

Closed-form dynamics of a 3-DOF spatial parallel manipulator by combining the Lagrangian formulation with the virtual work principle

Guiyang Xin · Hua Deng · Guoliang Zhong

Received: 26 August 2015 / Accepted: 16 July 2016 / Published online: 28 July 2016
© Springer Science+Business Media Dordrecht 2016

Abstract Dynamic modeling of spatial multi-degree-of-freedom (multi-DOF) parallel motion system is a challenging research because of its multi-closed-loop configuration, in contrast to serial manipulator. This study readdresses the issue of deriving explicit closed-form dynamic equations in the actuation space of non-redundant parallel mechanisms through the Lagrange modeling approach. Due to the complex relationship between passive joint variables and active joint variables, the application of the Lagrangian formulation for dynamic modeling of parallel mechanisms is considered nearly impossible. In this paper, the utilization of the virtual work principle makes the application of the Lagrangian formulation for parallel mechanisms possible and efficient. A closed-loop parallel mechanism is divided into several serial open-loop subchains. Explicit dynamic equations of each subchain can be derived by using the Lagrangian formalism straightforwardly with respect to their own local generalized coordinates. The principle of virtual work is used to combine the differential dynamic equations of the subsystems into an explicit dynamic model of the full parallel manipulator with respect to robot's active gener-

alized coordinates. The Jacobian and Hessian matrices play a crucial role in the transformation of the dynamic equations with respect to different generalized coordinates due to the application of the virtual work principle. The introduction of the cubic Hessian matrices makes the dynamic equations more compact. The procedure of modeling for a 3-DOF spatial parallel manipulator is taken as an application example of the proposed approach. The efficiency of the proposed approach and the correctness of the dynamic model are demonstrated by simulations and experiments.

Keywords Dynamics · Lagrangian formulation · Parallel mechanism · Virtual work · Hessian matrix

1 Introduction

It is well known that parallel mechanisms have significant advantages over serial mechanisms in terms of accuracy, rigidity and payload ability [1–4]. Compared with serial open-loop mechanisms, the kinematics of parallel mechanisms is more complex due to their multi-closed-loop structures. The complexity of kinematics leads to quite complicated dynamic model because dynamics is a natural extension of kinematics [5,6]. Therefore, the derivation of explicit dynamic equations in the actuation space for parallel mechanisms is often difficult except several particular manipulators. However, the precise dynamic equations in the actuation space play an important role in design

G. Xin · H. Deng · G. Zhong (✉)
School of Mechanical and Electrical Engineering,
Central South University, Changsha 410083, Hunan, China
e-mail: zhong001985@csu.edu.cn

G. Xin · H. Deng · G. Zhong
State Key Laboratory of High-Performance Complex
Manufacturing, Central South University,
Changsha 410083, Hunan, China

and operation of robotic devices. In design, dynamic equations are employed as the dynamic constraints for an optimal design process or as a tool to test performance indexes of a robot. As regards robot operations, dynamic equations are the most important part to develop control laws and determine controller's parameters; inverse dynamics are always used to compute the actuation forces as feedforward information in order to achieve desired motions, which are called the internal model control or model-based control. The trajectory tracking performance of most commercial parallel robots is limited because they are controlled by simple single-joint feedback controller ignoring the coupled dynamics. Unlike the serial mechanisms that have well-established modeling and control methods, most studies of the parallel mechanisms focus on kinematics and optimal design issues. To the best of our knowledge, the dynamic modeling of parallel mechanisms is still an open problem. Therefore, we aim for an efficient method for the derivation of the control-oriented dynamic model, i.e., the explicit closed-form dynamic equations in the actuation space.

The literature on the dynamic modeling is rather vast, and it will be impossible to mention all of them. For parallel mechanisms, there are mainly four kinds of methods to build the dynamic equations for them. The first one is the Newton–Euler method. Dasgupta and Mruthyunjaya [7] obtained the dynamic equations of two types of the Stewart platform via Newton–Euler method. Using this approach, the acceleration of every isolated rigid body should be obtained. All constraint forces and moments between the joints also should be obtained because of the application of the Newton–Euler equations for each rigid body, whereas the internal forces and moments of interaction between elements of a robot are useless for the control of a manipulator. The second method that is widely used is Kane's method. Kane proposed Kane's dynamical equations for robotics applications to formulate explicit equations of motion [8,9]. This approach uses Lagrange's form of d'Alembert's principle, as expounded by Houston and Passerello in [10–13]. Lagrange's form of d'Alembert's principle states that the sum of the total generalized active force and the total generalized inertia force, for each generalized coordinate of the system, is zero, which are Kane's dynamical equations. Kane's dynamical equations have the advantage of automatically eliminating “no-working” internal constraint forces. In order to efficiently obtain the gen-

eralized forces needed for Kane's equations, the concepts of partial angular velocity and partial velocity are introduced by Kane [8,9]. Subsequently, Houston and Amirouche made great contribution to the development of Kane's method [14–23]. As for dynamic modeling of parallel mechanisms, they employed the combination of Kane's dynamic equations and constrained kinematic equations or undetermined multipliers embedded in Kane's equations. Then, in order to derive the explicit dynamic equations with respect to the independent coordinates, the zero eigenvalue decomposition and pseudo-uptriangular decomposition methods are proposed by Houston and Amirouche, respectively, to eliminate the undetermined multipliers. Their method is an efficient method especially when the dimension of the constraint equations is large due to a large number of rigid bodies of the multi-body system. However, the derivation of partial velocities and accelerations are inevitable, which needs a bit of tedious calculations. Lih [24] studied Kane's method employed by Wang and Huston [15] and indicated that they didn't generate the closed-form dynamic equations. Then, he successfully used the principle of virtual work to get the closed-form dynamic model for closed-loop systems [24,25]. Currently, the principle of virtual work has been a very popular method to derive dynamic models of parallel mechanisms [26–29]. The principle of virtual work states that the work done by a dynamically equilibrium system undergoing arbitrarily displacement vanishes. It is salient that the “no-working” forces can be dropped out from the virtual work equation, which is very similar with Kane's method. The calculation of every body's acceleration is also demanded in this approach. Different to the Newton–Euler approach, the characteristic of the principle of virtual work for a system equaling zero is applied to combine all dynamic equations of isolated bodies into one dynamic equation of the total manipulator. However, Guo and Li [30] think the method based on principle of virtual work delivers only an implicit model of inverse dynamics. The last one is Lagrangian formulation. Traditionally, Lagrange's method considers the system as a whole instead of isolating each body as the other methods mentioned above. Lagrange's method is very successful for serial mechanisms since the Lagrangian function is simply expressed by the actuator variables for serial mechanisms, whereas quite a few of literature have used the Lagrangian formulation for parallel mechanisms due to the constraints introduced by the closed

loops. In [31–34], because of the complexity of the kinematics, dynamic model simplification was suggested, like neglecting Coriolis terms or inertia tensors. In [35], the passive joint variables are involved in generalized coordinates, which increases the dimension of dynamic equations. So, the result in [35] is not the expected explicit form of dynamics. Additionally, most of the previous works just consider the primary rigid body dynamics of a manipulator while neglect the friction even the actuator dynamics. However, for an actual physical prototype, the actuator dynamics and friction frequently play a large part in the governing equations of motion. It was demonstrated in [36] that for some systems, friction compensation yields significant improvement of control performance. Obviously, a precise dynamic model without neglecting any parts of the dynamic equations would benefit the perception of the system and controller design. Below we contribute to remove all these drawbacks. The novelty of this paper is the combination of Lagrangian formulation and the virtual work principle for being free of the complicated computation of acceleration and internal forces and moments. This is an important advantage of the modeling approach over the Newton–Euler formulation.

In this paper, the Lagrangian formulation is restudied in order to determine the explicit closed-form dynamic equations in the actuation space of parallel mechanisms by combining the Lagrangian formulation with the principle of virtual work. The closed-loop structure of the manipulator is opened into several serial subchains in order to form explicit energy functions of subsystems with respect to their local coordinates for the convenience of the subsequent differentiating calculation. The dynamics of subsystems corresponding to different local generalized coordinates can be easily obtained through Lagrange's equations of the second type because of the conveniences of the Lagrange formulation for serial robots. However, the dynamic equations of subsystems should be grouped into the closed-form dynamic equations of the manipulator corresponding to the active generalized coordinates. The virtual work principle could transform the generalized joint torques into another joint space according to the relationship between different generalized coordinates. The Jacobian and Hessian matrices, which represent the key of robot's kinematics, play the transformation bridge roles in combining different dynamic equations of isolated body subsystems. The cube Hessian

matrices are only utilized in the study of kinematics in [37,38]. Here, we introduce them into dynamic equations, which make the equations more concise. Additionally, the inclusion of friction in the derived rigid body dynamic equations is also discussed.

The goal of this paper is to propose an alternative dynamic modeling method for parallel robots. This method is embedded with four desired merits. (1) *Accuracy* the use of virtual work principle to deal with kinematic constraints makes the Lagrangian formulation available for parallel robots without neglecting any parts of dynamic equations; hence, the derived dynamic model is an accurate rigid body dynamic model without any simplification. (2) *Convenient implementation* there are only five steps to obtain the final dynamic model for any non-redundant parallel robots and the process is easy to be implemented by computer algebra program. (3) *Explicit expression* the derived dynamic equations are explicit with respect to active joints, which is important for the solution of forward dynamics. (4) *High efficiency* the derivation process for the explicit closed-form dynamic equations is high efficiency, and the computational cost for solution of inverse dynamics is low enough for online model-based control.

The paper is organized as follows. The problems of using the Lagrangian formulation to derive dynamic equations for parallel mechanisms are discussed in detail in Sect. 2. The proposed approach is then developed in Sect. 3. In Sect. 4, as an application example, the inverse kinematics of a 3-DOF parallel manipulator is analyzed. Then, the dynamic equations of motion are formulated through the proposed approach. In Sect. 5, the dynamic equations are verified by the commercial software ADAMS and experiments on a test bed. Some conclusions and future research directions are given in Sect. 6. Although the 3-DOF manipulator is used as an example to illustrate the modeling approach, the methodology can be used for other types of parallel manipulators.

