281959

Fu, C., Li, B., Ding, F., Lin, F., and Lu, G. (2021). Correlation filters for unmanned aerial vehicle-based aerial tracking: a review and experimental evaluation. *IEEE Geosci. Remote Sens. Mag.* 10, 125–160. doi: 10.1109/MGRS.2021.3072992

Fusco, F. (2020). Dynamic visual servoing for fast robotics arms. Automatic. École centrale de Nantes.

Hogan, N. (1985). Impedance control: an approach to manipulation: parts I, II, III. *ASME J. Dyn. Syst. Measure. Control* 107, 17–24. doi: 10.1115/1.3140701

Jeanneau, G., Bégoc, V., and Briot, S. (2020). "Geometrico-static analysis of a new collaborative parallel robot for safe physical interaction," in *Proceedings of the ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.*

Magrini, E., and De Luca, A. (2016). "Hybrid force/velocity control for physical human-robot collaboration tasks," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, Daejeon. doi: 10.1109/IROS.2016.7759151

Mason, M. T. (1981). Compliance and force control for computer controlled manipulators. *IEEE Trans. Syst. Man Cybern.* 11, 418–432. doi: 10.1109/TSMC.1981.4308708

Merlet, J.-P. (2006). *Parallel Robots, Vol. 128*. Springer Science & Business Media. doi: 10.1115/1.2121740

Mohebbi, A. (2012). Augmented image based visual servoing of a manipulator using acceleration command. *IEEE Trans. Indus. Electron.* 61, 5444–5452. doi: 10.1109/TIE.2014.2300048

Raibert, M. H., and Craig, J. J. (1981). Hybrid position/force control of manipulators. *ASME J. Dyn. Syst. Measure. Control* 102, 126–133. doi: 10.1115/1.3139652

Siciliano, B., and Villani, L. (1999). *Robot Force Control*. Springer Science & Business Media. doi: 10.1007/978-1-4615-4431-9

Tahri, O., and Chaumette, F. (2005). Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. Robot.* 21, 1116–1127. doi: 10.1109/TRO.2005.853500

Wu, J., Gao, Y., Zhang, B., and Wang, L. (2017). Workspace and dynamic performance evaluation of the parallel manipulators in a spray-painting equipment. *Robot. Comput. Integr. Manuf.* 44, 199–207. doi: 10.1016/j.rcim.2016.09.002

Xu, P., Li, B., and Chueng, C.-F. (2017). "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)*, Marina Bay Sands, 379–384. doi: 10.1109/ICAR.2017.8023636

Yang, G., Chen, I. M., Yeo, S. H., and Lin, W. (2008). "Design and analysis of a modular hybrid parallel-serial manipulator for robotised deburring applications," in Smart Devices and Machines for Advanced Manufacturing, eds L. Wang and J. Xi (London: Springer). doi: 10.1007/978-1-84800-147-3_7

Zabihifar, S., and Yuschenko, A. (2018). "Hybrid force/position control of a collaborative parallel robot using adaptive neural network," in *International Conference on Interactive Collaborative Robotics*, Leipzig. doi: 10.1007/978-3-319-99582-3_29

Zhu, M., Chriette, A., and Briot, S. (2020). "Control-based design of a DELTA robot," in *ROMANSY 23 - Robot Design, Dynamics and Control, Proceedings of the 23rd CISM IFToMM Symposium*, Sapporo. doi: 10.1007/978-3-030-58380-4_25

Zhu, M., Huang, C., Song, S., and Gong, D. (2022). Design of a gough stewart platform based on visual servoing controller. *Sensors* 22:2523. doi: 10.3390/s22072523