

Article

A Geometric Approach for Real-Time Forward Kinematics of the General Stewart Platform

Fangfang Yang ¹, Xiaojun Tan ¹, Zhe Wang ², Zhenfeng Lu ¹ and Tao He ^{1,*}

¹ School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510275, China; yangff7@mail.sysu.edu.cn (F.Y.); tanxj@mail.sysu.edu.cn (X.T.); luzhf6@mail2.sysu.edu.cn (Z.L.)

² Department of Advanced Design and Systems Engineering, City University of Hong Kong, Hong Kong; dr.zhe.wang@outlook.com

* Correspondence: hetao29@mail.sysu.edu.cn

Abstract: This paper presents a geometric approach for real-time forward kinematics of the general Stewart platform, which consists of two rigid bodies connected by six general serial manipulators. By describing the rigid-body motion as exponential of twist, and taking advantage of the product of exponentials formula, a step-by-step derivation of the proposed algorithm is presented. As the algorithm naturally solves all passive joint displacements, the correctness is then verified by comparing the forward-kinematic solutions from all chains. The convergence ability and robustness of the proposed algorithm are demonstrated with large amounts of numerical simulations. In all test cases, the proposed algorithm terminates within four iterations, converging with near-quadratic speed. Finally, the proposed algorithm is also implemented on a mainstream embedded motion controller. Compared with the incremental method, the proposed algorithm is more robust, with an average execution time of 0.48 ms, meeting the requirements of most applications, such as kinematic calibration, motion simulation, and real-time control.



Citation: Yang, F.; Tan, X.; Wang, Z.; Lu, Z.; He, T. A Geometric Approach for Real-Time Forward Kinematics of the General Stewart Platform. *Sensors* **2022**, *22*, 4829. <https://doi.org/10.3390/s22134829>

Academic Editor: Theodore E. Matikas

Received: 27 May 2022

Accepted: 23 June 2022

Published: 26 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: the Stewart platform; forward kinematics; geometric approach; real-time

1. Introduction

The Stewart platform, originally proposed by D. Stewart in 1965 [1], is one of the most celebrated parallel manipulators. As illustrated in Figure 1, it has a mounting base B and a moving platform P, connected by six linear actuators via spherical/universal joints. The Stewart platform has wide applications in flight simulation, animatronics, crane technology, underwater research, robotics and so on. Compared with its serial counterparts, the Stewart platform has higher force/torque capacity, greater structure rigidity and superior positioning accuracy. However, the complex coupling of chains also greatly increases the difficulties in forward kinematics, which is essential for both theoretical analysis and practical applications.

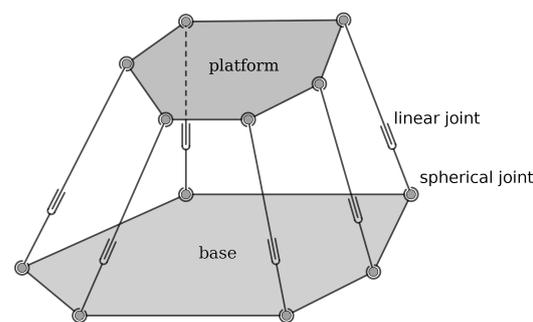


Figure 1. The 6-SPS Stewart platform.

For the real-time control of the general Stewart platform, either with optimized design or after kinematic calibration, a fast forward kinematics algorithm must be developed. Denoting b_i and p_i as the center points of the spherical joints corresponding to the i -th chain, the forward kinematics of the Stewart platform is equivalent to solving the following equations [2]:

$$\|\mathbf{R}p_i + \mathbf{T} - b_i\| = L_i, \text{ for } i = 1, 2, \dots, 6, \quad (1)$$

where \mathbf{R} and \mathbf{T} are the rotation matrix and translation vector of the moving platform to be solved, L_i is the given length of i -th leg.

