Robotica (2018) volume 36, pp. 588–606. © Cambridge University Press 2018
doi:10.1017/S0263574717000662

Emulation of pilot control behavior across a Stewart platform simulator

Mojtaba Eftekhari†* and Hossein Karimpour‡

†*Department of Mechanical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran*
‡*Department of Mechanical Engineering, Faculty of Engineering, University of Isfahan, Isfahan, Iran, E-mail: h.karimpour@eng.ui.ac.ir*

(Accepted December 22, 2017. First published online: January 22, 2018)

SUMMARY
This paper presents a model-based controller consisting of a feedback linearization scheme and a state-dependent proportional derivative (PD) controller adapted to a parallel flight simulator Stewart mechanism. This parallel robot is considered to emulate motions of highly maneuverable aircrafts, which require well-trained pilots. The simulations are based upon a reduced-model prototype built in order to verify kinematic design aspects and control laws. Indeterminacies in the mass distribution of the system will generally affect model-based controllers, necessitating compensation or the employment of robust control methods. Through introducing the pilot's sensorial feedback of acceleration, the pilot's behavior in giving commands is emulated via an optimization process, which tunes the controller coefficients accordingly. Stability of the designed control system is guaranteed via the Lyapunov approach. To further explore the system through perilous flight scenarios, three pre-designed maneuvers are selected as test cases. It is expected that closed-loop control tasks in which a pilot tracks a target, while at the same time the controller rejects disturbances and adapts itself to the pilot's progressive skills, are ameliorated through this arrangement. Numerical results show that the proposed method is found robust in the training process in conditions of parameters indeterminacy.

KEYWORDS: Stewart platform; Pilot-in-the-loop; Nonlinear PD controller; Genetic algorithm.

1. Introduction

Flight simulators provide an effective, efficient, and safe environment for training pilots in flight-critical maneuvers that may be encountered in real situations. Most simulators are equipped with a Stewart-type motion system, which consists of six linear actuators in a hexapod configuration, providing a full six degrees of freedom (6-DOF) platform for mimicking flight motion and sustaining the heavy mass of the simulator.

The argument for use of dynamic simulators is derived from the reproduction of motion cues experienced during flight.[1] It is undeniable that if pilots would train without any simulator motion, relying only on display systems, the motion accelerations present in real flight could disorient the pilot, affecting the performance or even entraining disastrous consequences. Therefore, flight simulator motion systems have to faithfully reproduce aircraft motion experienced in flight via realistic visual and vestibular feedbacks. The novelty of this paper lies in the following aspects. First, a model-based controller consisting of a feedback linearization scheme and a state-dependent proportional derivative (PD) controller is adapted to a parallel flight simulator mechanism. Then, through introducing the pilot's sensorial feedback of acceleration, the pilot's behavior in giving commands is emulated via an optimization process, which tunes the controller coefficients accordingly. It is expected that closed-loop control tasks in which a pilot tracks a target, while at the same time the controller rejects disturbances and adapts itself to the pilot's progressive skills, are ameliorated through this arrangement.

* Corresponding author. E-mail: mo.eftekhari@uk.ac.ir

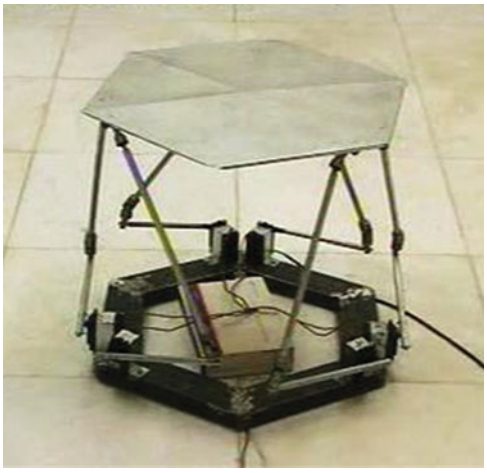

Fig. 1. Prototype of the Stewart platform with rotary actuators.

Compared with serial ones, parallel manipulators have relative advantages in terms of high stiffness, accuracy, and speed, thus enabling a vast range of applications like pick-and-place operation in electronic industry, precision machining, or multi-axis heavy duty testing jacks.[2] The key issues are the ways to meet the demand of high accuracy in addition to high speed and heavy load. Parametric uncertainties and highly coupled dynamics, respectively, due to payload variations and unmodeled interactions, such as links flexibility, friction, or actuators backlash, make the controller difficult to design. In the present case, the fidelity of the simulator in mimicking motion while inducing the physical feeling of piloting an aircraft depends mainly on the mechanism maneuverability and requires special controller attributes. Historically, Stewart platform was originally designed as a six-linear jack system for simulating flight conditions and training pilots.[3] Later, the mechanism was used as a robot wrist and a tendon actuated in form of parallel manipulators. Since then, several types of this mechanism have been proposed. Rotary Hexapods are more recent and peculiar designs of such parallel manipulators and were introduced by Hexel Crop.[4] Among different versions of Hexapods,[5–8] a new mechanism type called Pierrot's HEXA is considered in this paper. The mechanism is recognized as more efficient than its prismatic counterpart for accelerating the moving platform to required jerk levels while executing aggressive large magnitude maneuvers. As shown in Fig. 1, the mechanism consists of six articulated legs joined to a common platform plate. The lower links are connected to the base by revolute joints and the upper links are connected to the moving plate by universal joints. Upper and lower links are connected to each other by spherical joints.

The control strategies of parallel manipulators are generally divided in two schemes: decentralized and distributed control strategies. In the first control strategy, individual legs of the parallel manipulator are assumed as independent entities and the coupling effect from other legs is considered as a disturbance.[9] Although easily implemented in parallel to reduce computation load, nonetheless without synchronization, there is no guarantee that the group of decentralized individual controllers will prove stable in performing high-performance tasks. On the other hand, distributed control strategies are based on task space error fed back to an unified controller.[10,11] Unlike the decentralized strategy, the dynamic coupling of manipulators is taken into account in order to compensate the nonlinear dynamics of the parallel robot and hence permits to achieve better performance.

Many researchers utilized a PD controller plus a gravitation compensation, linear and nonlinear synchronized Proportional-Integral-Derivative (PID) controller for individual position control of parallel manipulators.[12–16] Davliakos and Papadopoulos have developed a model-based controller for a 6-DOF hydraulically driven Stewart platform.[17] The dynamic model contained the rigid body equations of the Stewart mechanism augmented by the hydraulic dynamics of the actuators. Another approach consisted of a nonlinear input–output linearization control method.[18] The control law included a PD part driving the tracking error to zero exponentially. Moreover, the same authors

designed a novel model-based impedance controller[19] for dealing with the electrohydraulic servo-actuators and compared the results with those obtained previously from the PD controller. The results of impedance controller were relatively superior, showing more smoothness during interaction with the environment.

Zubizarreta et al. implemented an extended computed torque controller on a 6-DOF UPS kinematic chain of the Stewart platform.[20] A sensitivity-based analysis was performed on the mechanism to increase the robustness and performance of the control law. An adaptive sliding mode controller was proposed by Dongsu and Hongbin for a 6-DOF Stewart platform.[21] The uncertain constant parameters were on-line estimated by the adaptive control part and the time-varying uncertain parameters, perceived as external disturbances, were compensated by the sliding mode part. Ting et al. relied on Lyapunov theory to prove the stability of a PID controller optimally tuned by genetic algorithm (GA).[22] Experimental results done in the presence of external load conditions showed that a satisfactory level of accuracy was maintained. Omran and Kassem proposed a similar optimization method to tune the parameters of a PD controller applied on a 6-DOF Stewart platform.[23] Controller gains were tuned to minimize the acceleration jerk delivered to the pilot according to mission characteristics imbedded into the cost function. Yang et al. analyzed, both theoretically and experimentally, the performance of a PD controller with gravity compensation on a 6-DOF hydraulic parallel manipulator.[24] The hydraulic system was decoupled by local velocity compensation via inner control loops, proving satisfying results in terms of stability, robustness, and accuracy.

Among the multitude of research works done in the past decade related to the control of various versions of Gough-Stewart platforms, the rotary actuated system is still moderately studied. Azizan et al. designed a stable model-based fuzzy controller for this type of manipulator.[25] Stability of the controller has been guaranteed via the Lyapunov approach. Eftekhari et al. presented the adaptability of a neuro-adaptive fuzzy controller on a platform designed to transport loads of unknown inertia.[26] The effectiveness of the neuro-fuzzy system has been approved by performing multiple maneuvers and its robustness was checked under various inertia loads.

One of the promising control methods enabling to compensate for uncertainties is the nonlinear PID control approach.[27] Many authors have proposed the control of the Stewart platform using sliding mode control but in practical applications, those conventional controllers show certain drawbacks including chattering and the lack of robustness to unmatched uncertainties. In this paper, a nonlinear augmented proportional plus derivative controller is proposed to deal with a 6-DOF Stewart platform with revolute actuators designed to emulate highly maneuverable aircrafts. The dynamic model of the system, consisting of interconnected rigid bodies, is built using Lagrange's method and becomes unconstrained through projecting the equations on the tangent to the constrained manifold. The next step consists of optimal gain selection for the nonlinear PD (NPD) controller. Various methods have been proposed to handle systems with matched and unmatched uncertainties. However, in the present case, GA is used to optimally select the gains according to the fulfillment of prerequisite conditions. Stability of the designed controller is guaranteed via Lyapunov approach. Fidelity of the simulator in recreating flight conditions can be supplemented by estimating the pilot's perception of flight through mathematical models fitted to the measured pilot's reactions. A tracking mission with pilot in the loop can serve to assess the effect of human factor and to conceive how to compensate for it without affecting system performance.

2. Manipulator Equations of Motion

2.1. Kinematics model

As shown in Fig. 2, the motion of the moving platform is described by two frames: an inertial frame (X, Y, Z) attached at the center of the footing base with unit vectors $(\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}})$ and a body frame (x_p, y_p, z_p) attached at the center of moving platform with unit vectors $(\hat{\mathbf{I}}_p,\ \hat{\mathbf{J}}_p,\ \hat{\mathbf{K}}_p)$, where the z_p axis is pointing outward. Other complementary coordinate systems (x_n, y_n, z_n), $n = 1, 2 \ldots 6$ and (x'_n, y'_n, z'_n), $n = 1, 2 \ldots 6$ are attached, respectively, to the bottom and upper links with unit vectors$(\hat{\mathbf{i}}_n, \hat{\mathbf{j}}_n, \hat{\mathbf{k}}_n)$ and $(\hat{\mathbf{i}}'_n, \hat{\mathbf{j}}'_n, \hat{\mathbf{k}}'_n)$, $n = 1 \ldots 6$.