2 Problem formulation

When neglecting the friction and other disturbances, for non-redundant N -DOF mechanisms, the explicit closed-form dynamic equations is a N -dimensional set of differential equations as

$$\mathbf{D}\ddot{\mathbf{q}}_a + \mathbf{C}\dot{\mathbf{q}}_a + \mathbf{G} = \boldsymbol{\tau}_a, \quad (1)$$

where $\mathbf{q}_a \in \mathbf{R}^N$ is the vector of the active generalized coordinates that is the coordinates of the actuators. $\boldsymbol{\tau}_a$ denotes the vector of the corresponding active forces. The inertia matrix, the Coriolis and centrifugal terms, and the gravitational forces are denoted by \mathbf{D} , \mathbf{C} and \mathbf{G} , respectively. Eq. (1) is given in the actuation space responding to \mathbf{q}_a , which is very useful for internal model control, dynamic simulation and real-time application. Eq. (1) can be directly derived by Lagrange's equations of the second type

$$\frac{d}{dt} \frac{\partial E_k(\mathbf{q}_a)}{\partial \dot{\mathbf{q}}_a} - \frac{\partial E_k(\mathbf{q}_a)}{\partial \mathbf{q}_a} + \frac{\partial E_p(\mathbf{q}_a)}{\partial \mathbf{q}_a} = \boldsymbol{\tau}_a, \quad (2)$$

where $E_k(\mathbf{q}_a)$ and $E_p(\mathbf{q}_a)$ denote the kinetic energy and potential energy of the whole system, respectively. $E_k(\mathbf{q}_a)$ and $E_p(\mathbf{q}_a)$ are expressed by \mathbf{q}_a that are the vector of active variables responding to time. For serial mechanisms, all coordinates are active joint variables. So the kinetic energy and potential energy are naturally expressed in terms of \mathbf{q}_a . For parallel mechanisms, the active joints are commonly prismatic joints, and the kinetic energy and potential energy are conveniently expressed by a set of general coordinates \mathbf{q} composed of both active and passive joint variables, rather than only expressed by active joint variables. Therefore, the kinetic energy and potential energy of parallel mechanisms denoted by $E_k(\mathbf{q})$ and $E_p(\mathbf{q})$, respectively, are in function of \mathbf{q} and time t . Hence, the partial derivatives with respect to \mathbf{q}_a , i.e., $\partial E_k(\mathbf{q})/\partial \mathbf{q}_a$ and $\partial E_p(\mathbf{q})/\partial \mathbf{q}_a$, are very hard to be computed. This is the special problem of parallel mechanisms in contrast to serial mechanisms. The most common way to solve this problem is to obtain the relationship between \mathbf{q}_a and \mathbf{q} by the geometrical constraints of the parallel mechanism, i.e.,

$$f(\mathbf{q}_a, \mathbf{q}) = 0. \quad (3)$$

In the best case, Eq. (3) could be transformed into an explicit form where every general coordinate could be determined by functions of the active general coordinates. Unfortunately, Eq. (3) always cannot be transformed into an explicit form, which makes it impossible to determine of $E_k(\mathbf{q}_a)$ and $E_p(\mathbf{q}_a)$ explicitly. As for traditional Lagrangian formalism, one must perform unnecessary labor in introducing and subsequently eliminating Lagrange multipliers. That is the

drawback of the usual way to apply the traditional Lagrangian formalism for dynamic modeling of parallel mechanisms. When using Kane's equations to obtain the dynamic model, as mentioned before, the pseudo-uppertriangular decomposition [18], the zero eigenvalue theorem [14], and the singular value decomposition method [39] are employed to eliminate the undetermined multipliers in Kane's equations. However, we are trying to make the Lagrangian formalism for parallel mechanisms to overcome this obstacle. In the following sections, we will show how the principle of virtual work provides an efficient approach to solve the abovementioned problems.

3 Lagrangian formalism for parallel mechanisms

Assume that the manipulator can be divided into several isolated rigid bodies. The vector of the generalized coordinates corresponding to i th ($i = 1, \dots, m$) isolated rigid body denotes \mathbf{q}_i . The Jacobian matrix between \mathbf{q}_i and \mathbf{q} can be obtained by kinematic analysis. So the velocity mapping relationship has the following equation,

$$\dot{\mathbf{q}}_i = \mathbf{J}_i \dot{\mathbf{q}}. \quad (4)$$

where $\mathbf{q}_i \in \mathbf{R}^{n_i}$, $\mathbf{q} \in \mathbf{R}^N$, $\mathbf{J}_i \in \mathbf{R}^{n_i \times N}$. The generalized forces corresponding to \mathbf{q}_i are defined as \mathbf{f}_i . According to the virtual work principle, the following forces transformation exists,

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \mathbf{f}_i, \quad (5)$$

where $\boldsymbol{\tau}_i$ indicates the mapping from \mathbf{f}_i to another generalized coordinates, i.e., \mathbf{q} . Eq. (5) for forces transformation gives us a new idea to formulate the dynamics of multi-body systems.

Firstly, we use the Lagrangian formulation to derive the dynamics of isolated bodies with respect to their local generalized coordinates, as follows,

$$\mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + \mathbf{G}_i(\mathbf{q}_i) = \mathbf{f}_i, \quad (6)$$

where \mathbf{f}_i is the vector of generalized forces. Left multiplying Eq. (6) by \mathbf{J}_i^T , the following equation is obtained,

$$\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{J}_i^T \mathbf{C}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \boldsymbol{\tau}_i. \quad (7)$$

Eq. (7) reveals the contribution to the generalized forces τ_i of the motion of the i th isolated rigid body. Here, the Coriolis and centrifugal terms $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i$ can be rewrite as follows,

$$C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)\dot{\mathbf{q}}_i, \tag{8}$$

where $\mathbf{H}_{Ci}(\mathbf{q}_i) \in \mathbf{R}^{n_i \times n_i \times n_i}$ is the cubic Hessian matrix; the sign “*” denotes a generalized scalar product [37,38]. And n_i denotes the dimension of \mathbf{q}_i . The generalized scalar product of two matrices $\mathbf{X} \in \mathbf{R}^{m \times n}$ and $\mathbf{Y} \in \mathbf{R}^{n \times p \times p}$ are defined as follows,

$$[\mathbf{XY}]_{k::} = \sum_{l=1}^n \mathbf{X}_{k:l} \mathbf{Y}_{l::} \in \mathbf{R}^{p \times p}, \quad k = 1, 2, \dots, m \tag{9}$$

where $\mathbf{XY} \in \mathbf{R}^{m \times p \times p}$. Equation (9) means that only when the column number of \mathbf{X} is equal to the layer number of \mathbf{Y} , the two matrices \mathbf{X} and \mathbf{Y} can be multiplied using the generalized scalar product.

Afterward, substituting Eq. (8) to (7), yields,

$$\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \dot{\mathbf{q}}_i^T * [\mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)]\dot{\mathbf{q}}_i + \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \tau_i. \tag{10}$$

where $\mathbf{J}_i^T \in \mathbf{R}^{N \times n_i}$. Therefore, $\mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i) \in \mathbf{R}^{N \times n_i \times n_i}$ according to Eq. (9). However, $\mathbf{q}_i^T \in \mathbf{R}^{1 \times n_i}$. This implies that only when $n_i = N$ is satisfied, Eq. (10) can be used. Otherwise, please don't use Hessian matrix and keep using $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$. Hereinafter, assume that n_i is equal to N . The application of Hessian matrix separates $\dot{\mathbf{q}}_i$ from $C_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$, which brings convenience to transform Eq. (10) to standard form of Eq. (1).

Next, differentiating Eq. (4) with respect to time, yields,

$$\ddot{\mathbf{q}}_i = \mathbf{J}_i \ddot{\mathbf{q}} + \dot{\mathbf{J}}_i \dot{\mathbf{q}} = \mathbf{J}_i \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}_i \dot{\mathbf{q}}. \tag{11}$$

Similarly, if n_i is not equal to N , please keep using $\dot{\mathbf{J}}_i$ and don't use \mathbf{H}_i .

Substituting Eqs. (4) and (11) to Eq. (10), the following equation is obtained,

$$\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\mathbf{J}_i \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * [\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i) * \mathbf{H}_i + \mathbf{J}_i^T \mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)\mathbf{J}_i]\dot{\mathbf{q}} + \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \tau_i. \tag{12}$$

The dynamic equations with respect to the generalized coordinates \mathbf{q} of the full manipulator can be derived by summing Eq. (12) with respect to each isolated rigid body, as follows,

$$\left[\sum_{i=1}^m \mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\mathbf{J}_i \right] \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \left[\sum_{i=1}^m (\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i) * \mathbf{H}_i + \mathbf{J}_i^T \mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)\mathbf{J}_i) \right] \dot{\mathbf{q}} + \sum_{i=1}^m \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \sum_{i=1}^m \tau_i = \tau. \tag{13}$$

Let \mathbf{q} be the vector of the active generalized coordinates. Thus, Eq. (13) takes the form of Eq. (1), as follows,

$$\left[\sum_{i=1}^m \mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\mathbf{J}_i \right] \ddot{\mathbf{q}}_a + \dot{\mathbf{q}}_a^T * \left[\sum_{i=1}^m (\mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i) * \mathbf{H}_i + \mathbf{J}_i^T \mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)\mathbf{J}_i) \right] \dot{\mathbf{q}}_a + \sum_{i=1}^m \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \tau_a. \tag{14}$$

If \mathbf{q} is not the vector of the active generalized coordinates, it is possible to find the relationships. Introductions,

$$\dot{\mathbf{q}} = \mathbf{J} \dot{\mathbf{q}}_a, \tag{15}$$

$$\tau_a = \mathbf{J}^T \tau, \tag{16}$$

$$\ddot{\mathbf{q}} = \mathbf{J} \ddot{\mathbf{q}}_a + \dot{\mathbf{q}}_a^T * \mathbf{H} \dot{\mathbf{q}}_a. \tag{17}$$

Introducing Eqs. (15)–(17) into Eq. (13) yields,

$$\left[\sum_{i=1}^m \mathbf{J}^T \mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i)\mathbf{J}_i \mathbf{J} \right] \ddot{\mathbf{q}}_a + \dot{\mathbf{q}}_a^T * \left[\sum_{i=1}^m \mathbf{J}^T \mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i) * \mathbf{H} \mathbf{J}_i + \sum_{i=1}^m (\mathbf{J}^T \mathbf{J}^T \mathbf{J}_i^T \mathbf{M}_i(\mathbf{q}_i) * \mathbf{H}_i \mathbf{J} + \mathbf{J}^T \mathbf{J}^T \mathbf{J}_i^T \mathbf{J}_i^T * \mathbf{H}_{Ci}(\mathbf{q}_i)\mathbf{J}_i \mathbf{J}) \right] \dot{\mathbf{q}}_a + \sum_{i=1}^m \mathbf{J}^T \mathbf{J}_i^T \mathbf{G}_i(\mathbf{q}_i) = \tau_a. \tag{18}$$

Equation (18) is the explicit closed-form dynamic equations in the actuation space of a general parallel

mechanism. The abovementioned algorithm giving the dynamic model of parallel mechanisms can be summarized to the following five steps.

- Step 1. A multi-closed-chain manipulator can be divided into several open-chain isolated rigid body by cutting selected joints.
- Step 2. Derive the dynamic equations of the isolated subchains with respect to the local generalized coordinates \mathbf{q}_i by the direct Lagrangian formulation.
- Step 3. Transform dynamic equations of each sub-chain into equations with respect to the same generalized coordinates \mathbf{q} (maybe the active generalized coordinates \mathbf{q}_a) through the virtual work principle.
- Step 4. Use Jacobian and Hessian matrices to transform local generalized coordinates to the same generalized coordinates.
- Step 5. Sum all the equations of subsystems to obtain the closed-form dynamic equations of the full manipulator. If $\mathbf{q} \neq \mathbf{q}_a$, transform the full dynamic equations with respect to \mathbf{q} to equations with respect to \mathbf{q}_a by the approach mentioned in steps 3 and 4.

4 Dynamics of a 3-DOF spatial parallel manipulator

There have been a great number of practical applications of parallel mechanisms, such as the aircraft simulator [40], the force/torque sensor [41], and the acceleration sensor [42]. In recent years, the limited-DOF manipulators that both maintain the inherent advantages of parallel mechanisms and possess several other advantages in terms of the total cost reduction in manufacturing [43] and wider workspace are attracting more attentions of researchers, for instance the Delta robot [44], and Tricept robot and TriVariant robot [45]. In this paper, the 3-DOF spatial parallel manipulator presented in reference [46], which contains 1-UP chain and 2-UPS chains, is used as an example to verify the effectiveness of the proposed modeling approach. U, P, and S denote universal, prismatic, and spherical joint respectively. P pair is the actuated joint driven by an actuator. This 3-DOF parallel manipulator can be used as cutting robot or grabbing robot after assembling corresponding terminal effectors. The CAD model of the 3-DOF manipulator is represented in Fig. 1.