Existing methods can be generally divided into four categories, analytical methods, numerical methods, extra-sensor methods, and intelligent algorithms. Initially, most researchers focused on finding all possible closed forms by considering special configurations where some of the base or platform spherical joints coincide. Griffis et al. [3] considered the forward kinematics of a 3–3 Stewart platform and obtained an eighth degree polynomial. Lin et al. [4] solved the forward kinematics of the 4–4 case. More difficult cases, such as 5–4 [5,6] and 6–4 [7] Stewart platforms were also solved analytically. The final 6–6 case was independently analysed and solved, such as in Wen et al. [8], Sreenivasan et al. [9], and Dasgupta et al. [10]. Further study of Equation (1) showed that the forward kinematics eventually boiled down to a univariate polynomial of 40-th order [11–13]. To actually compute these solutions, a number of root-finding methods were then proposed, including algebraic elimination [14–16], interval analysis [17], and continuation [12,18].

Although analytical methods can obtain all possible solutions to forward kinematics, they are sensitive to numerical errors. Moreover, in real applications, an extra procedure must be presented to determine a unique actual pose from all possible solutions. Numerical methods and extra-sensor methods are two common schemes for this purpose. Since forward kinematics essentially boils down to solving non-linear equations, Newton–Raphson iterative algorithms [19–21] were adopted extensively for their high computational efficiency and fast convergence speed. Xie et al. [22] formulated the forward kinematics as an unconstrained optimization problem on $SE(3)$ and compared their results with the Euler angle and quaternion parametrization schemes. However, the convergence of Newton–Raphson method depends heavily on the selection of initial values. Yang et al. [23] proposed the global Newton–Raphson method with monotonic descent algorithm to achieve global convergence regardless of the initial guess. Wang [24] tackled the issue by adopting a series of small leg length changes to finally obtain a unique forward kinematic solution. Pratik et al. [25] proposed a hybrid strategy that used a neural network as a bootstrap for the standard Newton–Raphson algorithm to yield a unique solution. The forward kinematics can also be settled with auxillary sensors. To simplify the sophisticated forward kinematics, as well as obtain a unique solution simultaneously, Bonev et al. [26] used three linear sensors connecting the moving platform and the base, Chiu et al. [27] used two rotatory sensors and one displacement sensor, and Cheng et al. [28] used one linear sensor to measure the distance of the moving platform. Although the extra-sensor method is proven effective, it comes with extra economic cost as well as assembly and measurement errors, which significantly limits the application range.

Recently, some researchers had also used machine learning techniques, such as support vector machines [29] and artificial neural network [30,31], to solve the problem. Chauhan et al. [32] proposed soft-computing-based schemes to resolve the forward kinematics of the Stewart platform, in which a neural-network-based function estimator was proposed for the forward kinematics and then trained the network with meta-heuristic optimization procedures to revise the weights and bias values of the neural network. The major problem with learning techniques is that the training process requires an abundant amount of data and usually takes hours for fine-tuning. Moreover, if kinematic calibration is present, the training process has to be done for every machine.

Although a lot of research has been done on the forward kinematics of the 6-SPS, the forward kinematics of the general Stewart platform has not yet been fully explored. Most

existing works focused on the standard 6-SPS structure, very few authors considered other variants of the Stewart platform such as 6-RUS [33] and 6-RSS [34]. The so-called general Stewart platform studied in existing works is actually a 6-SPS type platform with non-planar base and moving platform, which is inadequate for applications such as geometric calibration. For the general Stewart platform considered herein, every chain is essentially a general 6-degrees-of-freedom serial manipulator, with planar or non-planar mounting base and moving platform. By general serial manipulator, each joint can be prismatic, revolute, or helical, with no presumed geometric constraints between joints, such as perpendicularity, parallelism, and intersection. From here onwards, the general Stewart platform, if not otherwise stated, adheres to the above definition. Such generalization allows the control of a wide range of modified Stewart platforms designed to fulfil specific optimization criteria or special task requirements, such as in [35,36], where the modified Stewart platform was designed for better dexterity. In [37], a special Stewart platform with non-intersecting U-joint was considered for precision surgery, which clearly violates Equation (1). Moreover, such generation enables the kinematic calibration of the Stewart platform. Wang et al. [38] showed that the accuracy of the Stewart platform can be of the same level as a serial manipulator with the same nominal dimensions owing to manufacturing tolerances. Later, Masory et al. [39] showed that the accuracy of the Stewart platform can be enhanced by at least one order of magnitude after kinematic calibration, for which purpose, an error model of the Stewart platform, which can be regarded as the general Stewart platform, was established.