The moving platform has three translational and three rotational degrees of motions. The rotational motions of the moving platform are defined in a sequence of Euler angles (θ, φ, ψ), respectively, pitch, roll, and yaw about the y, x, and z body axes. The transformation of the body frame (x_p, y_p, z_p) relative

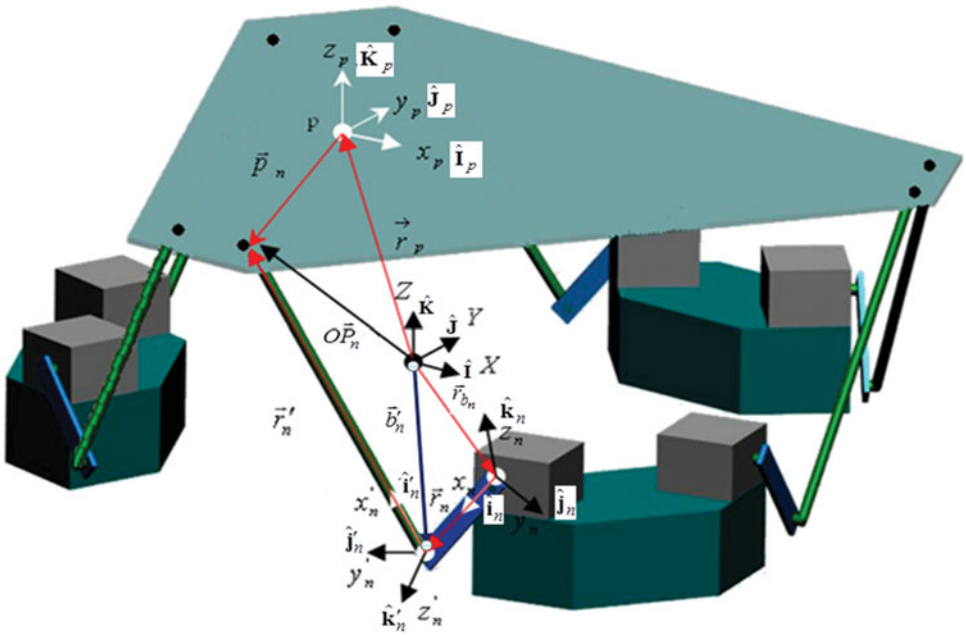

Fig. 2. 6-DOF model of the parallel Stewart manipulator.

to the inertial frame (X, Y, Z) is thus given by the following rotation matrix:

$$\mathbf{R}_P^I = \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\varphi & -S_\varphi \\ 0 & S_\varphi & C_\varphi \end{bmatrix} \times \begin{bmatrix} C_\psi & -S_\psi & 0 \\ S_\psi & C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} C_\theta C_\psi + S_\theta S_\varphi S_\psi & -S_\psi C_\theta + S_\theta S_\varphi C_\psi & S_\theta C_\varphi \\ C_\varphi S_\psi & C_\varphi C_\psi & -S_\varphi \\ -S_\theta C_\psi + C_\theta S_\psi S_\varphi & S_\theta S_\psi + C_\theta S_\varphi C_\psi & C_\theta C_\varphi \end{bmatrix} \tag{1}$$

where C and S denote cosine and sinusoid functions, respectively. Bottom links possess one rotational degree of freedom, while upper links hold twice. The rotation sequence of the bottom links starts by rotating around the z-axis with a fixed posture angle α_n, followed by a variable β_n rotation about y_n-axis. The transformation matrix, $\mathbf{R}_n^I$, from frame (x_n, y_n, z_n), $n = 1, 2 \ldots 6$ to the inertial frame (X, Y, Z) is obtained as

$$\mathbf{R}_n^I = \begin{bmatrix} C_{\alpha_n} & -S_{\alpha_n} & 0 \\ S_{\alpha_n} & C_{\alpha_n} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C_{\beta_n} & 0 & S_{\beta_n} \\ 0 & 1 & 0 \\ -S_{\beta_n} & 0 & C_{\beta_n} \end{bmatrix} = \begin{bmatrix} C_{\alpha_n} C_{\beta_n} & -S_{\alpha_n} & C_{\alpha_n} S_{\beta_n} \\ S_{\alpha_n} C_{\beta_n} & C_{\alpha_n} & S_{\alpha_n} S_{\beta_n} \\ -S_{\beta_n} & 0 & C_{\beta_n} \end{bmatrix} \quad n = 1, 2, \ldots, 6 \tag{2}$$

The rotation sequence of the upper links starts from rotating around the z_n-axis by an angle γ_n, followed by a rotation about the y'_n-axis by an angle λ_n. The transformation matrices from (x'_n, y'_n, z'_n), $n = 1, 2 \ldots 6$ to the corresponding (x_n, y_n, z_n), $n = 1, 2 \ldots 6$ and then with respect to the absolute frame (X, Y, Z) are given as

$$\mathbf{R}_{n'}^n = \begin{bmatrix} C_{\gamma_n} & -S_{\gamma_n} & 0 \\ S_{\gamma_n} & C_{\gamma_n} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} C_{\lambda_n} & 0 & S_{\lambda_n} \\ 0 & 1 & 0 \\ -S_{\lambda_n} & 0 & C_{\lambda_n} \end{bmatrix} = \begin{bmatrix} C_{\gamma_n} C_{\lambda_n} & -S_{\gamma_n} & C_{\gamma_n} S_{\lambda_n} \\ S_{\gamma_n} C_{\lambda_n} & C_{\gamma_n} & S_{\gamma_n} S_{\lambda_n} \\ -S_{\lambda_n} & 0 & C_{\lambda_n} \end{bmatrix} \quad n = 1, 2, \ldots, 6 \tag{3}$$

$$\mathbf{R}_{n'}^I = \mathbf{R}_n^I \mathbf{R}_{n'}^n \tag{4}$$

By referring to Fig. 2, the constraint equations, kinematically relating the variables to each other, can be written as follows:

$$(\vec{\mathbf{r}}_p - \vec{\mathbf{r}}_{bn}) + (\vec{\mathbf{p}}_n - \vec{\mathbf{r}}_n - \vec{\mathbf{r}}'_n) = \mathbf{0} \tag{5}$$

where $\vec{\mathbf{r}}_p$ and $\vec{\mathbf{r}}_{bn}$, respectively, denote position vectors of the platform center and revolute joints expressed in inertial frame (X, Y, Z). $\vec{\mathbf{p}}_n$ is the position vector of universal joints expressed in body frame (x_p, y_p, z_p), $\vec{\mathbf{r}}_n$ and $\vec{\mathbf{r}}'_n$ are the position vectors of bottom and upper link's extremities, respectively, in (x_n, y_n, z_n), $n = 1, 2 \ldots 6$ and (x'_n, y'_n, z'_n), $n = 1, 2 \ldots 6$ coordinate systems.

Equation (5) can be rewritten in coordinate form with respect to the inertial frame (X, Y, Z):

$$(\mathbf{r}_p - \mathbf{r}_{b_n}) + (\mathbf{R}^I_p \mathbf{p}_n - \mathbf{R}^I_n \mathbf{r}_n - \mathbf{R}^I_{n'} \mathbf{r}'_n) = \mathbf{0}\, n = 1, 2, \ldots, 6 \tag{6}$$

where components of $\mathbf{r}_p, \mathbf{r}_{bn}, \mathbf{p}_n, \mathbf{r}_n,$ and $\mathbf{r}'_n$ are represented as

$$\mathbf{r}_p = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix}, \mathbf{r}_{bn} = \begin{bmatrix} X_b \\ Y_b \\ 0 \end{bmatrix}, \mathbf{p}_n = \begin{bmatrix} x_{pn} \\ y_{pn} \\ 0 \end{bmatrix}, \mathbf{r}_n = \begin{bmatrix} l_n \\ 0 \\ 0 \end{bmatrix}, \mathbf{r}'_n = \begin{bmatrix} l'_n \\ 0 \\ 0 \end{bmatrix} \quad n = 1, 2, ..., 6 \tag{7}$$

Equation (6) is obtained in the form of 18 algebraic equations as

$$f_i(\mathbf{q}) = \mathbf{0} i = 1, 2, \ldots, 18 \tag{8}$$

where $\mathbf{q}$ is the generalized coordinate vector constituted of

$$\mathbf{q} = \left[X_p Y_p Z_p\, \theta\, \phi\, \psi\;\; \beta_1\, \beta_2\, \beta_3\, \beta_4\, \beta_5\, \beta_6\;\; \gamma_1\, \gamma_2\, \gamma_3\, \gamma_4\, \gamma_5\, \gamma_6\, \lambda_1\, \lambda_2\, \lambda_3\, \lambda_4\, \lambda_5\, \lambda_6\right]^{\mathrm{T}} \tag{9}$$

The forward kinematic of the mechanism is obtained by integrating with respect to time the kinematic constraints purposely cast in a differential form but inverse kinematics can be solved algebraically as well as via integration. The explicit solution to the inverse kinematic problem can be obtained from the constraint equations formed by the closed chain of links. It suffices to obtain the length of ith upper link (i.e. l'_n) in terms of kinematic variables related to each leg:

$$\left\| \vec{\mathbf{OP}}_n - \vec{\mathbf{b}}'_n \right\| = l'_n \qquad n = 1, 2, \ldots, 6, \quad \mathbf{OP}_n = \begin{bmatrix} P_{nx} \\ P_{ny} \\ P_{nz} \end{bmatrix} \tag{10}$$

As shown in Fig. 2, $\vec{\mathbf{OP}}_n$ and $\vec{\mathbf{b}}'_n$ are, respectively, position vectors of the universal and spherical joints in the inertial frame. By doing some algebraic manipulations, the initial configuration variables in terms of inputs result to

$$\alpha_n = \tan^{-1}\left(\frac{\vec{\mathbf{r}}_{b_n} \cdot \hat{\mathbf{J}}}{\vec{\mathbf{r}}_{b_n} \cdot \hat{\mathbf{I}}}\right) \qquad n = 1, 2, \ldots, 6 \tag{11}$$

$$\beta_n = \tan^{-1}\left(\frac{A \pm \sqrt{(B^2 - 4AC)}}{2A}\right) \qquad n = 1, 2, \ldots, 6 \tag{12}$$

$$\gamma_n = \tan^{-1}\left(\frac{\vec{\mathbf{r}}_n \cdot \hat{\mathbf{j}}_n}{\vec{\mathbf{r}}_n \cdot \hat{\mathbf{i}}_n}\right) \tag{13}$$

$$\lambda_n = \tan^{-1}\left(\frac{\vec{\mathbf{r}}'_n \cdot \hat{\mathbf{k}}'_n}{\sqrt{\left(\vec{\mathbf{r}}'_n \cdot \vec{\mathbf{j}}'_n\right)^2 + \left(\vec{\mathbf{r}}'_n \cdot \vec{\mathbf{i}}'_n\right)^2}}\right) \tag{14}$$

where

$$A = 2l_n \cos(\alpha_n)X_b + 2l_n \sin(\alpha_n)Y_b + 2P_{nx}l_n \cos(\alpha_n) + 2l_nP_{ny} \sin(\alpha_n) \tag{15}$$

$$B = 2l_nP_{nz} + 2Z_bl_n \tag{16}$$

$$C = l_n'^2 - l_n^2 - P_{nx}^2 - P_{ny}^2 - P_{nz}^2 - X_b^2 - Y_b^2 - Z_b^2 + 2P_{nx}X_b + 2P_{ny}Y_b + 2P_{nz}Z_b \tag{17}$$

and where l_n is the length of ith bottom link and X_b, Y_b, and Z_b are the position coordinates of the revolute joints in the inertial frame (X, Y, Z).