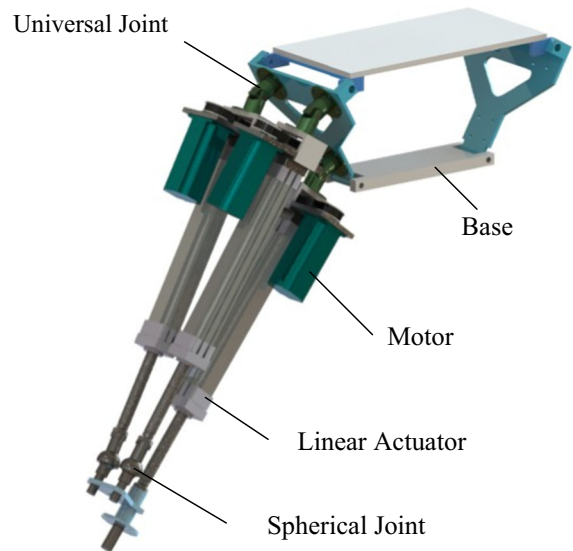


Fig. 1 Spatial 3-DOF parallel manipulator with three linear actuators

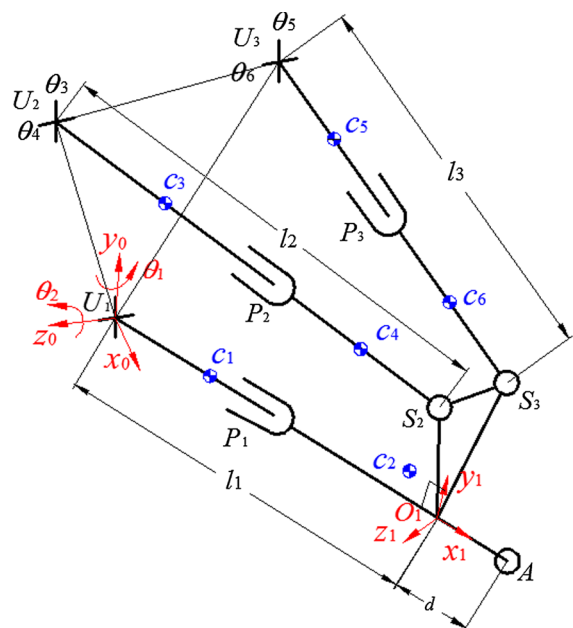


Fig. 2 Schematic diagram of the 1-UP&2-UPS manipulator

4.1 Kinematics

Figure 2 shows the schematic diagram of the 3-DOF manipulator, which can be decomposed into six isolated rigid bodies. The six isolated rigid bodies are called Link 1 to 6, respectively, and their corresponding center of mass is defined as c_1, \dots, c_6 as shown

in Fig. 2. $U_i (i = 1, 2, 3)$ represents the center of the U joint. $S_i (i = 2, 3)$ represents the center of the S joint. A denotes the end point of the UP limb. O_1 is the intersection of the axis of the UP limb and the normal plane in which $S_i (i = 2, 3)$ are located. $\Delta U_1 U_2 U_3$ and $\Delta O_1 S_2 S_3$ are isosceles triangles. $\theta_i (i = 1, \dots, 6)$ denotes the rotational angle of U joint. Frame $U_1 - x_0 y_0 z_0$ is fixed on the base, while y_0 axis is coincident with the first rotational axis of U_1 , i.e., the rotational axis of θ_1 ; z_0 axis is parallel to the second rotational axis, i.e., the rotational axis of θ_2 ; the x_0 axis satisfies the right-hand rule. The axial axis of the UP limb is coincident with x_1 . y_1 lies in the plane of $\Delta O_1 S_2 S_3$ and is normal to the $S_2 S_3$. $\mathbf{l}_1 \equiv \overrightarrow{U_1 O_1}$, $\mathbf{l}_2 \equiv \overrightarrow{U_2 S_2}$ and $\mathbf{l}_3 \equiv \overrightarrow{U_3 S_3}$ are defined as components in the base frame $U_1 - x_0 y_0 z_0$. Define the following vectors of generalized coordinates,

$$\mathbf{q}_a = [l_1 \quad l_2 \quad l_3]^T, \quad \mathbf{q} = [\theta_1 \quad \theta_2 \quad l_1]^T, \\ \mathbf{q}_1 = [\theta_3 \quad \theta_4 \quad l_2]^T, \quad \mathbf{q}_2 = [\theta_5 \quad \theta_6 \quad l_3]^T.$$

\mathbf{q} , \mathbf{q}_1 , \mathbf{q}_2 are the local generalized coordinates for three open-chains respectively, i.e., the UP chain and two UPS chains. According to the proposed approach in Sect. 3, the dynamics of every isolated part can be solved by the direct Lagrangian formulation with respect to the local generalized coordinates.

Firstly, as usual, the relationship between the position of point A at the tip and the position of actuators should be derived to implement trajectory tracking. We can obtain the coordinate transformation matrix between frames $U_1 - x_0 y_0 z_0$ and $O_1 - x_1 y_1 z_1$, as follows,

$${}^0_1\mathbf{T} = Rot(y_0, \theta_1)Rot(z_0, \theta_2)Trans(x_0, l_1) \\ = \begin{bmatrix} c_1 c_2 & -c_1 s_2 & s_1 & l_1 c_1 c_2 \\ s_2 & c_2 & 0 & l_1 s_2 \\ -s_1 c_2 & s_1 s_2 & c_1 & -l_1 s_1 c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (19)$$

where l_1 denotes $|\mathbf{l}_1|$; s_i and $c_i (i=1,2)$ denote $\sin \theta_i$ and $\cos \theta_i$, respectively. Then, the position vector of the tip A in frame $U_1 - x_0 y_0 z_0$ can be derived,

$${}^0\mathbf{p}_A = \begin{bmatrix} {}^0x_A \\ {}^0y_A \\ {}^0z_A \end{bmatrix} = \begin{bmatrix} (l_1 + d)c_1 c_2 \\ (l_1 + d)s_2 \\ -(l_1 + d)s_1 c_2 \end{bmatrix}, \quad (20)$$

where d denotes the length between the tip A and point O_1 . According to Eq. (20), the following equation is obtained,

$$\begin{cases} l_1 = \sqrt{{}^0x_A^2 + {}^0y_A^2 + {}^0z_A^2} - d, \\ \theta_2 = \arcsin\left(\frac{{}^0y_A}{l_1 + d}\right), \\ \theta_1 = \arcsin\left(\frac{-{}^0z_A}{(l_1 + d)c_2}\right). \end{cases} \quad (21)$$

According to geometrical relationship, we can obtain,

$$\begin{cases} \begin{bmatrix} \mathbf{l}_2 \\ 0 \end{bmatrix} = {}^0_1\mathbf{T} \begin{bmatrix} {}^1\mathbf{p}_{S_2} \\ 1 \end{bmatrix} - \begin{bmatrix} {}^0\mathbf{p}_{U_2} \\ 1 \end{bmatrix}, \\ \begin{bmatrix} \mathbf{l}_3 \\ 0 \end{bmatrix} = {}^0_1\mathbf{T} \begin{bmatrix} {}^1\mathbf{p}_{S_3} \\ 1 \end{bmatrix} - \begin{bmatrix} {}^0\mathbf{p}_{U_3} \\ 1 \end{bmatrix}. \end{cases} \quad (22)$$

where ${}^0\mathbf{p}_{U_2} = [0 \ a_y \ a_z]^T$, ${}^0\mathbf{p}_{U_3} = [0 \ a_y \ -a_z]^T$, ${}^1\mathbf{p}_{S_2} = [0 \ b_y \ b_z]^T$, ${}^1\mathbf{p}_{S_3} = [0 \ b_y \ -b_z]^T$. Substituting Eq. (19) into Eq. (22), leads to

$$\begin{cases} \mathbf{l}_2 = \begin{bmatrix} -b_y c_1 s_2 + b_z s_1 + l_1 c_1 c_2 \\ b_y c_2 + l_1 s_2 - a_y \\ b_y s_1 s_2 + b_z c_1 - l_1 s_1 c_2 - a_z \end{bmatrix}, \\ \mathbf{l}_3 = \begin{bmatrix} -b_y c_1 s_2 - b_z s_1 + l_1 c_1 c_2 \\ b_y c_2 + l_1 s_2 - a_y \\ b_y s_1 s_2 - b_z c_1 - l_1 s_1 c_2 + a_z \end{bmatrix}. \end{cases} \quad (23)$$

Thus, the displacements of l_2 and l_3 can be calculated as

$$l_2 = \sqrt{\mathbf{l}_2^T \mathbf{l}_2} \\ = \sqrt{a_y^2 + a_z^2 + b_y^2 + b_z^2 + l_1^2 - 2a_y b_y c_2 - 2a_y l_1 s_2 - 2a_z b_z c_1 - 2a_z b_y s_1 s_2 + 2a_z l_1 s_1 c_2}, \quad (24)$$

$$l_3 = \sqrt{\mathbf{l}_3^T \mathbf{l}_3} \\ = \sqrt{a_y^2 + a_z^2 + b_y^2 + b_z^2 + l_1^2 - 2a_y b_y c_2 - 2a_y l_1 s_2 - 2a_z b_z c_1 + 2a_z b_y s_1 s_2 - 2a_z l_1 s_1 c_2}. \quad (25)$$

The inverse kinematics is given by Eqs. (21), (24) and (25). The velocity and acceleration equations can be derived by differentiating Eqs. (20), (24) and (25) as follows,

$${}^0\dot{\mathbf{p}}_A = \mathbf{J}_A \dot{\mathbf{q}}$$

$$= \begin{bmatrix} -(l_1 + d)s_1c_2 & -(l_1 + d)c_1s_2 & c_1c_2 \\ 0 & (l_1 + d)c_2 & s_2 \\ -(l_1 + d)c_1c_2 & (l_1 + d)s_1s_2 & -s_1c_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{l}_1 \end{bmatrix}, \tag{26}$$

$$\dot{\mathbf{q}}_a = \mathbf{J}_a \dot{\mathbf{q}}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ \frac{a_zb_zs_1 - a_zb_ys_1s_2 + a_zl_1c_1c_2}{l_2} & \frac{a_yb_ys_2 - a_y l_1c_2 - a_zb_ys_1c_2 - a_zl_1s_1s_2}{l_2} & \frac{l_1 - a_ys_2 + a_zs_1c_2}{l_2} \\ \frac{a_zb_zs_1 + a_zb_yc_1s_2 - a_zl_1c_1c_2}{l_3} & \frac{a_yb_ys_2 - a_y l_1c_2 + a_zb_ys_1c_2 + a_zl_1s_1s_2}{l_3} & \frac{l_1 - a_ys_2 - a_zs_1c_2}{l_3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{l}_1 \end{bmatrix}. \tag{27}$$

Then, we can obtain the following equation,

$${}^0\ddot{\mathbf{P}}_A = \mathbf{J}_{leg} \ddot{\mathbf{q}}_a = \mathbf{J}_A \mathbf{J}_a^{-1} \ddot{\mathbf{q}}_a, \tag{28}$$

where \mathbf{J}_{leg} is the overall velocity Jacobian matrix of the manipulator. Then, the acceleration mapping relationship can be obtained by differentiating Eq. (28), as follows,

$${}^0\ddot{\mathbf{P}}_A = \mathbf{J}_{leg} \ddot{\mathbf{q}}_a + \dot{\mathbf{q}}_a^T * \mathbf{H}_{leg} \dot{\mathbf{q}}_a. \tag{29}$$