In this paper, we intend to develop a numerical algorithm for real-time forward kinematics of the general Stewart platform. For this purpose, a geometric approach is adopted for describing rigid body motion. Different from Denavit–Hartenberg and dual-quaternion parametrization, the geometric approach describes rigid body motion via exponential mapping of twist. The product of exponentials (POE) formula is utilized for the forward kinematics of the general serial manipulator. Using a loop closure condition, a very nice and neat differential relationship between task space and joint space are then established, based on which an iterative algorithm is presented. Simulation studies are then carried out to verify the effectiveness and convergence ability of the proposed algorithm. Finally, the algorithm is implemented on an embedded controller to demonstrate its capability in real applications.

The major contributions of this paper are summarized as follows.

1. A geometric algorithm is proposed for real-time forward kinematics of the general Stewart platform, with no geometric constraints, such as perpendicularity, parallelism, and intersection, are presumed between joints. The proposed method is derived with regard to a general Stewart platform, and can be readily applied to other existing spatial manipulators, including but not limited to 6-RUS and 6-RSS manipulators.
2. The proposed algorithm is successfully implemented on an embedded controller. Compared with the incremental method, which only applies to 6-SPS method, the proposed algorithm is more robust, with a comparative execution time of 0.48 ms, which is sufficient for most real-time applications.

The rest of this paper is organized as follows. In Section 2, an introduction to the mathematical foundation of the proposed geometric method is presented, based on which the details of the proposed forward kinematics algorithm is presented in Section 3. To demonstrate the effectiveness of the proposed algorithm, simulations and experiments are presented in Section 4. Finally, Section 5 concludes this paper.

2. Notation and Terminology

In this section, we present the mathematical tools for modelling a general serial robot, the notation and motivation presented herein basically follow [40].

2.1. Lie Group and Lie Algebra

Mathematically, a Lie group is a group that is also a differentiable manifold. To every Lie group we can associate a Lie algebra whose underlying vector space is the tangent

space of the Lie group at the identity element. An elementary introduction to Lie group and Lie algebra can be found in [41].

The collection of all rotation matrices, known as the special orthogonal group $SO(3)$, is a Lie Group. The corresponding Lie algebra, denoted as $\mathfrak{so}(3)$, is the set of all screw-symmetric 3×3 matrices. $\mathfrak{so}(3)$ and \mathbb{R}^3 are isomorphic via the hat operator $(\cdot)^\wedge : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$, defined as follows:

$$(\cdot)^\wedge : \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \mapsto \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (2)$$

Additionally, the inverse of the hat operator is defined as the vee operator $(\cdot)^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$.

The set of all the rigid body transformations, known as the special Euclidean group $SE(3)$, is also a Lie group. An element $g \in SE(3)$ can be completely described by a 4×4 homogeneous matrix:

$$\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{P} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \quad (3)$$

where $\mathbf{R} \in SO(3)$ denotes the rotation between the two coordinate frames and $\mathbf{P} \in \mathbb{R}^3$ denotes the spatial displacement of the origin. The corresponding Lie algebra $\mathfrak{se}(3)$ is isomorphic to \mathbb{R}^6 , with the hat operator $(\cdot)^\wedge : \mathbb{R}^6 \rightarrow \mathfrak{se}(3)$ defined as

$$(\cdot)^\wedge : \begin{bmatrix} v \\ \omega \end{bmatrix} \mapsto \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}. \quad (4)$$

Likewise, the inverse of the hat operator is defined as the vee operator $(\cdot)^\vee : \mathfrak{se}(3) \mapsto \mathbb{R}^6$.