By differentiating Eq. (8), the equations can be written in matrix form as

$$\mathbf{A}\dot{\mathbf{q}} = \mathbf{0} \tag{18}$$

where $\mathbf{A} = \frac{\partial f}{\partial \mathbf{q}}$ is the Jacobian matrix. By splitting the generalized coordinate vector $\mathbf{q}$ into $\mathbf{u} = [X_p\, Y_p\, Z_p\, \theta\ \varphi\ \psi]^{\mathrm{T}}$, $\mathbf{V}_1 = [\beta_1\, \beta_2\, \beta_3\, \beta_4\, \beta_5\, \beta_6]^{\mathrm{T}}$, $\mathbf{V}_2 = [\gamma_1\, \gamma_2\, \gamma_3\, \gamma_4\, \gamma_5\, \gamma_6\, \lambda_1\, \lambda_2\, \lambda_3\, \lambda_4\, \lambda_5\, \lambda_6]^{\mathrm{T}}$ arrays and assuming that A can be partitioned correspondingly, we obtain

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{V}}_2 \end{bmatrix} = -\begin{bmatrix} \mathbf{A}_u & \mathbf{A}_{v2} \end{bmatrix}^{-1} \mathbf{A}_{v1}\dot{\mathbf{V}}_1 \tag{19}$$

In forward kinematics, taking integral from Eq. (19) results in output arrays $\mathbf{u}$ and $\mathbf{V}_2$ whenever the input array $\mathbf{V}_1$ is given as a function of time. Whenever the array $\mathbf{u}$ is given as a function of time, the arrays $\mathbf{V} = [\mathbf{V}_1,\ \mathbf{V}_2]^{\mathrm{T}}$ can be obtained by taking integral from the following equation:

$$\dot{\mathbf{V}} = -\mathbf{A}_v^{-1}\mathbf{A}_u\dot{\mathbf{u}} \tag{20}$$

Initial conditions for Eq. (20) can be obtained from Eqs. (11)—(14). Equation (20) will be utilized in Section 3 for obtaining the reduced-order governing equations.

3. Dynamic Model

Since the parallel robots are mostly used in applications requiring high accuracy and high-speed motions, deriving the equations of motion through an accurate model of the robot is primarily important. In recent years, many research works have been conducted on the dynamics of parallel manipulators,[28,29] with consideration on computational efficiency and accuracy in the presence of kinematic constraints. In this section, the equations of motion of the Stewart platform are obtained through a Lagrangian formulation.[28] The detailed expressions of derived equations are given in the appendix. Cutting the parallel manipulator at their common point, one arrives to an open-chain system including six independent serial manipulators. The dynamic model of the whole parallel manipulator can be formulated by combining the dynamics of those serial manipulators under the closed-loop constraints. The governing equation of the constrained dynamical systems can be expressed typically as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{A}^{\mathrm{T}}\rho + \mathbf{B}\mathbf{\Lambda} = \mathbf{0} \tag{21}$$

In the above equations, $\mathbf{M} = \mathbf{M}(\mathbf{q})$ is the generalized inertia matrix, $\mathbf{A} = \mathbf{A}(\mathbf{q})$ is the Jacobian matrix of constraints, $\mathbf{B} = \mathbf{B}(\mathbf{q})$ is the coefficient matrix of the external forces, $\mathbf{h} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ is the nonlinear terms vector, ρ is Lagrange's coefficient vector, and $\mathbf{\Lambda} = [\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6]^{\mathrm{T}}$ is the vector of actuation torques. For simulation and control design processes, independent equations of motion are required. Moreover, it is necessary that the Lagrange's coefficient vector be removed from equations. The standard approach consists of projecting the equations on the space tangent to the constrained manifold by pre-multiplying Eq. (21) into the orthogonal complement of matrix $\mathbf{A}^{\mathrm{T}}$ (finding the null-space of matrix $\mathbf{A}$), yielding to equations freed of Lagrange's multipliers:[30]

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) + \tilde{\mathbf{B}}\mathbf{\Lambda} = 0 \tag{22}$$

In the above equations, $\tilde{\mathbf{M}} = (\mathbf{A^c})^{\mathbf{T}}\mathbf{M}$, $\tilde{\mathbf{B}} = (\mathbf{A^c})^{\mathbf{T}}\mathbf{B}$, $\tilde{\mathbf{h}} = (\mathbf{A^c})^{\mathbf{T}}\mathbf{h}$. $\mathbf{A^c}$ is the complementary orthogonal matrix (i.e. $\mathbf{AA^c} = 0$). The inverse dynamics problem is to determine the external actuation vector or $\mathbf{\Lambda}$ in terms of the command motion (i.e. $\mathbf{q}(t)$). In the forward dynamic problem, the matrix $\tilde{\mathbf{M}}$ and the vector $\mathbf{q}$ in Eq. (22) can be partitioned according to pre-defined task variables $\mathbf{u}$ and joint variables $\mathbf{V}$. Using constraint equations Eq. (20) in combination with Eq. (22), a new form of the equations can be derived in which the second derivative of the dependent variables is replaced in terms of the independent acceleration variables. As a first step, matrix $\tilde{\mathbf{M}}$ and vector $\mathbf{q}$ in Eq. (22) have to be partitioned accordingly:

$$\tilde{\mathbf{M}}_u\ddot{\mathbf{u}} + \tilde{\mathbf{M}}_v\ddot{\mathbf{V}} + \tilde{\mathbf{h}}(\mathbf{q},\dot{\mathbf{q}}) + \tilde{\mathbf{B}}\mathbf{\Lambda} = \mathbf{0} \tag{23}$$

Then, by differentiating Eq. (20) and substituting in Eq. (23), equations will yield to the following compact form:

$$\bar{\mathbf{M}}\ddot{\mathbf{u}} + \bar{\mathbf{h}}(\mathbf{q}, \dot{\mathbf{q}}) = \bar{\mathbf{B}}\mathbf{\Lambda} \tag{24}$$

where $\bar{\mathbf{M}} = \tilde{\mathbf{M}}_u - \tilde{\mathbf{M}}_v\mathbf{A}_v^{-1}\mathbf{A}_u$, $\bar{\mathbf{h}} = \tilde{\mathbf{h}} - \tilde{\mathbf{M}}_v\mathbf{A}_v^{-1}\dot{\mathbf{A}}\dot{\mathbf{q}}$, $\bar{\mathbf{B}} = \tilde{\mathbf{B}}$.

Considering Eq. (24), the solution to the forward dynamics problem can be attained via integrating those equations for the six independent variables of the system. Through them, the 18 remaining variables of the mechanism can be calculated via forward and inverse kinematics relations.

3.1. Dynamic model validation

Any mechanical system without any dissipative agent conserves energy over time or is increasing/decreasing to the amount of work done on it in this interval. In order to verify the correctness of the equations of motion, an indirect way is to check whether or not this conservation principle applies. In the present case, validation of the dynamic model of the Stewart platform can be verified by investigating whether the work integration expression until time *t,*

$$W = \int_{t_0}^{t} \tau\, \dot{q}\, dt \tag{25}$$

is balanced with the sum of kinetic and potential energies at that instant:

$$\begin{aligned} &T = T_{\text{platf}} + T_{\text{links}}, \qquad U = U_{\text{platf}} + U_{\text{links}}, \\ &T_{\text{platf}} = \tfrac{1}{2}m_pV_p^2 + \tfrac{1}{2}\omega_p^T I_p\omega_p, \quad T_{\text{links}} = \tfrac{1}{2}\sum_{i=1}^{N} m_iV_i^2 + \tfrac{1}{2}\sum_{i=1}^{N}\omega_i^T I_i\omega_i \end{aligned} \tag{26}$$

where T_p and U_p are the kinetic and potential energy of the moving plate and T_{links} and U_{links} are the kinetic and potential energy of the links, respectively. The result presented in Fig. 3 is comparing the evolution of those scalar values with each other. The fluctuation in energy level is due to the work accomplished by the actuators in following the commands consisting of twisting motion of the platform at a constant maintained elevation, in a repetitive manner as shown in Fig. 4. The negligible difference between work and total energy verifies that the work–energy principle is fulfilled, validating the model correctness. The trend has been repeated for few cycles to make sure that numerical error accumulation is not pronounced.