Next, we derive the relationship between \mathbf{q} and \mathbf{q}_i ($i = 1, 2$). \mathbf{q}_1 can be determined making $U_1 O_1 S_2 U_2$ a closed-chain, as follows,

$${}^0\mathbf{p}_{S_2}$$

$$= \begin{bmatrix} l_2c_3c_4 \\ l_2s_4 \\ -l_2s_3c_4 \end{bmatrix} = \begin{bmatrix} -b_yc_1s_2 + b_zs_1 + l_1c_1c_2 \\ b_yc_2 + l_1s_2 - a_y \\ b_ys_1s_2 + b_zc_1 - l_1s_1c_2 - a_z \end{bmatrix}. \tag{30}$$

According to Eq. (30) we can obtain

$$\theta_3 = \arctan \frac{b_ys_1s_2 + b_zc_1 - l_1s_1c_2 - a_z}{b_yc_1s_2 - b_zs_1 - l_1c_1c_2}, \tag{31}$$

$$\theta_4 = \arcsin \frac{b_yc_2 + l_1s_2 - a_y}{l_2}. \tag{32}$$

Similarly, \mathbf{q}_2 can be solved through the closed-chain $U_1 O_1 S_3 U_3$, as follows,

$$\theta_5 = \arctan \frac{b_ys_1s_2 - b_zc_1 - l_1s_1c_2 + a_z}{b_yc_1s_2 + b_zs_1 - l_1c_1c_2}, \tag{33}$$

$$\theta_6 = \arcsin \frac{b_yc_2 + l_1s_2 - a_y}{l_3}. \tag{34}$$

Secondly, the Jacobian and Hessian matrices should be obtained for dynamic modeling. The Jacobian

matrix \mathbf{J}_a for \mathbf{q}_a and \mathbf{q} has been given in Eq. (27). The Hessian matrix \mathbf{H}_a can be obtained by differentiation of Eq. (26). It should be noted that all of the calculations can be implemented using a symbolic algebra package as MAPLE, which can help us to simplify the calculations, and to calculate \mathbf{J}_i and \mathbf{H}_i for \mathbf{q} and \mathbf{q}_i ($i = 1, 2$) by differentiating Eqs. (24), (25) and (31)–(34).

4.2 Dynamic modeling using Lagrangian formulation

As shown in Fig. 2, according to the proposed approach, we need to obtain the dynamic equations for each link using the Lagrangian formulation. Here, we take the computation of dynamic equations for Link 1 as an example. The kinetic energy and potential energy of Link 1 with respect to \mathbf{q} are as follows,

$$E_{k1}(\mathbf{q}) = \frac{1}{2} (m_1 {}^0\dot{\mathbf{p}}_{c1}^T(\mathbf{q}) {}^0\dot{\mathbf{p}}_{c1}(\mathbf{q}) + \boldsymbol{\omega}_1^T \mathbf{I}_{c1} \boldsymbol{\omega}_1), \tag{35}$$

$$E_{p1}(\mathbf{q}) = m_1 {}^0\mathbf{g}^T \mathbf{P}_{c1}(\mathbf{q}), \tag{36}$$

where

$${}^0\mathbf{P}_{c1} = \begin{bmatrix} x_{c1}c_1c_2 - y_{c1}c_1s_2 + z_{c1}s_1 \\ x_{c1}s_2 + y_{c1}c_2 \\ -x_{c1}s_1c_2 + y_{c1}s_1s_2 + z_{c1}c_1 \end{bmatrix},$$

$${}^0\mathbf{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix},$$

$$\mathbf{I}_{c1} = \begin{bmatrix} L_{1xx} & L_{1xy} & L_{1xz} \\ L_{1xy} & L_{1yy} & L_{1yz} \\ L_{1xz} & L_{1yz} & L_{1zz} \end{bmatrix},$$

x_{c1} , y_{c1} , and z_{c1} are constant variables, m_1 denotes the mass of Link 1, \mathbf{I}_{c1} means the moment of inertia of Link 1. Thus, introducing Eqs. (35) and (36) into Lagrange's equations of the second type, i.e., Eq. (2), yields,

$$\mathbf{M}_1(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}_{c1}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{G}_1(\mathbf{q}) = \boldsymbol{\tau}_1. \tag{37}$$

Similarly, the dynamic equations of Link k ($k = 2, \dots, 6$) can be derived using the identical approach, as follows,

Link 2:

$$\mathbf{M}_2(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}_{c2}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{G}_2(\mathbf{q}) = \boldsymbol{\tau}_2. \tag{38}$$

As an example, Appendix 1 gives the expressions of Eq. (38) in detail.

Link 3–4:

$$\begin{aligned} \mathbf{M}_k(\mathbf{q}_1)\ddot{\mathbf{q}}_1 + \dot{\mathbf{q}}_1^T * \mathbf{H}_{ck}(\mathbf{q}_1)\dot{\mathbf{q}}_1 + \mathbf{G}_k(\mathbf{q}_1) \\ = \mathbf{f}_k. (k = 3, 4) \end{aligned} \tag{39}$$

Link 5–6:

$$\begin{aligned} \mathbf{M}_k(\mathbf{q}_2)\ddot{\mathbf{q}}_2 + \dot{\mathbf{q}}_2^T * \mathbf{H}_{ck}(\mathbf{q}_2)\dot{\mathbf{q}}_2 + \mathbf{G}_k(\mathbf{q}_2) \\ = \mathbf{f}_k. (k = 5, 6) \end{aligned} \tag{40}$$

Then, the kinematic mapping between the local generalized coordinates \mathbf{q}_i and \mathbf{q} has been solved in Sect. 4.1, as follows,

$$\dot{\mathbf{q}}_i = \mathbf{J}_i(\mathbf{q}, \mathbf{q}_i)\dot{\mathbf{q}}. (i = 1, 2) \tag{41}$$

$$\ddot{\mathbf{q}}_i = \mathbf{J}_i(\mathbf{q}, \mathbf{q}_i)\ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}_i(\mathbf{q})\dot{\mathbf{q}}. (i = 1, 2) \tag{42}$$

According to the virtual work principle, that implies

$$\boldsymbol{\tau}_k = \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{f}_k, (k = 3, 4) \tag{43}$$

$$\boldsymbol{\tau}_k = \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{f}_k. (k = 5, 6) \tag{44}$$

Combining Eqs. (41)–(44) and (39), (40) leads to

$$\begin{aligned} \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{M}_k(\mathbf{q}_1)\mathbf{J}_1(\mathbf{q}, \mathbf{q}_1)\ddot{\mathbf{q}} \\ + \dot{\mathbf{q}}^T * [\mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{M}_k(\mathbf{q}_1) * \mathbf{H}_1(\mathbf{q}) \\ + \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1) * \mathbf{H}_{ck}(\mathbf{q}_1)\mathbf{J}_1(\mathbf{q}, \mathbf{q}_1)]\dot{\mathbf{q}} \\ + \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{G}_k(\mathbf{q}_1) \\ = \boldsymbol{\tau}_k, (k = 3, 4) \end{aligned} \tag{45}$$

$$\begin{aligned} \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{M}_k(\mathbf{q}_2)\mathbf{J}_2(\mathbf{q}, \mathbf{q}_2)\ddot{\mathbf{q}} \\ + \dot{\mathbf{q}}^T * [\mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{M}_k(\mathbf{q}_2) * \mathbf{H}_2(\mathbf{q}) \end{aligned}$$

$$\begin{aligned} + \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2) * \mathbf{H}_{ck}(\mathbf{q}_2)\mathbf{J}_2(\mathbf{q}, \mathbf{q}_2)]\dot{\mathbf{q}} \\ + \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{G}_k(\mathbf{q}_2) = \boldsymbol{\tau}_k. (k = 5, 6) \end{aligned} \tag{46}$$

The dynamic equations with respect to \mathbf{q} can be derived by summing equations of all the links, i.e., Eqs. (37), (38), (45), (46), as follows,

$$\begin{aligned} \mathbf{M}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\dot{\mathbf{q}} \\ + \mathbf{G}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = \boldsymbol{\tau}, \end{aligned} \tag{47}$$

where $\mathbf{M}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = \sum_{k=1}^2 \mathbf{M}_k(\mathbf{q}) + \sum_{k=3}^4 \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)$

$$\mathbf{M}_k(\mathbf{q}_1)\mathbf{J}_1(\mathbf{q}, \mathbf{q}_1) + \sum_{k=5}^6 \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{M}_k(\mathbf{q}_2)\mathbf{J}_2(\mathbf{q}, \mathbf{q}_2),$$

$$\begin{aligned} \mathbf{H}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = \sum_{k=1}^2 \mathbf{H}_{ck}(\mathbf{q}) \\ + \sum_{k=3}^4 \left[\mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{M}_k(\mathbf{q}_1) * \mathbf{H}_1(\mathbf{q}) \right. \\ \left. + \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1) * \mathbf{H}_{ck}(\mathbf{q}_1)\mathbf{J}_1(\mathbf{q}, \mathbf{q}_1) \right] \\ + \sum_{k=5}^6 \left[\mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{M}_k(\mathbf{q}_2) * \mathbf{H}_2(\mathbf{q}) \right. \\ \left. + \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2) * \mathbf{H}_{ck}(\mathbf{q}_2)\mathbf{J}_2(\mathbf{q}, \mathbf{q}_2) \right], \end{aligned}$$

$$\mathbf{G}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = \sum_{k=1}^2 \mathbf{G}_k(\mathbf{q}) + \sum_{k=3}^4 \mathbf{J}_1^T(\mathbf{q}, \mathbf{q}_1)\mathbf{G}_k(\mathbf{q}_1)$$

$$+ \sum_{k=5}^6 \mathbf{J}_2^T(\mathbf{q}, \mathbf{q}_2)\mathbf{G}_k(\mathbf{q}_2),$$

$$\boldsymbol{\tau} = \sum_{k=1}^6 \boldsymbol{\tau}_k.$$

Equation (47) is the dynamic equations of the whole system, however, which is not represented by the active generalized coordinates \mathbf{q}_a . Similarly, use the force transformation matrices to derive the final dynamic model of the manipulator. The velocity and acceleration mapping between \mathbf{q} and \mathbf{q}_a have been obtained in Sect. 4.1, as follows,

$$\dot{\mathbf{q}}_a = \mathbf{J}_a(\mathbf{q}, \mathbf{q}_a)\mathbf{q}, \tag{48}$$

$$\ddot{\mathbf{q}}_a = \mathbf{J}_a(\mathbf{q}, \mathbf{q}_a)\ddot{\mathbf{q}} + \dot{\mathbf{q}}^T * \mathbf{H}_a(\mathbf{q})\mathbf{q}. \tag{49}$$

Therefore, the dynamic equations of the whole manipulator can be derived as

$$\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}}_a + \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \dot{\mathbf{q}}_a)\dot{\mathbf{q}}_a + \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = \boldsymbol{\tau}_a, \tag{50}$$

where $\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) = [\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]^T \mathbf{M}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) \mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)$,

$$\begin{aligned} \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \dot{\mathbf{q}}_a) &= \dot{\mathbf{q}}_a^T * [[\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]^T [\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]]^T \\ &* \mathbf{H}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) \mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a) \\ &- [\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]^T [\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]^T \\ \mathbf{M}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) \mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a) * \mathbf{H}_a(\mathbf{q}) \mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a), \\ \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2) &= [\mathbf{J}_a^{-1}(\mathbf{q}, \mathbf{q}_a)]^T \mathbf{G}(\mathbf{q}, \mathbf{q}_1, \mathbf{q}_2). \end{aligned}$$

Eq. (50) are the expected explicit closed-form dynamic equations for the analyzed 3-DOF spatial parallel manipulator, which is very useful for real-time control for a MIMO system. $\boldsymbol{\tau}_a = [f_1 \ f_2 \ f_3]^T$ is the vector of the three linear actuators' forces.