Given a rigid body motion $g(t) \in SE(3)$, the spatial velocity and body velocity are defined as

$$\mathbf{V}_s = (\dot{\mathbf{g}}\mathbf{g}^{-1})^\wedge \quad (5)$$

$$\mathbf{V}_b = (\mathbf{g}^{-1}\dot{\mathbf{g}})^\wedge. \quad (6)$$

The spatial velocity and body velocity are essentially the same thing expressed in different frames, their relationship can be conveniently expressed using the adjoint map associated with g , $Ad_g(\cdot) : \mathfrak{se}(3) \rightarrow \mathfrak{se}(3)$ defined as

$$Ad_g(\cdot) : \xi \mapsto (g \cdot \hat{\xi} \cdot g^{-1})^\vee. \quad (7)$$

2.2. Exponential Map

For a vector field X in tangent space $T_e G$ at group identity e , there exists a smooth one-parameter subgroup $\gamma_X(t)$ of a Lie group G parametrized by a scalar $t \in \mathbb{R}$. There also exists an exponential map defined as

$$\exp : T_e G \times \mathbb{R} \mapsto G, (X, t) \mapsto \exp(tX). \quad (8)$$

For $SO(3)$ and $SE(3)$, the exponential map corresponds to the usual matrix exponentiation $\exp(\cdot) : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$,

$$\exp(\cdot) : \mathbf{M} \mapsto \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{M}^n. \quad (9)$$

3. Forward Kinematics of the General Stewart Platform

In this section, armed with the tools presented in Section 2, an algorithm is proposed for the forward kinematics of the general Stewart platform.

3.1. Kinematics of General Serial Robot

For a general serial robot with any number of helical joints, as shown in Figure 2, usually a spatial frame S and a tool frame T are attached at the base and the end-effector, respectively. The transformation of tool frame with respect to the spatial frame can be expressed via the POE formula [40]:

$$\mathbf{g}_{st}(\Theta) = e^{\hat{\zeta}_1 \theta_1} \dots e^{\hat{\zeta}_n \theta_n} \mathbf{g}_{st}(\mathbf{0}), \tag{10}$$

where ζ_i and θ_i are the twist and displacement corresponding to the i -th joint, respectively; $\mathbf{g}_{st}(\mathbf{0})$ is the initial rigid displacement between the tool frame and the spatial frame.

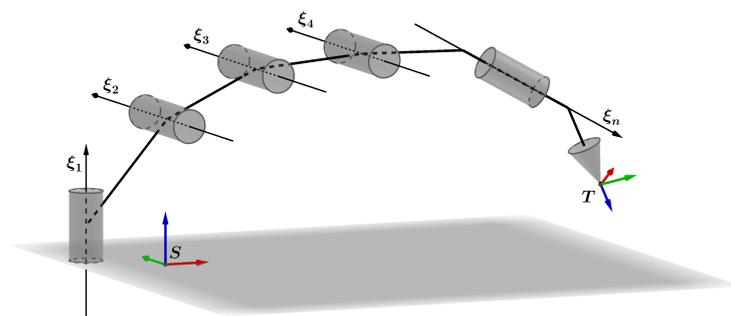


Figure 2. A general serial robot with n joints.