4. Nonlinear PD Compensator in Task Space

Model-based controllers such as computed torque attempt to linearize the system through feedback of the nonlinear terms. The remaining tracking error due to various sources of error is eliminated by a linear PD control. However, in practice, robustness against modeling error or external disturbance depends on the extent of uncertainty. The acceptable level of error in pose variables and derivatives is determined by the consequence of such discrepancies in flight maneuvers' accuracy which is much more pronounced in real flight conditions because of its effects on induced aerodynamics forces magnitude. Hence, the aptitude in following commands with the highest fidelity cannot be underestimated but the sensitivity depends particularly on the flight speed. Other factors such as

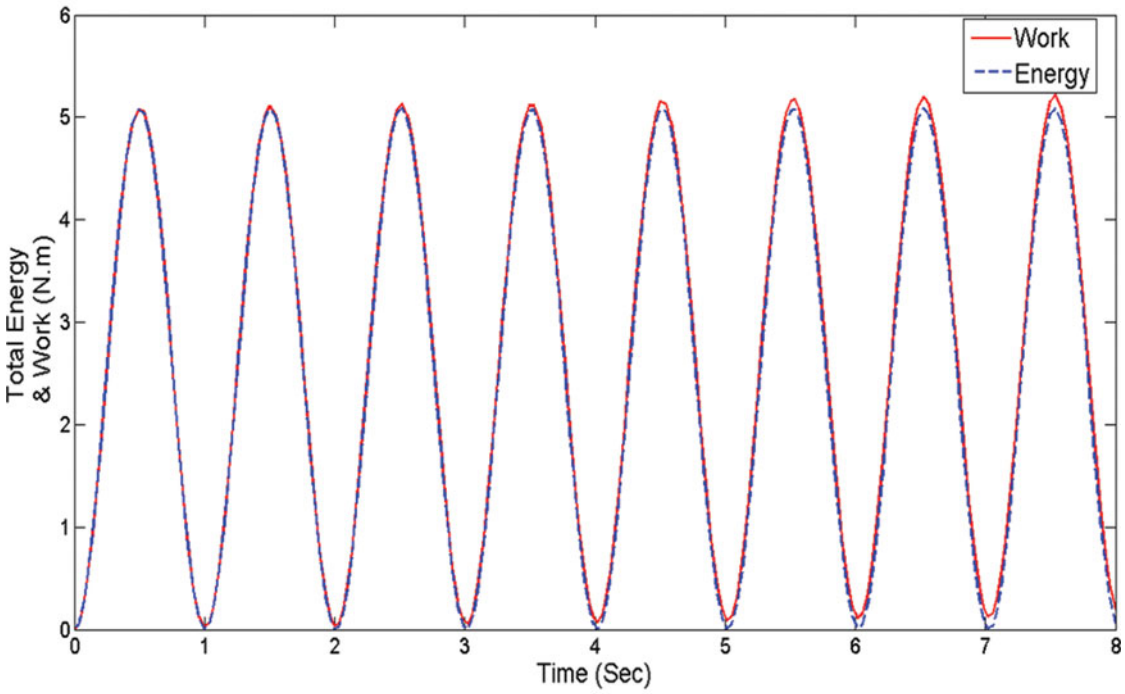

Fig. 3. Comparison of total energy and work evolution.

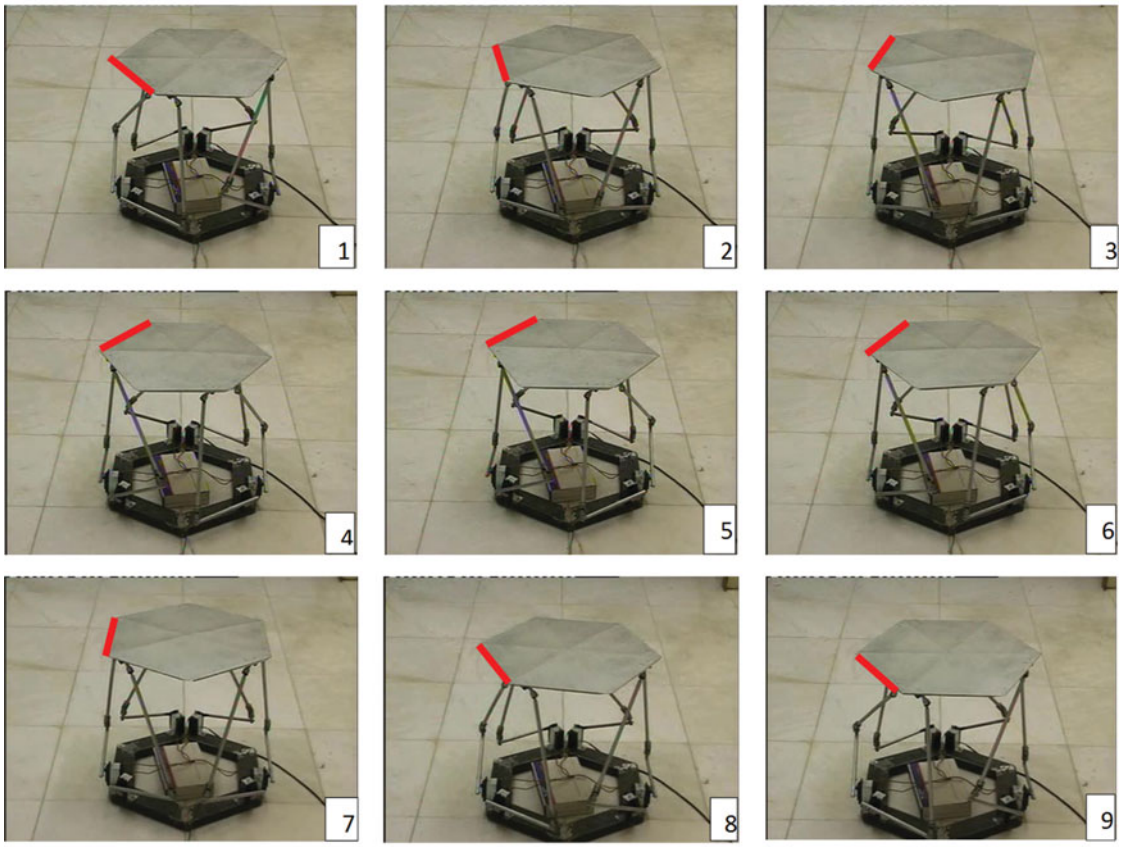

Fig. 4. Snapshots of the twisting motion of the platform about the z-axis at equal intervals of time (images 1–5 represent the first phase of motion and 6–9 its reverse).

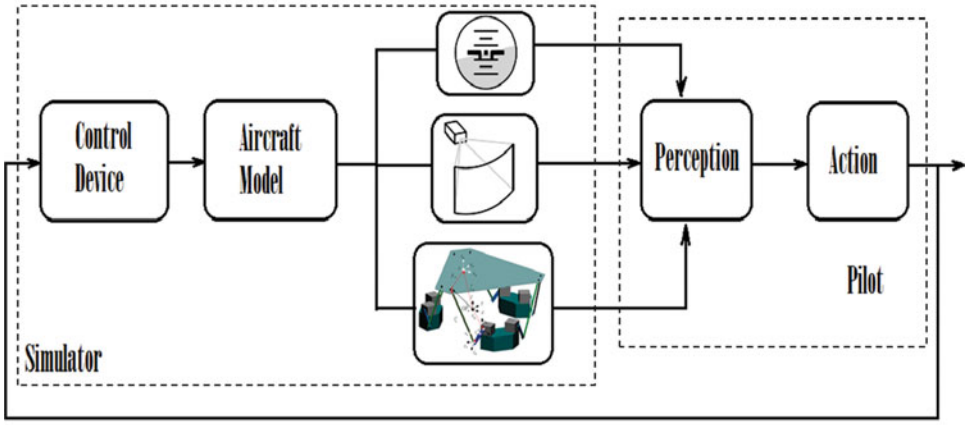

Fig. 5. Block diagram of Stewart platform with pilot in the loop.

motion and vision clues can help the pilot in reestablishing from stall to steady flight conditions. Figure 5 illustrates the pilot-in-the-loop block diagram which is considered here.

In order to improve the control accuracy and disturbance rejection ability, a combination of the NPD control with the conventional control strategies is employed on the platform. It is important to note that the method will be assorted to constrained dynamic systems based on the independent reduced form of governing equations.

In traditional computed torque controller (CTC), one term is dedicated to the dynamics compensation defined by the desired trajectory and the second term is devoted to tracking error elimination. Reconstituted, the input control torque results

$$\mathbf{\Lambda} = \bar{\mathbf{B}}^{-1}\left\{\bar{\mathbf{M}}\left[\ddot{\mathbf{u}}_d + K_v.(\dot{\mathbf{u}}_d - \dot{\mathbf{u}}) + K_p.(\mathbf{u}_d - \mathbf{u})\right] + \bar{\mathbf{h}}\right\} \tag{27}$$

Indeed, this control law expressed in task space is formatted to fit in Eq. (24) of Stewart platform, which upon substitution and noting that $\bar{\mathbf{M}}$ is a positive definite matrix yields to

$$\begin{aligned} &\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \\ &\mathbf{e} = \mathbf{u}_d - \mathbf{u} \end{aligned} \tag{28}$$

where $\mathbf{u}$ is the task space vector obtained from the actuated joint space state $[\beta_1\ \beta_2\ \beta_3\ \beta_4\ \beta_5\ \beta_6]^{\mathrm{T}}$ and $\mathbf{u}_d$ is the desired task space vector. Root locus and state space feedback linearization are common methods which are used to design the controller. However, as mentioned, the CTC is not robust against uncertain factors, such as modeling error and friction. To overcome this problem, the nonlinear computed torque (NCT) controller is designed by replacing the linear PD internal loop controller in CTC with the NPD algorithm. In simple terms, the NPD controller has a similar structure as its linear counterpart, namely any control structure of the form

$$\tau\,(t) = k_p\,(.)\,e\,(t) + k_d\,(.)\,\dot{e}\,(t) \tag{29}$$

where $k_p(.)$ and $k_d(.)$ are state-dependent or time-varying proportional and derivative gains. The superiority of NPD controller in tracking and disturbance rejection relies on its generality compared with the linear PD controllers as can be obviously recognized through its definition:[2]

$$k_p\,(e) = \begin{cases} k_p|e|^{\alpha_1 - 1} & |e| > \delta_1 \\ k_p{\delta_1}^{\alpha_1 - 1} & |e| \le \delta_1 \end{cases} \tag{30}$$

$$k_v\,(\dot{e}) = \begin{cases} k_v|\dot{e}|^{\alpha_2 - 1} & |\dot{e}| > \delta_2 \\ k_v{\delta_2}^{\alpha_2 - 1} & |\dot{e}| \le \delta_2 \end{cases} \tag{31}$$

where α_1 and α_2 can be determined in the interval [0.5, 1.0] and [1.0, 1.5], respectively. The particular choice of exponents changes the response characteristics but is proven to keep the system stable.