5 Simulation and experiment

5.1 Simulation validation

In this section, the inverse dynamics is used to validate the dynamic model. For the inverse dynamic problem, the time history of a predetermined trajectory is given. And the problem is to determine the active forces or torques required to produce that movement. That means the left parts of Eq. (50) can be calculated in terms of the movement.

In order to validate the correctness of the dynamic model, the results derived by the proposed approach and these obtained by ADAMS are compared. ADAMS is a mature and reliable commercial software which is based on numerical method for equation formulation [24]. ADAMS provides a good test bed for simulation. However, numerical expressions can't clearly provide physical insight into the system structure and can be difficult for engineers to develop better control laws with less known physics. Additionally, the model in ADAMS cannot be used for real-time control on physical prototypes. Those are the reasons for why the dynamic equations should be derived by algebraic formulae rather than by ADAMS. The kinematic and

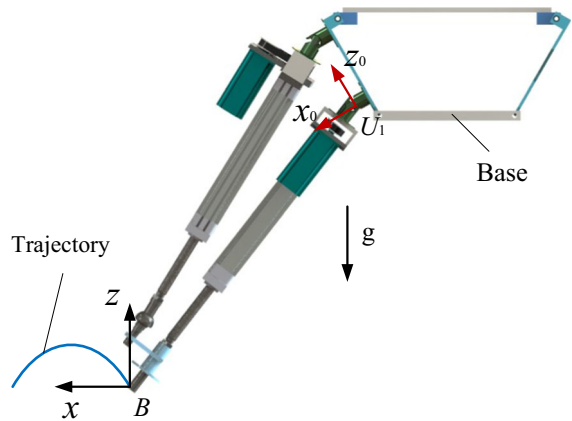


Fig. 3 Schematic diagram for simulation

dynamic parameters can be directly obtained from the virtual prototype in ADAMS. Table 1 shows the architecture and dynamic parameters of the 3-DOF manipulator depicted in Fig 1.

As shown in Fig. 3, we set the trajectory of the end-effector of the manipulator with respect to the inertial reference frame $B - xyz$ as follows

$$\begin{cases} x(t) = St/T - (S/2\pi) \sin(2\pi t/T) & t \in [0, T], \\ y(t) = 0 & t \in [0, T], \\ z(t) = 2Ht/T - (H/2\pi) \sin(2\pi t/T) & t \in [0, T/2], \\ z(t) = 2H - 2Ht/T + (H/2\pi) \sin(4\pi t/T) & t \in [T/2, T], \end{cases} \tag{51}$$

where T denotes the period of the movement, S and H determine the length and height of the movement. Set $T = 1\text{ s}$, $S = 200\text{ mm}$, $H = 100\text{ mm}$. As shown in Fig. 3, $U_1 - x_0y_0z_0$ has a 30° angle with respect to the vertical plane. Thus, the vector of gravitational acceleration in $U_1 - x_0y_0z_0$ is ${}^0\mathbf{g} = [-g \cdot \sin 30^\circ \ g \cdot \cos 30^\circ \ 0]^T$. The simulation results are shown in Fig. 4. From Fig. 4, we can see a good agreement between the results derived by ADAMS and those obtained from algebraic formulae, which validates the correctness of the dynamic equations and further means our method is right.

In addition, the dynamic model in the joint space is useful for optimization of motion planning, since the motion planning is affected by the capabilities of motors. Of course, some software, e.g., ADAMS and Solidworks, also can finish that work. As mentioned before, another role of the dynamic equations

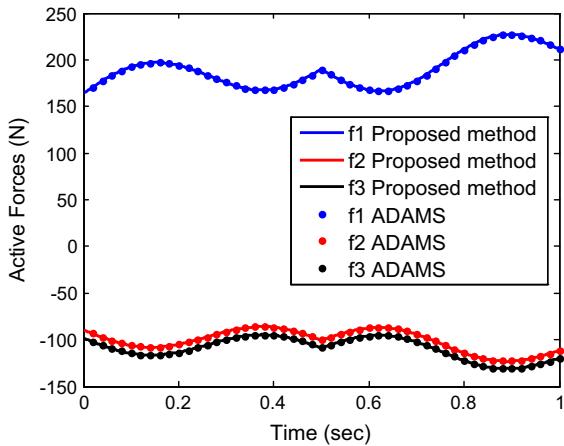


Fig. 4 Active forces computed by the proposed method and ADAMS

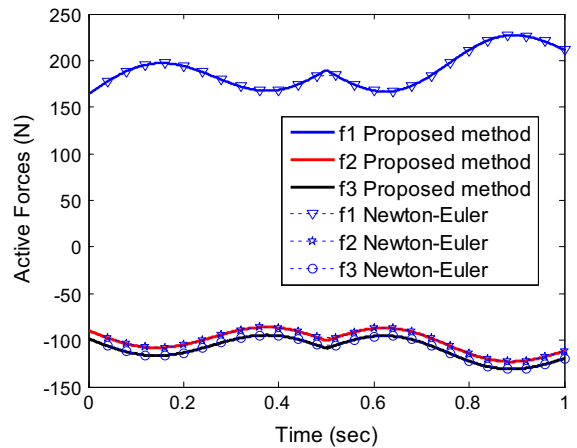


Fig. 6 Active forces computed by the proposed method and Newton–Euler method

that the commercial software can't replace is that the model can be used to analyze the actuator forces with respect to the physical nature. The inertial forces, the Coriolis and centrifugal forces, and the gravitational forces can be determined from Eq. (50), respectively, impossibly from ADAMS. Figure 5 shows the three different terms for the chosen trajectory of Eq. (51). We can see that the Coriolis and centrifugal forces are very small compared with the other two terms. Thus, in some cases, according to the simulation results, the terms which give small contribution on the active forces can be neglected in order to simplify the computations [47]. However, when the dynamic model is employed in model-based control, the neglecting for any part of dynamic equations will affect the control performance largely, which will be shown in Sect. 5.3.

5.2 Computational efficiency

In order to compare our method with existing methods, we also used the Newton–Euler method to obtain the dynamic equations of the 3-DOF manipulator. Figure 6 shows the comparison of simulation results obtained by Newton–Euler method and our method. Obviously, if the results are not identical, one of the methods must be wrong under the precondition of no miscalculation. The computer algebra program MAPLE is employed to implement the algorithms of dynamic modeling using these two methods. The programs are executed in a computer with Windows 8.1 equipped with Intel Core i7-4700HQ CPU. The computation time of our method to obtain the final dynamic equations is 22.461 s, whereas the time consumption of Newton–Euler method is 24.213 s.

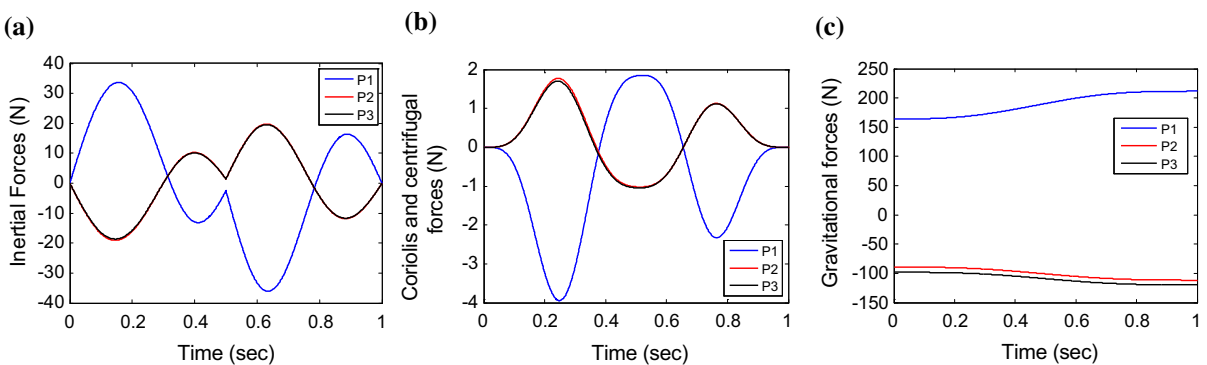
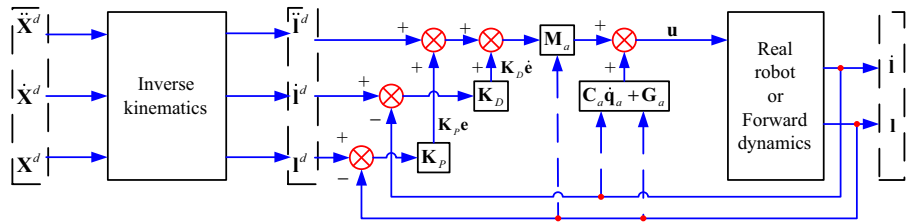


Fig. 5 Simulation results decomposed into three terms: **a** inertial forces, **b** coriolis and centrifugal forces, and **c** gravitational forces

Fig. 7 Computed-torque control scheme



In the earlier studies, some researchers [51–54] compared computational cost for different algorithms using the number of floating point operations, i.e., multiplications and additions. Gosselin [52] even parallelize the computation on several processors in order to improve the computational speed. Tsai [53], Khalil and Guegan [54] only derived the implicit models rather than the explicit models with very little computational cost. However, as it is recognized in [54], implicit dynamic model cannot be used for solving forward dynamics in simulation of motion control. Since the process of dynamic modeling is an offline work, we think it is not necessary to pursue faster computational efficiency. Moreover, some convenient software, such as MAPLE and MATLAB, can liberate researchers from complex computations. It’s more important to obtain precise explicit closed-form dynamics in the actuation space because of their crucial role for model-based control. In Sect. 5.3, we will demonstrate that the inverse and forward dynamics derived from our method could be used in online model-based control.

5.3 Implementation in model-based control

Computed-force control is chosen as the model-based control scheme to track the desired trajectory mentioned in Sect. 5.1, i.e., the profile $\mathbf{X} = [x(t) \ y(t) \ z(t)]^T$ represented by Eq. (51). As shown in Fig. 7, the control scheme is implemented in joint space based on the dynamic equations in the actuation space. The control law is

$$\mathbf{u} = \mathbf{M}_a(\ddot{\mathbf{q}}_a^d + \mathbf{K}_P\mathbf{e} + \mathbf{K}_D\dot{\mathbf{e}}) + \mathbf{C}_a\dot{\mathbf{q}}_a + \mathbf{G}_a, \quad (52)$$

where $\mathbf{K}_p = \text{diag}(k_p)_{3 \times 3}$, $\mathbf{K}_D = \text{diag}(k_D)_{3 \times 3}$. For a given control input \mathbf{u} , the resulting motion can be determined by the numerical integration of the actuator accelerations. As for the 3-DOF manipulator, we can obtain the discrete state space model according to Eq. (50), as follows,

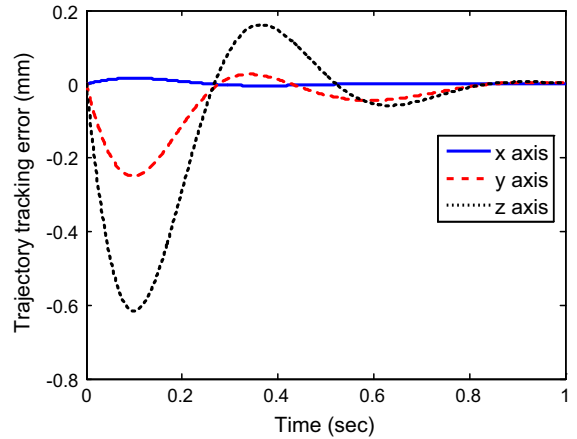


Fig. 8 Trajectory tracking error using precise inverse dynamics

$$\begin{cases} \mathbf{q}(k+1) = \text{diag}(T_s)_{3 \times 3} \dot{\mathbf{q}}(k) + \mathbf{q}(k), \\ \dot{\mathbf{q}}(k+1) = \text{diag}(T_s)_{3 \times 3} \mathbf{M}_a^{-1}(k) [\mathbf{u} - \mathbf{C}_a(k)\dot{\mathbf{q}}_a(k) - \mathbf{G}_a(k)], \end{cases} \quad (53)$$

where $T_s = 1ms$ denotes the sample period.