By replacing the gains with state-dependent gains in the control law (Eq. (27)), we get

$$\mathbf{\Lambda} = \bar{\mathbf{B}}^{-1}\left\{\bar{\mathbf{M}}\left[\ddot{\mathbf{u}}_d + \mathbf{K}_v(\dot{\mathbf{e}})(\dot{\mathbf{u}}_d - \dot{\mathbf{u}}) + \mathbf{K}_p(\mathbf{e})(\mathbf{u}_d - \mathbf{u})\right] + \bar{\mathbf{h}}\right\} \tag{32}$$

and inserting it into the dynamic model Eq. (22), the closed-loop system is obtained:

$$\ddot{\mathbf{e}} + \mathbf{K}_v(\dot{\mathbf{e}})\dot{\mathbf{e}} + \mathbf{K}_p(\mathbf{e})\mathbf{e} = 0 \tag{33}$$

where $\mathbf{K}_v(\dot{\mathbf{e}})$ and $\mathbf{K}_p(\mathbf{e})$ are defined purposely as

$$\mathbf{K}_v(\dot{\mathbf{e}}) = \mathrm{diag}(k_{v1}(\dot{e}_1), k_{v2}(\dot{e}_2), k_{v3}(\dot{e}_3), k_{v4}(\dot{e}_4), k_{v5}(\dot{e}_5), k_{v6}(\dot{e}_6)) \tag{34}$$

$$\mathbf{K}_p(\mathbf{e}) = \mathrm{diag}(k_{p1}(e_1), k_{p2}(e_2), k_{p3}(e_3), k_{p4}(e_4), k_{p5}(e_5), k_{p6}(e_6)) \tag{35}$$

where k_{vi}, k_{pi}, $i = 1, 2, \ldots 6$ are positive constant gains and e_i, $i = 1 \ldots 6$ determine the task space error variables. In the present problem, the parameters $\alpha_1, \alpha_2, \delta_1, \delta_2, k_{pi}$, and k_{vi} are tuned by GA. The main point of the controller design procedure is to determine the parameters such that the closed-loop control system, Eq. (33), is stable. Thus, a Lyapunov function candidate[2] is chosen as

$$V(e, \dot{e}) = \frac{1}{2}\dot{\mathbf{e}}^{\mathrm{T}}\dot{\mathbf{e}} + \int_0^e |\xi|^{\mathrm{T}}\mathbf{K}_p(\xi)\mathbf{d}\xi \tag{36}$$

where $\int_0^e |\xi|^{\mathrm{T}}\mathbf{K}_p(\xi)\mathbf{d}\xi = \sum_{i=1}^{6}\int_0^{e_i} |\xi_i| k_{pi}(\xi_i) d\xi_i$. Obviously, the first term in Eq. (36) is positive definite and the second term is positive definite according to the following lemma.

Lemma:[27] Consider the continuous diagonal matrix $\mathbf{K}_p : R^2 \to R^{6\times 6}$ and assume that there exist class K functions $\alpha_i(.)$ such that $|x|k_{pi}(x) \geq \alpha_i(|x|)$, $x \in R$, $i = 1, 2, \ldots 6$, then $\int_0^e |\xi|^{\mathrm{T}}\mathbf{K}_p(\xi)\mathbf{d}\xi > 0$, $\forall e \neq 0 \in R^2$ and $\int_0^e |\xi|^{\mathrm{T}}\mathbf{K}_p(\xi)\mathbf{d}\xi \to \infty$ *as* $|e| \to \infty$

Differentiating $V(e, \dot{e})$ with respect to time yields

$$\dot{V}(e, \dot{e}) = \dot{\mathbf{e}}^{\mathrm{T}}\ddot{\mathbf{e}} + \mathbf{e}^{\mathrm{T}}\mathbf{K}_p(\mathbf{e})\dot{\mathbf{e}} \tag{37}$$

Multiplying both sides of the closed-loop system Eq. (33) by $\dot{\mathbf{e}}^{\mathrm{T}}$, one obtains $\dot{\mathbf{e}}^{\mathrm{T}}\ddot{\mathbf{e}} + \dot{\mathbf{e}}^{\mathrm{T}}\mathbf{K}_v(\dot{\mathbf{e}})\dot{\mathbf{e}} + \dot{\mathbf{e}}^{\mathrm{T}}\mathbf{K}_p(\mathbf{e})\mathbf{e} = 0$. Paying attention to the fact that $\mathbf{K}_p(\mathbf{e})$ is a diagonal matrix and $\dot{\mathbf{e}}^{\mathbf{T}}\mathbf{K}_p(\mathbf{e})\mathbf{e} = \mathbf{e}^{\mathbf{T}}\mathbf{K}_p(\mathbf{e})\dot{\mathbf{e}}$, substituting this last expression into Eq. (37) results in

$$\dot{V}(e, \dot{e}) = -\dot{\mathbf{e}}^{\mathbf{T}}\mathbf{K}_v(\dot{\mathbf{e}})\dot{\mathbf{e}} \tag{38}$$

As $\mathbf{K}_v(\dot{\mathbf{e}})$ is a diagonal positive definite matrix so $\dot{V}$ becomes negative semi-definite. By defining the state-space variables $\binom{e}{\dot{e}}$ and considering Eq. (33) as an autonomous system, $\dot{V} \neq 0$ along any solution of the differential equation except the origin. Hence, there is no invariant set corresponding to $\dot{V} = 0$ and therefore the closed-loop system is asymptotically stable to the origin according to LaSalle's theorem.[31]

4.1. Criterion for pilot's sensorial feedback

One of the objectives of this research is to search optimal values for the controller gains $k_p, k_v, \alpha_1, \alpha_2, \delta_1$, and δ_2. The controller gains are optimally tuned based on the position and orientation deviation of the moving platform measured during a determined lapse of time. Optimal gain tuning can improve the tracking performance and accuracy of the manipulator but has to repeat the optimization process for each maneuver profile. According to that the flight simulator is expected to emulate the physical conditions of flight as realistic as possible, a good practice may consist to stimulate the pilot's sensorial feedback of translational and rotational accelerations, bringing him clues to react

handily in different eventual scenarios. For achieving such purpose, the sum of the weighted integral of acceleration error is considered as cost function which is defined as

$$F_{\text{cost}} = \sum_{n=1}^{6} w_n \int_0^{t\text{span}} \left| \frac{d^2 e_n(t)}{dt^2} \right| dt \quad n = 1, 2, \ldots, 6 \tag{39}$$

where e_i , $i = 1, 2 \ldots 6$ are defined as

$$e_1 = x_d - x,\ e_2 = y_d - y,\ e_3 = z_d - z,\ e_4 = \theta_d - \theta,\ e_5 = \phi_d - \phi,\ e_6 = \psi_d - \psi \tag{40}$$

Scaling factors w_n are tuned to normalize the acceleration error in order that the summation of different physical values becomes meaningful to minimize.

4.2. Optimal weight tuning via genetic algorithm

Evolutionary algorithms (EAs) have found application as optimization techniques in different areas of engineering.[32] These algorithms are derivative free and stochastic-based methods being inspired from the theory of natural evolution.[33] In the present work, GA as one ramification of EAs algorithms is utilized to optimize tracking performance with respect to controller parameters.[23] Fitness evaluation, selection, cross-over, and mutation are GA operators which are applied to individuals (i.e. solutions) of population repeatedly in the main loop of GA until the final generation of individuals is generated. This final population breeds the local optimal solution in an optimization problem. In the conventional process of optimization, there is no indication that an absolute extremum state has been reached. In the present case, there is also no clue to find out whether the problem is convex. Nonetheless an obvious advantage in this approach compared to gradient descent methods is that the search is not based on gradient information, and has therefore, no requirements on the continuity or convexity of space. In fact, by introducing gene mutation, the odds of getting stuck in a local minimum is reduced as this anomaly adds to the diversity of the specimens and by this way, a number of points out of the vicinity range will also be systematically investigated to get out of eventual local minima.

Among real and binary-coded types of GA, the former one is used to encode the PD tuning parameters in each chromosome (individual) and the value of Eq. (33) serves as fitness function.

5. Results and Discussion

The optimal process of tuning controller gains is evaluated according to values tabulated in Tables I and II, specifying the attributed GA algorithm characteristics and the Stewart platform parameters, respectively. Results contain the linear and nonlinear computed torque controller and those obtained by an adaptive neuro-fuzzy controller.[26] The performance of the controller is tested for three desired trajectories. The parametric equations of these trajectories are considered as

$$\begin{aligned} &\text{Path 1 :} \\ &x_d = y_d = 0, \\ &z_d = z_0 + a\sin(\omega t), \quad z_0 = 0.346,\ a = 0.1,\ \omega = \pi(\text{rad/s}), \\ &\theta = \phi = \psi = 0 \end{aligned} \tag{41}$$

$$\begin{aligned} &\text{Path 2 :} \\ &x_d = 0.01 - 0.01 \times \cos(\pi t), \\ &y_d = 0.01 \times \sin(\pi t), \\ &z_d = z_0, \qquad z_0 = 0.346, \\ &\theta = \phi = \psi = 0 \end{aligned} \tag{42}$$

$$\begin{aligned} &\text{Path 3 :} \\ &x_d = 0,\ y_d = 0,\ z_d = z_0, \qquad z_0 = 0.346, \\ &\theta = (\pi/36) \times \sin(\pi t), \\ &\phi = (\pi/36) \times \sin(2\pi t), \\ &\psi = 0 \end{aligned} \tag{43}$$

Table I. GA settings applied in this study.

Selection type	Cross-over type	Mutation type	Cross-over rate	Mutation probability	Population size	Max. no. of generation
Tournament	Arithmetic	Uniform	0.75	0.02	100	100

Table II. System's parameters.

Mass of bottom link	$m = 0.039$(kg)	Length of bottom link	$l = 0.18$(m)
Mass of upper link	$m' = 0.065$(kg)	Length of upper link	$l' = 0.3$(m)
Mass of moving platform		$m_p = 1.5$(kg)	

Table III. Linear torque controller coefficients.

	k_{v1}	k_{p1}	k_{v2}	k_{p2}	k_{v3}	k_{p3}	k_{v4}	k_{p4}	k_{v5}	k_{p5}	k_{v6}	k_{p6}
Path 1	60.55	916.57	74.33	1381.22	93.49	2185.08	38.34	367.44	56.77	805.65	65.62	1076.48
Path 2	150.55	5666.32	129.22	4174.40	160.33	6429.39	76.34	1456.9	186.77	8720.74	197.08	9710.12
Path 3	90.55	2049.82	69.22	1197.80	90.33	2039.84	97.34	2368.68	86.77	1882.24	75.08	1409.24

5.1. Application of the controller

The proposed optimization algorithm is applied to the Stewart platform by generating random initial populations of the controller gains for the linear and nonlinear computed torque controllers. The algorithm converges to optimal gain values which depend on the desired maneuver, as indicated in Tables III and IV for the linear and nonlinear controllers, respectively. The optimal values of controller parameters are given in the following.

Corresponding solution diagrams are depicted in Figs. 6–11. Figure 6 shows the time history of the moving platform coordinates in following trajectory 1, which consists of an ascending/descending maneuver in an oscillating pattern. The order of maximum position error is about 10^{-3}(m) and the order of maximum rotation is 0.1 degree, which is quite adequate for flight simulator applications. Translation in the z-direction perfectly coincides with the desired trajectory.

Figure 7 shows the platform position and rotation time history for trajectory 2. The desired trajectory represents a turning maneuver which is pursued with a maximum deviation equal to 10^{-3}(m) for position and 0.06 (deg) for rotation. The third desired trajectory is defined as a simultaneous rocking motion of the platform about its x and y axes, shown in (Fig. 8). Optimal values of k_v and k_p for NCT controller are presented in Table IV. In comparison to optimal values of CTC cited in Table III, the values of NCT controller in Table IV decreased substantially within the interval (0–100), while also displaying a lower error percentage in trajectory tracking. The same trend is repeated for other trajectories, showing a relative supremacy of the NCT controller in comparison to its linear counterpart on precision enhancement and control effort.

For the sake of completeness, the simulation results of proposed NCT controller on an RSU platform are compared with the simulation results of a neuro-fuzzy adaptive controller that was accomplished in previous works.[26] As shown in Fig. 9, the position error of the platform is presented in following trajectory 1 with two simulated controllers: (a) neuro-fuzzy adaptive controller and (b) NCT controller. Error magnitude of the NCT controller did not change significantly in comparison to neuro-fuzzy adaptive controller. Similar results are obtained for trajectories 2 and 3. However, in comparison to neuro-fuzzy adaptive controller, NCT controller is found more robust against uncertain parameters such as platform mass. Figure 10 shows the simulation results of NCT controller when the moving plate mass is increased from 1.8 kg to 5 kg. It is worth to mention that the neuro-fuzzy adaptive controller[26] could not track the desired trajectory when encountered with a mass discrepancy higher than 3 kg.

Table IV. Nonlinear torque controller coefficients.

	k_{v1}	k_{p1}	k_{v2}	k_{p2}	k_{v3}	k_{p3}	k_{v4}	k_{p4}	k_{v5}	k_{p5}	k_{v6}	k_{p6}	α_1	α_2	δ_1	δ_2
Path 1	62.55	978.12	71.33	1272.00	90.49	2047.1	42.34	448.16	58.77	863.47	67.62	1143.1	0.91	1.46	0.03	0.3
Path 2	54.53	743.92	76.20	1452.30	60.3	909.89	86.34	1863.6	96.8	2341.1	33.01	273.57	0.95	1.21	0.03	0.03
Path 3	50.85	646.43	30.25	228.76	29.85	222.75	32.45	263.25	33.48	280.22	25.08	157.25	0.85	1.47	1.47	0.08

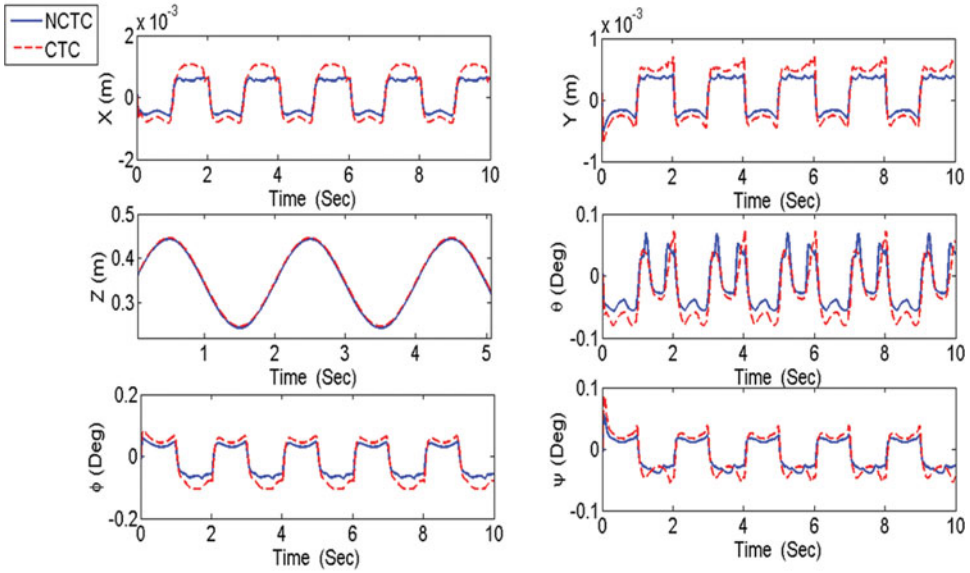

Fig. 6. Position and rotation of moving platform for trajectory 1.

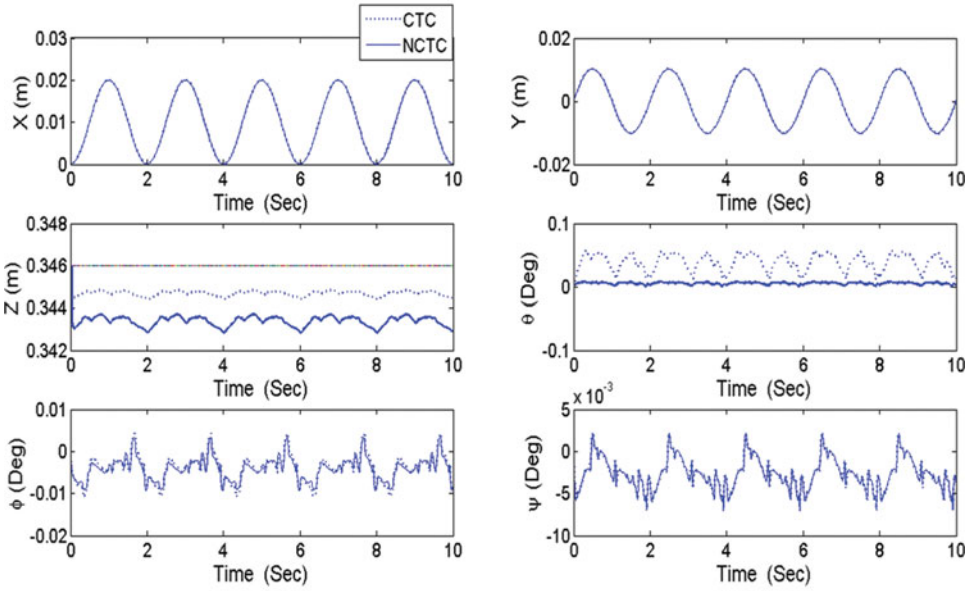

Fig. 7. Position and rotation of moving platform for trajectory 2.

For assessing transient performance in following sudden changing commands, a unit-step elevation is entered as input to the linear and nonlinear computed torque controllers' loop with results compared in Fig. 11.

5.2. Effect of pilot behavior on control

For identifying pilot control behavior, the effect of the pilot has been implemented into the controller loop in order to simulate a more realistic situation. As illustrated in Fig. 5, the scheme indicates that pilots rely both on visual and motion cues to generate commands for the control task, although the latter is barely used in low bandwidth conditions. For the identification of pilot control behavior, the

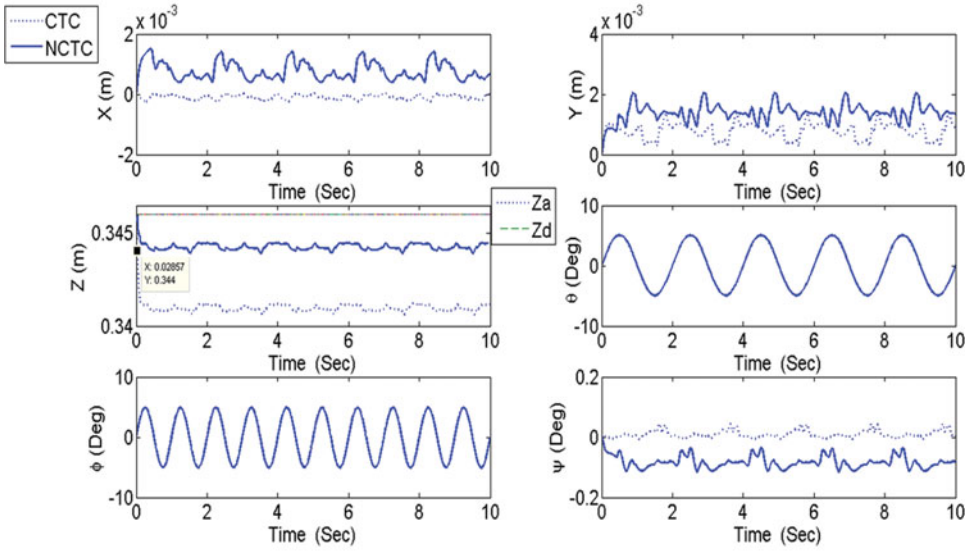

Fig. 8. Position and rotation of moving platform for trajectory 3.