According to the simplified dynamic model suggested in literature [31–34], the control law has to use an inaccurate inverse dynamic model to compute the control input. Obviously, for a set of parameters of feedback gains, i.e., \mathbf{K}_P and \mathbf{K}_D , the more precise the inverse dynamic model is, the better control performance becomes. That is validated by the simulation results as shown in Figs. 8 and 9. With the same feedback gains $k_p = 160$, $k_D = 10$, the trajectory tracking performance based on precise inverse dynamics is much better than that depended on inverse dynamics neglecting Coriolis and centrifugal terms, which indicates the importance to obtain a precise dynamic model for the use of model-based control. As for other manipulators with larger inertia or tracking faster trajectories, the effect of model accuracy on control performance will be highlighted. The control simulation algorithm is executed by MATLAB on the same computer mentioned in Sect. 5.2. Functions “tic” and “toc” of MAT-

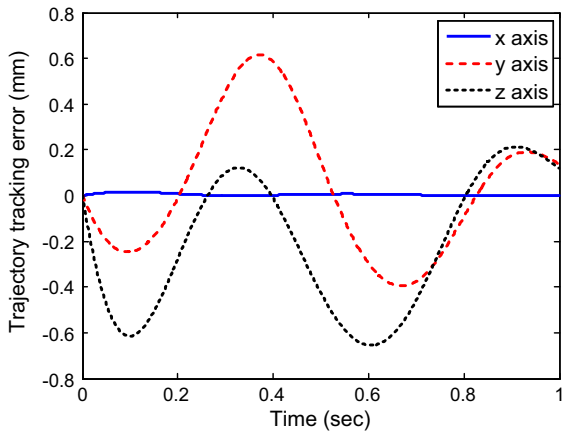


Fig. 9 Trajectory tracking error using inaccurate inverse dynamic without Coriolis and centrifugal part

LAB give the run time of the control cycle, which is 0.66646 ms. Several tasks, such as motion planning and the computation of inverse dynamics, should be accomplished in the control cycle. Among a control cycle, the computation time for inverse dynamics is 0.52695 ms. Generally, the servo control cycle for a commercial control-hardware is between 1 ms and tens of milliseconds. Therefore, the dynamic model derived by our modeling method is feasible for online control.

5.4 Experiment analysis

Similar to the CAD model displayed in Fig. 1, a test bed has been built as shown in Fig. 10. The linear actuator is composed of a 400 W DC servo motor, a timing belt transmission with 2:1 reduction ratio and a ball screw with 5 mm lead. The parameters of the test bed are identical to those shown in Table 1 with the exception of the mass parameters. Here, the parameters of mass and moment of inertia are a bit smaller than those shown in Table 1. The motors are driven by Elmo’s motor drives (G-MOLWHI15/100EE), which are connected to an embedded controller (CX2030) made by the company Beckhoff through an EtherCAT bus. Each motor drive has three control modes: position loop mode, velocity loop mode and torque loop mode. Additionally, the motor drives can provide real-time values of displacement, velocity, acceleration and torque. We have done two experiments using the position loop mode and the torque loop mode respectively.



Fig. 10 Photograph of the test bed

The sever motor drive can give real-time torques of motors, whereas Eq. (50) gives the push/pull forces of prismatic joints. There is a transformation relationship between motor torque and push/pull force in a linear actuator composed of timing belt transmission and ball screw, as follows,

$$\tau_{mi} = \frac{L}{2\pi R} \tau_{ai}, \quad (i = 1, 2, 3) \tag{54}$$

where L is the lead of the ball screw; R is the reduction ratio of the timing belt transmission; τ_{mi} denotes the motor torque of each linear actuator; τ_{ai} denotes the active forces of prismatic joints. Therefore, Eq. (50) can be transformed into the following style in terms of motor torques,

$$(\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}}_a + \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \dot{\mathbf{q}}_a)\dot{\mathbf{q}}_a + \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2))L/(2\pi R) = \boldsymbol{\tau}_m. \tag{55}$$

The friction contributions must be incorporated into the rigid body dynamic equations given in Eq. (55) because friction effects are usually important for actual industrial robots [48]. It has been observed that friction can cause more than 50% error in some heavy industrial manipulators [49]. The classical Stribeck friction model [50] is adopted in this paper. Including friction the dynamic equations of a manipulator can be written as

Table 1 Parameters of the 3-DOF parallel manipulator

Parameter	Value (Kg)	Parameter	Value (Kg $m^2 10^{-4}$)	Parameter	Value (mm)
m_1	3.888	I_{c1}	diag[101 865 908]	a_y	150
m_2	1.623	I_{c2}	diag[2 189 206]	a_z	72.5
m_3	3.888	I_{c3}	diag[101865 908]	b_y	65
m_4	0.584	I_{c4}	diag[0.54 93 99]	b_z	30
m_5	3.888	I_{c5}	diag[101865 908]	d	136
m_6	0.584	I_{c6}	diag[0.54 93 99]	$l_{1 t=0}$	578.5
				$l_{2 t=0}$	642.5
				$l_{3 t=0}$	642.5

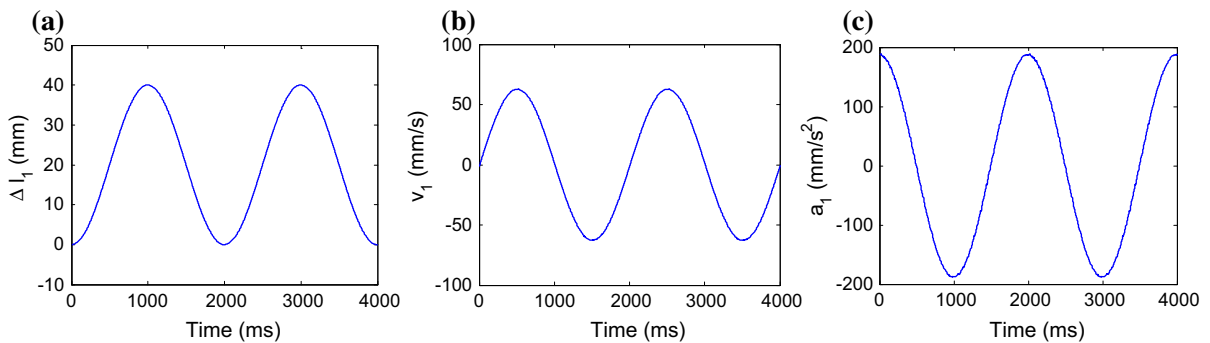


Fig. 11 Measured motion of P1 in the first experiment using position loop mode: **a** displacement, **b** velocity, **c** acceleration

$$\begin{aligned}
 & (\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}}_a + \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_a)\dot{\mathbf{q}}_a \\
 & + \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2))L/(2\pi R) + \boldsymbol{\tau}_f = \boldsymbol{\tau}_m, \quad (56)
 \end{aligned}$$

where $\boldsymbol{\tau}_f$ is the friction torque vector. However, identifying the friction parameters for each joint is very difficult. The friction forces of passive joints are much smaller than those of active joints because of the pre-stress force on the ball screw. Thus, after neglecting the friction of passive joints, Eq. (54) is rewritten as

$$\begin{aligned}
 & (\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}}_a + \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_a)\dot{\mathbf{q}}_a \\
 & + \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2))L/(2\pi R) + \boldsymbol{\tau}_{fa} = \boldsymbol{\tau}_m, \quad (57)
 \end{aligned}$$

where $\boldsymbol{\tau}_{fa} = [\tau_{fa1} \ \tau_{fa2} \ \tau_{fa3}]^T$ is the friction torque vector of the three linear actuators. The Stribeck friction model for each prismatic joint is as follows,

$$\begin{aligned}
 \tau_{fai} &= [F_c + (F_s - F_c)e^{-|\dot{l}_i/v_s|^\delta}] \text{sgn}(\dot{l}_i) \\
 &+ F_v \dot{l}_i, \quad (i = 1, 2, 3) \quad (58)
 \end{aligned}$$

where the unit of τ_{fai} is N · m; the unit of \dot{l}_i is mm/s. The friction model is identified by some experimental data using least square method, as follows,

$$\begin{aligned}
 \tau_{fai} &= [0.059 + (0.07 - 0.059)e^{-|\dot{l}_i/0.3|^{1.5}}] \text{sgn}(\dot{l}_i) \\
 &+ 0.002\dot{l}_i. \quad (i = 1, 2, 3) \quad (59)
 \end{aligned}$$

Moreover, the inertia of motor’s shaft and of the ball screw should be added, and these two contributions are defined as \mathbf{I}_m . Then, the complete dynamic equations of the manipulator explicitly showing the motor torques for the convenience of experiments can be expressed as follows,

$$\begin{aligned}
 & (\mathbf{M}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2)\ddot{\mathbf{q}}_a + \mathbf{C}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_a)\dot{\mathbf{q}}_a \\
 & + \mathbf{G}_a(\mathbf{q}_a, \mathbf{q}, \mathbf{q}_1, \mathbf{q}_2))L/(2\pi R) \\
 & + \boldsymbol{\tau}_{fa} + \mathbf{I}_m\ddot{\boldsymbol{\theta}}_m = \boldsymbol{\tau}_m, \quad (60)
 \end{aligned}$$

where $\ddot{\boldsymbol{\theta}}_m = [\ddot{\theta}_{m1} \ \ddot{\theta}_{m2} \ \ddot{\theta}_{m3}]^T$ is the angular acceleration vector of the three motors, and $\ddot{\theta}_{mi} = 2\pi R\dot{l}_i/L$ ($i = 1, 2, 3$). $\boldsymbol{\tau}_m = [\tau_{m1} \ \tau_{m2} \ \tau_{m3}]^T$ is the torque vector of the three motors.