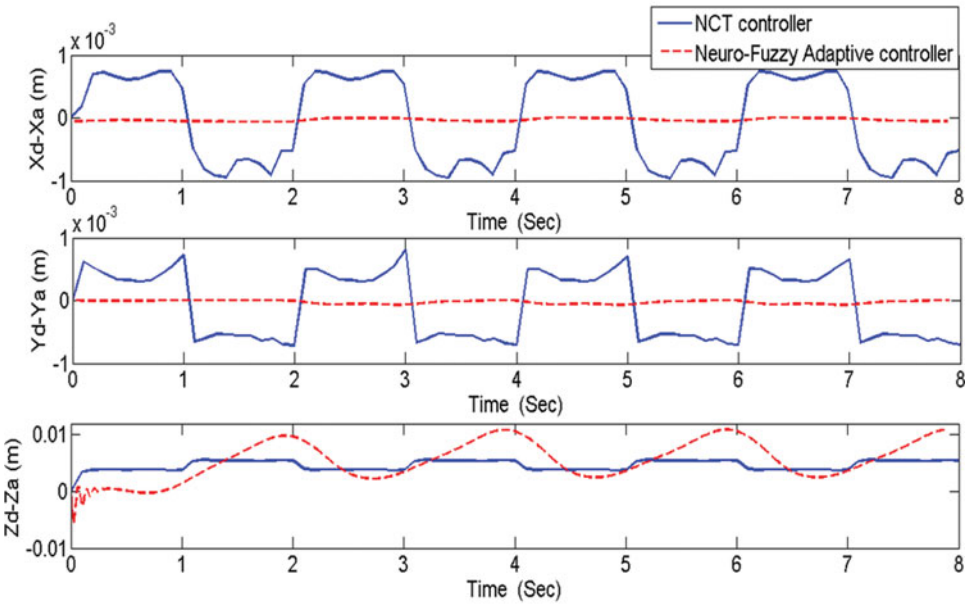

Fig. 9. Position error for RSU platform for neuro-fuzzy adaptive controller and NCT controller.

pilot block representing his visual response to tracking error is denoted byH_e, which is given as

$$H_e = \frac{e^{-Ts}}{t_{\text{lag}}s + 1} \tag{44}$$

where T is defined as the pilot visual perception time delay and the equalization characteristics of the pilot is presented by the lag constant t_{lag} . The model of the visual response is based on the work by Nieuwenhuizen and Bulthoff[1] and is simplified in this paper. The error feedback that is supplied into the controller is obtained and filtered by the pilot block as follows:

$$e = H_e\, e_{\text{real}}, \quad e_{\text{real}} = u_{\text{desired}} - u \tag{45}$$

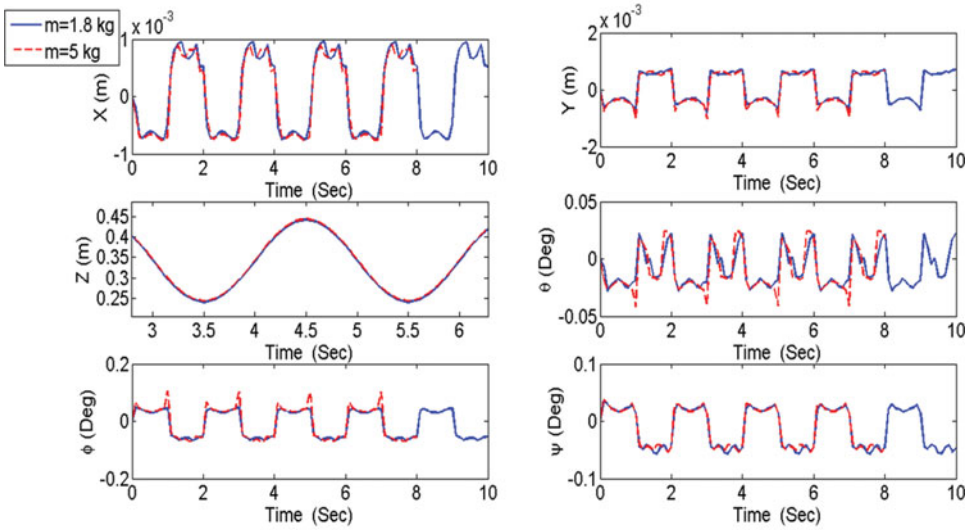

Fig. 10. Position, rotation, and velocity of moving platform for trajectory 1 with different payloads ($m_p = 1.8$ kg and $m_p = 5$ kg).

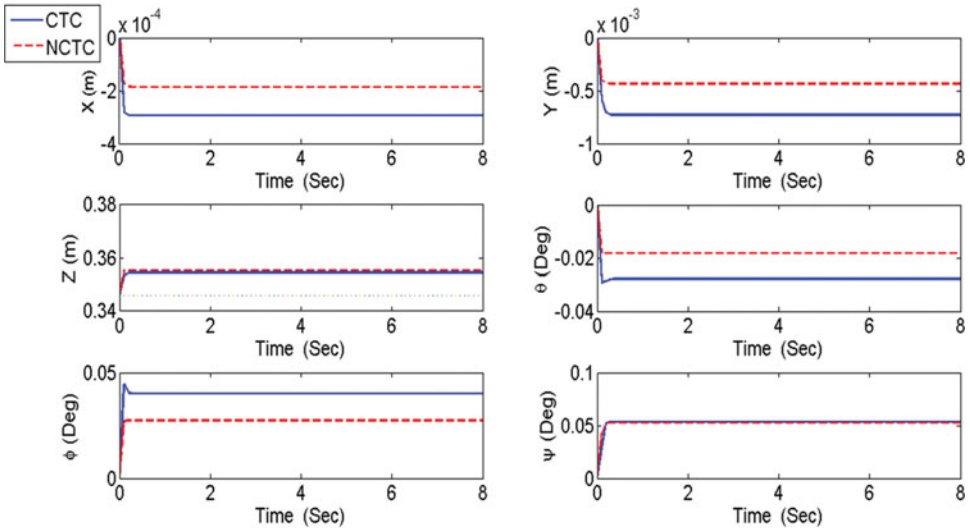

Fig. 11. Position, rotation, and velocity of moving platform in pursuing a step command.

This will permit to understand the role of pilot's reactions against extreme flight conditions and to eventually compensate for the attenuated performance in real application. The simulation results for trajectory 1 are presented in Fig. 12, showing an acceptable level of errors despite erroneous feedbacks.

6. Conclusions

In this paper, an optimal NPD compensated torque controller was designed for a Stewart-type parallel manipulator developed as a flight simulator. The asymptotic stability of the parallel manipulator system controlled by the NCT controller was also proven. Parameters of the controller were optimized using GA algorithm. Cost function was defined as a summation of integral acceleration error. The proposed algorithm obtained optimal values for the linear and nonlinear computed torque controller parameters. Simulation measurements of the controller were presented in numerical results.

The following results are obtained from the numerical results.

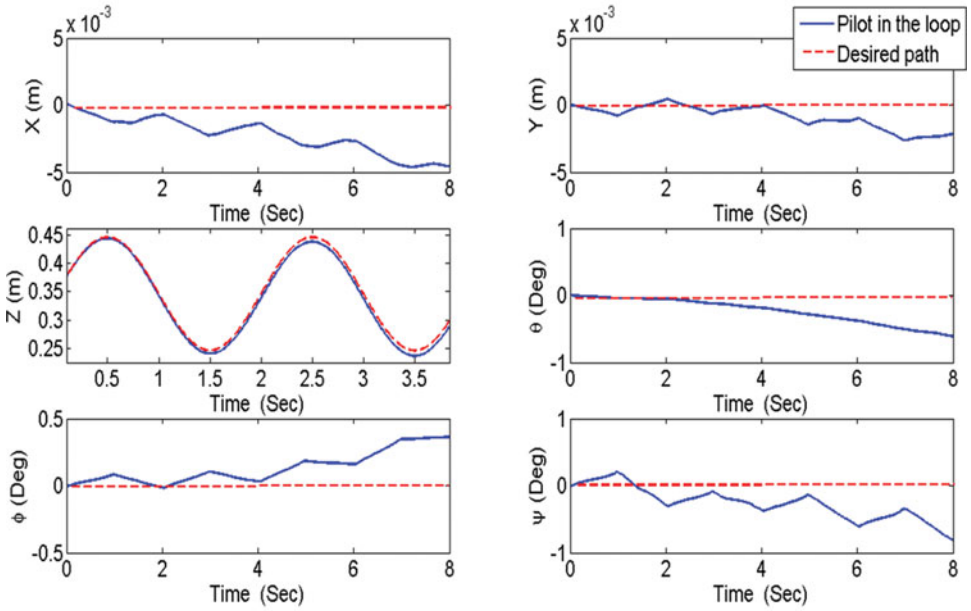

Fig. 12. Position, rotation, and velocity of moving platform with pilot control behavior (trajectory 1).

1. Optimal gain values of NCT controller are found smaller than the optimal values of CT controller, resulting in reduced control effort.
2. Compared with CT controller and neuro-fuzzy adaptive controller, the NCT controller is found robust against the uncertain factors of modeling error. As shown, the controller can still track the desired trajectory even if the platform mass is substantially varied from nominal values.
3. A transfer function has been considered for inserting the pilot in the loop in order to evaluate the effect of human sensory on the tracking performance. The controller was able to compensate for this effect.

References

1. F. M. Nieuwenhuizen and H. H. Bulthoff, "The MPI CyberMotion simulator: A novel research platform to investigate human control behavior," *J. Comput. Sci. Eng.* **7**(2), 122–131 (2013).
2. W. W., Shang and S. Cong, "Nonlinear computed torque control for a high speed planar parallel manipulator," *Mechatronics* **19**(6), 987–992 (2009).
3. D. Stewart, "A platform with six degrees of freedom," *Proc. Inst. Mech. Eng.* **180**(5), 371–386 (1965).
4. J. P. Merlet, *Parallel Robots Solid Mechanics and Its Applications* (Kluwer Academic Publishers, Springer, Netherlands, 2006).
5. M. Schwab, "Hexapod," WIPO Patent No. WO2011089198 (2011).
6. B. Shchokin and F. Janabi-Sharifi, "Design and kinematic analysis of a rotary positioner," *Robotica* **25**(1), 75–85 (2007).
7. L. Campos, F. Bourbonnais, I. A. Bonev and P. Bigras, "Development of a Five-Bar Parallel Robot with Large Workspace," *Proceedings of the ASME 2010 International Design Engineering Technical Conferences*, Montreal, QC, Canada (2010).
8. O. Bohigas, L. Ros and M. Manubens, "A complete method for workspace boundary determination on general structure manipulators," *IEEE Trans. Robot.* **28**(5), 993–1006 (2012).
9. S. Stan, R. Balan, V. Maties and C. Rad, "Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot," *J. Mech.*, **75**(1), 62–66 (2009).
10. H. Kim, Y. Cho and K. Lee, "Robust nonlinear task space control for 6 DOF parallel manipulator," *Automatica* **41**(6), 1591–1600 (2005).
11. I. Davliakos and E. Papadopoulos, "Model-based control of a 6-dof electrohydraulic Stewart-Gough platform," *Mech. Mach. Theory* **43**(11), 1385–1400 (2008).
12. A. Omran, A. Kassem, G. El-Bayoumi and M. Bayoumi, "Mission-based optimal control of Stewart manipulator," *Aircr. Eng. Aerosp. Technol. J.* **81**(3), 147–153 (2009).
13. C. Yang, Q. Huang, H. Jiang, O. Peter and J. Han, "PD control with gravity compensation for hydraulic 6-DOF parallel manipulator," *Mech. Mach. Theory* **45**(4), 666–677 (2010).