In the first experiment, the motor of P1 worked under the position loop mode. Set the motion of P1 as a sine function, while P2 and P3 are locked. As shown in Fig. 11, the motion of P1 measured by an encoder is very smooth. The simulated motor torques can be obtained

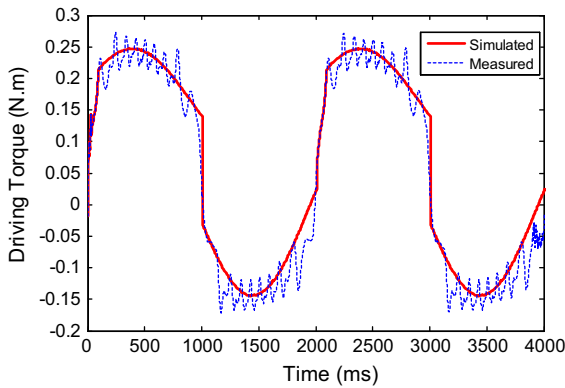


Fig. 12 Comparison between measured and estimated torques in the first experiment

by introducing the motion data shown in Fig. 11 into Eqs. (50) and (60). As depicted in Fig. 12, the torque of the motor shows some fluctuation while the result calculated by the dynamic model is very smooth. The reason for the fluctuation of the measured value is that the controller of the driver should change the torque to follow the motion function since the friction is varying during a rotational circle of the ball screw. In the second experiment, the P1 motor works under the torque loop mode, then the motor drive will try to guarantee the predetermined profile of the motor torque. The measured torque and simulated torque are shown in Fig. 14. In contrast to the results of the first experiment, the motion of P1 shows large fluctuation as shown in Fig. 13 while the actual torque is smooth. The explanation of the fluctuation in this second experiment also indicates the variation of friction over the rotational circle of the ball screw. In the same way, the model-computed torque of the second experiment can be obtained by introducing the measured displacement, velocity and

acceleration data into the dynamic equations. Neglecting the fluctuation caused by friction, the measured results and dynamic model outputs have a good fitting in both two experiments, which validates the correctness of the dynamic model as well. However, if it is desired to increase the precision of the simulated results, the variation of the friction with the rotation of the motor shaft should be taken into account. Whereas the main purpose of this paper is to derive the multi-rigid-body dynamics, a more accurate model of friction will be discussed in the future work.

5.5 Trajectory tracking experiments

In this section, the trajectory planned in Sect. 5.1, i.e., the profile $\mathbf{X} = [x(t) \ y(t) \ z(t)]^T$ represented by Eq. (51), was tracked by using classic PD control and computed-torque control, respectively. Classic PD controller is implemented in each single motor with the same parameters of $k_P = 500$ and $k_D = 20$. The friction model represented by Eq. (59) is added to the computed-torque controller motioned in Sect. 5.3. Other parameters of the computed-torque controller are identical with those in Sect. 5.3. Control programs are explored using ST language that is similar to C language. The servo control cycle in the control-hardware CX2030 produced by Beckhoff is 1 ms, which is enough to accomplish all tasks including motion planning and the computation of inverse kinematics and dynamics for computed-torque control. The control performance is shown in Fig. 15. As shown in Fig. 15, the peak error occurs in the middle, because motor's rotary direction changes around that time. As expected, model-based control outperforms the classic PD con-

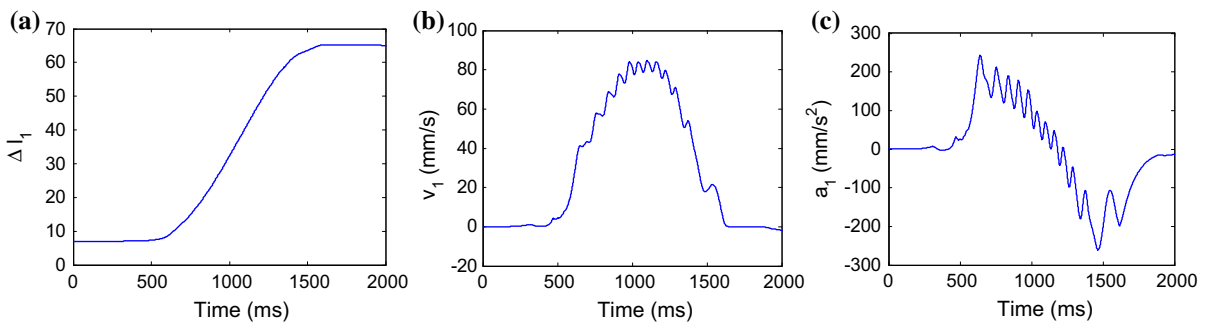


Fig. 13 Measured motion of P1 in the second experiment using torque loop mode: **a** displacement, **b** velocity, **c** acceleration

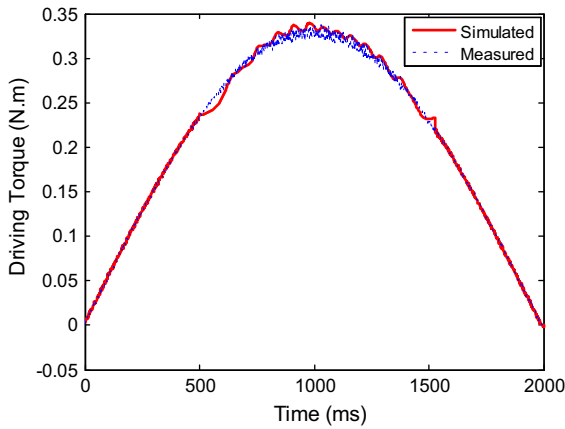


Fig. 14 Comparison between measured and estimated torques in the second experiment

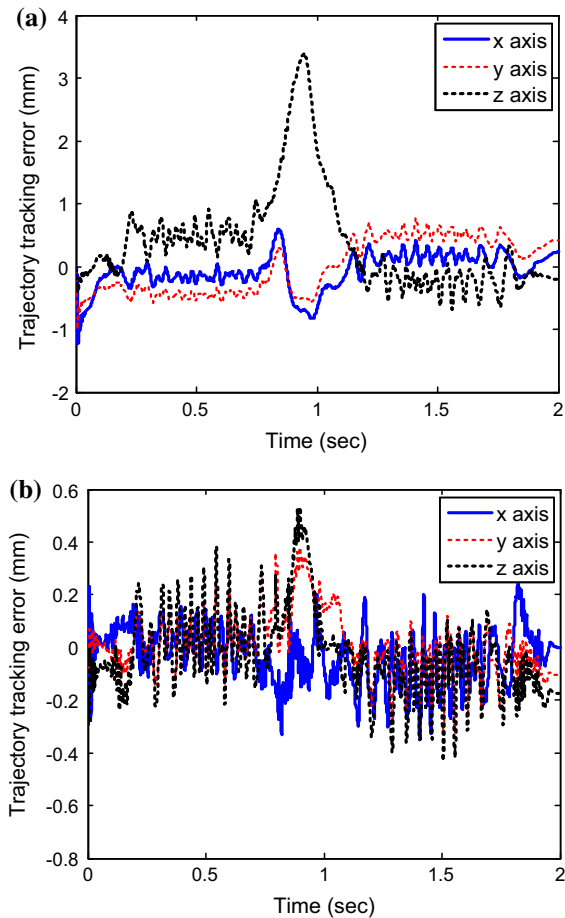


Fig. 15 Comparison of trajectory tracking control errors in experiments by using classical PID control and model-based control: **a** classic PID control, **b** computed-torque control

trol significantly due to the compensation of dynamic model.

6 Conclusions

This paper presents a systematic methodology for determining the inverse dynamics equations of complex non-redundant parallel manipulators by the combination of the Lagrangian formulation and the virtual work principle. Based on the proposed algorithm which is summarized as five steps, the dynamic equations of a 3-DOF spatial parallel manipulator have been obtained and shown to be valid by comparison with 3rd parts software and by experiments. The determination of the dynamic equations of a complex multi-closed-loop mechanism can be simplified by splitting the system into independent substructures with respect to suitable generalized coordinates. The constraints are introduced by the way of Jacobian and Hessian matrices. The Jacobian and Hessian matrices keep the explicit attributes of the dynamic equations, when they are transformed into different coordinate subsystems according to the virtual work principle. The study presented here provides a sound basis for future work on the context of model-based control. Moreover, the modeling approach and valid verification method can also be applied to other parallel mechanisms. In the future, the model-based control will be studied using the derived dynamic equations of the manipulator.

Acknowledgments This work was supported by National Basic Research Program (973) of China (Grant No. 2013CB035504), National Natural Science Foundation of China (Grant No. 51405515), and the Self-selected Topic Fund of State Key Laboratory of High-Performance Complex Manufacturing (Grant No. zzyjkt2015-02). Additionally, the authors want to express their gratitude to the reviewers of this paper for their constructive remarks and suggestions.

Appendix 1: Detailed expression of the dynamic equations of Link 2

The dynamic equations of Link 2 are given as follows:

$$\mathbf{M}_2(\mathbf{q}) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix},$$

$$M_{11} = m_2(-l_1 y_{c2} - x_{c2} y_{c2}) \sin(2\theta_2) + m_2(0.5l_1^2 + l_1 x_{c2} + 0.5x_{c2}^2 - 0.5y_{c2}^2) \cos(2\theta_2)$$

$$\begin{aligned}
 &+ m_2(0.5l_1^2 + l_1x_{c2} + 0.5x_{c2}^2 + 0.5y_{c2}^2), \\
 M_{12} &= L_{2xy} \sin \theta_1 + L_{2yz} \cos \theta_1, \\
 M_{13} &= 0, \\
 M_{21} &= L_{2xy} \sin \theta_1 + L_{2yz} \cos \theta_1, \\
 M_{22} &= m_2(x_{c2}^2 + 2x_{c2}l_1 \\
 &\quad + y_{c2}^2 + l_1^2) + 0.5L_{2xx} - 0.5L_{2xx} \cos(2\theta_1) \\
 &\quad + L_{2xz} \sin(2\theta_1) + 0.5L_{2zz} \cos(2\theta_1) + 0.5L_{2zz}, \\
 M_{23} &= -m_2y_{c2}, \\
 M_{31} &= 0, \\
 M_{32} &= -m_2y_{c2}, \\
 M_{33} &= m_2.
 \end{aligned}$$

$$\mathbf{H}_{c2}(\mathbf{q}) = \begin{bmatrix} \begin{bmatrix} H_{111} \\ H_{112} \\ H_{113} \end{bmatrix} & \begin{bmatrix} H_{121} \\ H_{122} \\ H_{123} \end{bmatrix} & \begin{bmatrix} H_{131} \\ H_{132} \\ H_{133} \end{bmatrix} \\ \begin{bmatrix} H_{211} \\ H_{212} \\ H_{213} \end{bmatrix} & \begin{bmatrix} H_{221} \\ H_{222} \\ H_{223} \end{bmatrix} & \begin{bmatrix} H_{231} \\ H_{232} \\ H_{233} \end{bmatrix} \\ \begin{bmatrix} H_{311} \\ H_{312} \\ H_{313} \end{bmatrix} & \begin{bmatrix} H_{321} \\ H_{322} \\ H_{323} \end{bmatrix} & \begin{bmatrix} H_{331} \\ H_{332} \\ H_{333} \end{bmatrix} \end{bmatrix},$$

$$\begin{aligned}
 H_{111} &= 0, \\
 H_{112} &= m_2(l_1^2 + c_{2x}^2 + c_{2y}^2 + 2l_1c_{2x}) \sin(2\theta_2) \\
 &\quad + m_2 \cos(2\theta_2)(2l_1c_{2y} + 2c_{2x}c_{2y}), \\
 H_{113} &= m_2(l_1 + x_{c2} + x_{c2} \cos(2\theta_2) \\
 &\quad + y_{c2} \sin(2\theta_2) + l_1 \cos(2\theta_2)), \\
 H_{121} &= 0, \\
 H_{122} &= -0.5L_{2xx} \sin(2\theta_1) - L_{2xz} \cos(2\theta_1) \\
 &\quad + 0.5L_{2zz} \sin(2\theta_1), \\
 H_{123} &= 0, \quad H_{131} = 0, \quad H_{132} = 0, \quad H_{133} = 0, \\
 H_{211} &= L_{2yz} \sin \theta_1 + L_{2xy} \cos \theta_1 \\
 &\quad + m_2l_1x_{c2} \sin(2\theta_2) \\
 &\quad + m_2x_{c2}y_{c2} \cos(2\theta_2) + 0.5m_2l_1^2 \sin(2\theta_2) \\
 &\quad + 0.5m_2x_{c2}^2 \sin(2\theta_2) - 0.5m_2y_{c2}^2 \sin(2\theta_2), \\
 H_{212} &= L_{2xx} \sin(2\theta_1) + 2L_{2xz} \cos(2\theta_1) - L_{2zz} \sin(2\theta_1), \\
 H_{213} &= 0, \quad H_{221} = 0, \quad H_{222} = 0, \quad H_{223} = 2m_2l_1 \\
 &\quad + 2m_2x_{c2}, \quad H_{231} = 0, \quad H_{232} = 0, \quad H_{233} = 0, \\
 H_{311} &= 0.5m_2(l_1 \cos(2\theta_2)
 \end{aligned}$$

$$\begin{aligned}
 &+ x_{c2} \cos(2\theta_2) - y_{c2} \sin(2\theta_2) + l_1 + x_{c2}), \\
 H_{312} &= 0, \quad H_{313} = 0, \quad H_{321} = 0, \quad H_{322} \\
 &= m_2(l_1 + x_{c2}), \quad H_{323} = 0, \\
 H_{331} &= 0, \quad H_{332} = 0, \quad H_{333} = 0
 \end{aligned}$$

G₂(q)

$$= \begin{bmatrix} 0.5m_2g((x_{c2} + l_1) \sin \theta_1 \cos \theta_2 \\ -y_{c2} \sin \theta_1 \sin \theta_2) \\ 0.5m_2g((x_{c2} + l_1) \cos \theta_1 \sin \theta_2 \\ +y_{c2} \cos \theta_1 \cos \theta_2) + 0.866g((x_{c2} + l_1) \cos \theta_2 \\ -y_{c2} \sin \theta_2) \\ m_2g(-0.5 \cos \theta_1 \cos \theta_2 + 0.866 \sin \theta_2) \end{bmatrix}.$$

References

1. Choi, H., Ryu, J.: Singularity analysis of a four degree-of-freedom parallel manipulator based on an expanded 6 × 6 Jacobian matrix. *Mech. Mach. Theory* **57**, 51–61 (2012)
2. Tsai, M., Shiau, T., Tsai, Y., Chang, T.: Direct kinematic analysis of a 3-PRS parallel mechanism. *Mech. Mach. Theory* **38**, 71–83 (2003)
3. Li, Y., Xu, Q.: Kinematic analysis of a 3-PRS parallel manipulator. *Robot. Comput.-Integr. Manuf.* **23**, 395–408 (2007)
4. Ding, W., Deng, H., Li, Q., Xia, Y.: Control-orientated dynamic modeling of forging manipulators with multi-closed kinematic chains. *Robot. Comput.-Integr. Manuf.* **30**, 421–431 (2014)
5. Li, Y., Wang, J., Wang, L., Liu, X.: Inverse dynamics and simulation of a 3-DOF spatial parallel manipulator. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003)*, pp. 4092–4097 (2003)
6. Geng, Z., Haynes, L., Lee, J., Carroll, R.: On the dynamic model and kinematic analysis of a class of Stewart platforms. *Robot. Auton. Syst.* **9**, 237–254 (1992)
7. Dasgupta, B., Mruthyunjaya, T.: Closed-form dynamic equations of the general Stewart platform through the Newton–Eular approach. *Mech. Mach. Theory* **33**, 993–1012 (1998)
8. Kane, T.R., Levinson, D.A.: The use of Kane’s dynamical equations in robotics. *Int. J. Rob. Res.* **2**, 3–21 (1983)
9. Kane, T.R., Levinson, D.A.: Formulation of equations of motion for complex spacecraft. *J. Guid. Control Dynam.* **3**, 99–112 (1980)
10. Huston, R.L., Passerello, C.E.: On Lagranges form of d’Alembert’s principle. *Matrix Tensor Quart.* **23**, 109–112 (1973)
11. Huston, R.L., Passerello, C.E., Harlow, M.W.: Dynamics of multirigid-body systems. *J. Appl. Mech.* **45**, 889–894 (1978)
12. Huston, R.L., Passerello, C.: On multi-rigid-body system dynamics. *Comput. Struct.* **10**, 439–446 (1979)
13. Huston, R.L., Passerello, C.E.: Multibody structural dynamics including translation between the bodies. *Comput. Struct.* **12**, 713–720 (1980)

14. Kamman, J.W., Huston, R.L.: Constrained multibody system dynamics an automated approach. *Comput. Struct.* **18**, 999–1003 (1984)
15. Wang, J.T., Huston, R.L.: Kane's equations with undetermined multipliers—application to constrained multibody systems. *J. Appl. Mech.* **54**, 424–429 (1987)
16. Lin, H.H., Huston, R.L., Coy, J.J.: On dynamic loads in parallel shaft transmissions: Part I—modelling and analysis. *J. Mech., Trans., and Automation* **110**, 221–225 (1988)
17. Ider, S.K., Amirouche, F.M.L.: Coordinate reduction in the dynamics of constrained multibody systems—a new approach. *J. Appl. Mech.* **55**, 899–904 (1988)
18. Amirouche, F.M.L., Jia, T.: Pseudouptriangular decomposition method for constrained multibody systems using Kane's equations. *J. Guid. Control Dynam.* **11**, 39–46 (1988)
19. Amirouche, F.M.L., Jia, T., Ider, S.K.: A recursive householder transformation for complex dynamical systems with constraints. *J. Appl. Mech.* **55**, 729–734 (1988)
20. Ider, S.K., Amirouche, F.M.L.: Numerical stability of the constraints near singular positions in the dynamics of multibody systems. *Comput. Struct.* **33**, 129–137 (1989)
21. Amirouche, F.M.L., Ider, S.K., Trimble, J.: Analytical method for the analysis and simulation of human locomotion. *J. Biomech. Eng.* **112**, 379–386 (1990)
22. Amirouche, F.M.L., Shareef, N.H.: Gain in computational efficiency by vectorization in the dynamic simulation of multi-body systems. *Comput. Struct.* **41**, 293–302 (1991)
23. Amirouche, F.M.L., Xie, M.: An explicit matrix formulation of the dynamical equations for flexible multibody systems: a recursive approach. *Comput. Struct.* **46**, 311–321 (1993)
24. Lieh, J.: Computer-oriented closed-form algorithm for constrained multibody dynamics for robotics applications. *Mech. Mach. Theory* **29**, 357–371 (1994)
25. Lieh, J. Computer-aided modeling and control of multibody systems for robotics application. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1991)*, pp. 2385–2390 (1991)
26. Tsai, L.: Solving the inverse dynamics of a Stewart–Gough manipulator by the principle of virtual work. *J. Mech. Des* **122**, 3–9 (2000)
27. Abdellatif, H., Grotjahn, M., Heimann, B.: High efficient dynamics calculation approach for computed-force control of robots with parallel structures. In: *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC'05)*, pp. 2024–2029 (2005)
28. Codourey, A., Burdet, E.: A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1997)*, pp. 1612–1618 (1997)
29. Enferadi, J., Tootoonchi, A.: Inverse dynamics analysis of a general spherical star-triangle parallel manipulator using principle of virtual work. *Nonlinear Dyn.* **61**, 419–434 (2010)
30. Guo, H., Li, H.: Dynamic analysis and simulation of a six degree of freedom Stewart platform manipulator. *P. I. Mech. Eng. C-J. Mec.* **220**, 61–72 (2006)
31. Nguyen, C., Pooran, F.: Dynamic analysis of a 6 DOF CKCM robot end-effector for dual-arm telerobot systems. *Robot. Auton. Syst.* **5**, 377–394 (1989)
32. Lebre, G., Liu, K., Lewis, F.L.: Dynamic analysis and control of a Stewart platform manipulator. *J. Robot. Syst.* **10**, 629–655 (1993)
33. Lee, K., Shah, D.: Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Trans. Robot. Autom.* **4**, 361–367 (1988)
34. Caccavale, F., Siciliano, B., Villani, L.: The Tricept robot: dynamics and impedance control. *IEEE-ASME T. Mech.* **8**, 263–268 (2003)
35. Chen, C.: A Lagrangian formulation in terms of quasi-coordinates for the inverse dynamics of the general 6-6 Stewart platform manipulator. *JSME Int. J. Ser. C* **46**, 1084–1090 (2003)
36. Grotjahn, M., Heimann, B., Abdellatif, H.: Identification of friction and rigid-body dynamics of parallel kinematic structures for model-based control. *Multibody Syst. Dyn.* **11**, 273–294 (2004)
37. Zhu, S., Huang, Z., Ding, H.: Forward/reverse velocity and acceleration analysis for a class of lower-mobility parallel mechanisms. *J. Mech. Des* **129**, 390–396 (2007)
38. Huang, Z.: Modelling formulation of 6-dof multi-loop parallel manipulators, part 1-Kinematic influence matrix. In: *Proceedings of 4th IFToMM conference on mechanisms and CAD*, pp. 163–170 (1985)
39. Singh, R.P., Likins, P.W.: Singular value decomposition for constrained dynamical systems. *J. Appl. Mech.* **52**, 943–948 (1985)
40. Stewart, D.: A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **180**, 371–386 (1965)
41. Liang, Q., Zhang, D., Wang, Y., Coppola, G., Ge, Y.: PM based multi-component F/T sensors—State of the art and trends. *Robot. Comput.-Integr. Manuf.* **29**, 1–7 (2013)
42. Gao, Z., Zhang, D.: Flexure parallel mechanism: configuration and performance improvement of a compact acceleration sensor. *J. Mech. Robot* **4**, 031002 (2012)
43. Li, Y., Xu, Q.: Kinematic analysis and design of a new 3-DOF translational parallel manipulator. *J. Mech. Des* **128**, 729–737 (2006)
44. Pierrot, F., Reynaud, C., Fournier, A.: DELTA: a simple and efficient parallel robot. *Robotica* **8**, 105–109 (1990)
45. Poppeová, V., Bulej, V.: Development of simulation software and control system for mechanism with hybrid kinematic structure. In: *Proceedings of 2010 41st International Symposium on Robotics (ISR) and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1304–1309 (2010)
46. Huang, T., Li, M., Zhao, X., Mei, J., Chetwynd, D., Hu, S.: Conceptual design and dimensional synthesis for a 3-DOF module of the TriVariant—a novel 5-DOF reconfigurable hybrid robot. *IEEE Trans. Robot.* **21**, 449–456 (2005)
47. Li, Y., Staicu, S.: Inverse dynamics of a 3-PRC parallel kinematic machine. *Nonlinear Dyn.* **67**, 1031–1041 (2012)
48. Bona, B., Indri, M.: Friction compensation in robotics: an overview. In: *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC'05)*, pp. 4360–4367 (2005)
49. Kermani, M., Wong, M., Patel, R., Moallem, M., Ostojic, M.: Friction compensation in low and high-reversal-velocity manipulators. In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, pp. 4320–4325 (2004)

50. Armstrong-Hélouvry, B.: *Control of Machines with Friction*. Springer, New York (1991)
51. Hollerbach, J.M.: A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. *IEEE Trans. Syst. Man, Cybern. Syst.* **10**, 730–736 (1980)
52. Gosselin, C. M.: Parallel computational algorithms for the kinematics and dynamics of parallel manipulators. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1993)*, pp. 883–888 (1993)
53. Tsai, L.W.: Solving the inverse dynamics of a Stewart–Gough manipulator by the principle of virtual work. *J. Mech. Des* **122**, 3–9 (2000)
54. Khalil, W., Guegan, S.: Inverse and direct dynamic modeling of Gough–Stewart robots. *IEEE Trans. Robot.* **20**, 754–761 (2004)