14. Y. X. Su, D. Sun, L. Ren, X. Wang and J. K. Mills, "Nonlinear PD Synchronized Control for Parallel Manipulators," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005) pp. 1374–1379, .
15. Y. X. Su, D. Sun, L. Ren, X. Wang and J. K. Mills, "Robust nonlinear task space control for 6 DOF parallel manipulator," *Automatica* **41**(9), 1591–1600 (2005).
16. Y. X. Su, B. Y. Duan and C. H. Zheng, "Nonlinear PID control of a six-DOF parallel manipulator," *IEE Proc. Control Theory Appl.* **151**(1), 95–102 (2004).
17. I. Davliakos and E. Papadopoulos, "Model-based control of a 6-dof electrohydraulic Stewart-Gough platform," *Mech. Mach. Theory* **43**, 1385–1400 (2008).
18. I. Davliakos and E. Papadopoulos, "Model-Based Position Tracking Control for a 6-dof Electrohydraulic Stewart Platform," *Proceeding of the 15th Mediterranean Conference on Control & Automation* (Jul. 27–29, 2007).
19. I. Davliakos and E. Papadopoulos, "Impedance model-based control for an electrohydraulic Stewart platform," *Eur. J. Control* **5**, 560–577 (2009).
20. A. Zubizarreta, M. Marcosa, I. Cabanes and C. Pinto, "A procedure to evaluate extended computed torque control configurations in the Stewart-Gough platform," *Robot. Auton. Syst.* **59**, 770–781 (2011).
21. W. Dongsu and G. Hongbin, "Adaptive sliding control of six-DOF flight simulator motion platform," *Chin. J. Aeronaut.* **20**, 425–433 (2007).
22. Y. Ting, C.-C. Li and T. Van Nguyen, "Composite controller design for a 6DOF Stewart nanoscale platform," *Precis. Eng.* **37**, 671–683 (2013).
23. A. Omran and A. Kassem, "Optimal task space control design of a Stewart manipulator for aircraft stall recovery," *Aerosp. Sci. Technol.* **15**, 353–365 (2011).
24. C. Yang, Q. Huang, H. Jiang, O. Peter and J. Han, "PD control with gravity compensation for hydraulic 6-DOF parallel manipulator," *Mech. Mach. Theory* **45**, 666–677 (2010).
25. H. Azizan, M. Keshmiri and M. Jafarinasab, "Designing a Stable Model-Based Fuzzy Controller for a Novel 6-DOF Parallel Manipulator with Rotary Actuators," Proceedings of the *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2011)*, Budapest, Hungary (Jul. 3–7, 2011).
26. M. Eftekhari, M. Eftekhari and H. Karimpour, "Neuro-fuzzy adaptive control of a revolute Stewart platform carrying payloads of unknown inertia," *Robotica* **33**(9), 2001–2024 (2014). doi: http://dx.doi.org/10.1017/S0263574714001222
27. R. Kelly and C. Ricardo, "A class of nonlinear PD-type controller for robot manipulator," *J. Robot. Syst.* **13**, 793–802 (1996).
28. D. Liang, Y. Song, T. Sun and G. Dong, "Optimum design of a novel redundantly actuated parallel manipulator with multiple actuation modes for high kinematic and dynamic performance," *Nonlinear Dyn.* **83**(1), 631–658 (2016).
29. D. Liang, Y. Song and T. Sun, "Nonlinear dynamic modeling and performance analysis of a redundantly actuated parallel manipulator with multiple actuation modes based on FMD theory," *Nonlinear Dyn.* **89**(1), 391–428 (2017). doi:10.1007/s11071-017-3461-x
30. M. Zefran, "Lagrangian Dynamics," **In:** *Robotics and Automation Handbook* (T. R. Kurfess, ed.) (University of Illinois at Chicago, Chicago, 2005).
31. J. P. LaSalle, "Stability theory for ordinary differential equations," *J. Differ. Equ.* **4**, 57–65 (1968).
32. D. Ashlock, *Evolutionary Computation for Modeling and Optimization* (Springer, Berlin, 2006).
33. C. Darwin, *The Origin of Species* (Dent Gordon, London, 1973).

Appendix

The dynamic equations of the Stewart platform are derived following Lagrangian formulation as

$$\begin{aligned} &\tfrac{d}{dt}\left(\tfrac{\partial L}{\partial \dot{q}_i}\right) - \tfrac{\partial L}{\partial q_i} = Q_{nc,i} + \left(\mathbf{A}^{\mathrm{T}}\rho\right)_i, \qquad i = 1, \ldots, n \\ &Q_{nc,i} = \sum_{j=1}^{N_F} \vec{\mathbf{F}}_{nc,j} \cdot \tfrac{\partial \vec{\mathbf{r}}_j}{\partial q_i} + \sum_{j=1}^{N_\tau} \tau_{nc,j} \cdot \tfrac{\partial \vec{\omega}_j}{\partial \dot{q}_i}, \\ &L = T - U \end{aligned} \tag{A1}$$

where $\mathbf{q}$ is the generalized coordinate vector constituted of

$$\mathbf{q} = \left[X_p Y_p Z_p\ \theta\ \phi\ \psi\ \ \beta_1\ \beta_2\ \beta_3\ \beta_4\ \beta_5\ \beta_6\ \ \gamma_1\ \gamma_2\ \gamma_3\ \gamma_4\ \gamma_5\ \gamma_6\ \lambda_1\ \lambda_2\ \lambda_3\ \lambda_4\ \lambda_5\ \lambda_6\right]^{\mathrm{T}} \tag{A2}$$

The kinetic energy and potential energy T and U of the constrained system, respectively, are evaluated as follows:

$$\begin{aligned} T &= T_p + \sum_{n=1}^{6} T_n + T'_n \\ T &= \tfrac{1}{2}(m_p \vec{\mathbf{V}}_p \cdot \vec{\mathbf{V}}_p + \vec{\omega}_p \cdot \overleftrightarrow{\mathbf{I}}_p \cdot \vec{\omega}_p + \sum_{\mathrm{n}=1}^{6} \left(\vec{\omega}_n \cdot \overleftrightarrow{\mathbf{I}}_n \cdot \vec{\omega}_n + m'_n \vec{\mathbf{V}}'_n \cdot \vec{\mathbf{V}}'_n + \vec{\omega}'_n \cdot \overleftrightarrow{\mathbf{I}}'_n \cdot \vec{\omega}'_n \right) \\ U &= U_p + \sum_{n=1}^{6} U_n + U'_n = m_p g \vec{\mathbf{R}}_p \cdot \hat{\mathbf{K}} + \sum_{n=1}^{6} m_n g \vec{\mathbf{R}}_n \cdot \hat{\mathbf{K}} + m'_n g \vec{\mathbf{R}}'_n \cdot \hat{\mathbf{K}} \end{aligned} \tag{A3}$$

where m_p, m_n, and m'_n are the mass values of the moving platform, lower links, and upper links, respectively. $\vec{\mathbf{R}}_p, \vec{\mathbf{R}}_n$, and $\vec{\mathbf{R}}'_n$ are the position vectors of the center of mass corresponding to the moving platform, lower links, and upper links in the inertial frame, respectively. The linear velocity vectors are derived in inertial frame as

$$\vec{\mathbf{V}}_p = \dot{\vec{\mathbf{R}}}_p, \ \vec{\mathbf{V}}_n = \dot{\vec{\mathbf{R}}}_n, \ \vec{\mathbf{V}}'_n = \dot{\vec{\mathbf{R}}}'_n \tag{A4}$$

Moreover, $\vec{\omega}_p$, $\vec{\omega}_n$, and $\vec{\omega}'_n$ are the angular velocity vectors of the moving platform, lower links, and upper links in related body frames, respectively. The angular velocity vector of the moving platform is obtained as follows:

$$\begin{aligned} \boldsymbol{\Omega} &= \dot{\mathbf{R}}_p^I (\mathbf{R}_p^I)^{\mathrm{T}} = \begin{bmatrix} 0 & -\dot{\theta}\mathrm{S}_\varphi - \dot{\psi} & \dot{\theta}\mathrm{C}_\varphi \mathrm{C}_\psi + \dot{\varphi}\mathrm{S}_\psi \\ \dot{\theta}\mathrm{S}_\varphi + \dot{\psi} & 0 & \dot{\theta}\mathrm{C}_\varphi \mathrm{S}_\psi - \dot{\varphi}\mathrm{C}_\psi \\ -\dot{\theta}\mathrm{C}_\varphi \mathrm{C}_\psi - \dot{\varphi}\mathrm{S}_\psi & \dot{\varphi}C_\psi - \dot{\theta}C_\varphi S_\psi & 0 \end{bmatrix}, \\ \vec{\omega}_p^{\text{Inertial frame}} &= \begin{bmatrix} \Omega_{3,2} & \Omega_{1,3} & \Omega_{2,1} \end{bmatrix}^{\mathrm{T}} \\ \vec{\omega}_p^{\text{Body frame}} &= (\mathbf{R}_p^I)^{\mathrm{T}} \vec{\omega}_p^{\text{Inertial frame}} = \vec{\omega}_p \end{aligned} \tag{A5}$$

$\overleftrightarrow{\mathbf{I}}_p$, $\overleftrightarrow{\mathbf{I}}_n$, and $\overleftrightarrow{\mathbf{I}}'_n$ are the dyadic of the Inertia moment of the moving platform, lower links, and upper links in related body frames, respectively, and are obtained as

$$\mathbf{I}_p = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}, \quad \mathbf{I}'_n = \begin{bmatrix} I'_{xn} & 0 & 0 \\ 0 & I'_{yn} & 0 \\ 0 & 0 & I'_{zn} \end{bmatrix}, \quad \mathbf{I}_n = \begin{bmatrix} I_{xn} & 0 & 0 \\ 0 & I_{yn} & 0 \\ 0 & 0 & I_{zn} \end{bmatrix} \tag{A6}$$

The generalized forces vector is

$$\begin{aligned} Q_{nc,i} &= \sum_{J=1}^{6} \vec{\tau}_{nc,J} \frac{\partial \vec{\omega}_J}{\partial \dot{q}_i} \qquad\qquad i = 1, 2, \ldots, 24 \\ \mathbf{Q}_{nc} &= [0, 0, 0, 0, 0, 0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^{\mathrm{T}} \end{aligned} \tag{A7}$$

By substituting Eqs. (A3) and (A7) in Eq. (A1), the governing equations are obtained.