The spatial velocity of the tool frame is defined as

$$\mathbf{V}_{st} = \left(\dot{\mathbf{g}}_{st} \mathbf{g}_{st}^{-1} \right)^\vee = [\zeta'_1 \quad \zeta'_2 \quad \dots \quad \zeta'_n] [\dot{\theta}_1 \quad \dot{\theta}_2 \dots \dot{\theta}_n]^T, \tag{11}$$

where

$$\zeta'_i = \begin{cases} \zeta_1 & i = 1, \\ \prod_{k=1}^{i-1} \text{Ad}_{e^{\hat{\zeta}_k \theta_k}} \zeta_i & \text{otherwise.} \end{cases} \tag{12}$$

The spatial Jacobian matrix is defined as

$$\mathbf{J}_{st} = [\zeta'_1 \quad \zeta'_2 \quad \dots \quad \zeta'_n]. \tag{13}$$

At this point, it is worth mentioning that when $d\Theta$ is small enough,

$$\mathbf{g}_{st}(\Theta + d\Theta) - \mathbf{g}_{st}(\Theta) \approx (\mathbf{J}_{st} d\Theta)^\wedge \mathbf{g}_{st}(\Theta). \tag{14}$$

The relation follows directly from the definition of \mathbf{V}_{st} , and shall be useful in the following derivations.

3.2. The Proposed Algorithm

First of all, consider the i -th chain of the general Stewart platform, denote ζ_i^j and θ_i^j twist and joint displacement of the j -th joint. According to Equation (10), the forward kinematics can be written as:

$$\mathbf{g}_i = \prod_{j=1}^6 e^{\hat{\zeta}_i^j \theta_i^j} \mathbf{g}_i(\mathbf{0}), \tag{15}$$

where $\mathbf{g}_i(\mathbf{0})$ is the initial rigid body displacement. It is shown that the minimal number of parameters needed to determine the initial position of joint twists and rigid body displacements is $5h + 4r + 2t + 6$ [42,43], where h is the number of helical joints, r is the number of revolute joints, t is the number of prismatic joints, and the constant 6 refers to

the number of parameters needed to determine the initial rigid displacement. For instance, the Stewart platform considered later in Section 4 contains 1 helical joint, 4 revolute joints, and 1 prismatic joint in each chain, hence 29 geometric parameters are needed. Subsequently, a set of 174 geometric parameters are needed for all six chains.

At this point, the forward kinematics of the general Stewart platform is equivalent to solving the following equations:

$$\mathbf{g}_i = \mathbf{g}_{st}, \text{ for } i = 1, \dots, 6, \quad (16)$$

where \mathbf{g}_{st} is the 4×4 homogeneous matrix representing the configuration of tool frame at a given input to solve. Excluding the actuator of each chain, there are a total of 30 unknown passive joint displacements, meanwhile there are 6 unknowns in \mathbf{g}_{st} . Hence, a total of 36 unknowns are present. On the other hand, each of the equations provides essentially 6 constraints, so the number of constraints is 36 in general (degenerate in case of singularities). Hence, the system of equations is complete as the number of variables agrees with the number of constraints. To solve the forward kinematics, consider the m -th and the n -th chains of the general Stewart platform, immediately we have

$$\mathbf{g}_m = \mathbf{g}_{st} = \mathbf{g}_n, \quad (17)$$

where \mathbf{g}_m and \mathbf{g}_n denote the forward kinematics of m -th and n -th chains, respectively. Generally, Equation (17) does not hold at initial joint values. Nonetheless, according to Equation (14), if the joint deviations are small enough, the following approximation holds:

$$((\mathbf{J}_m \delta \Theta_m)^\wedge + \mathbf{I}_4) \mathbf{g}_m \approx ((\mathbf{J}_n \delta \Theta_n)^\wedge + \mathbf{I}_4) \mathbf{g}_n, \quad (18)$$

where \mathbf{J}_m and \mathbf{J}_n are the spatial Jacobians corresponding to m -th chain and n -th chain, respectively; $\delta \Theta_m$ and $\delta \Theta_n$ are the joint deviations corresponding to m -th chain and n -th chain, respectively. Multiplying both sides with \mathbf{g}_m^{-1} , the following approximation is obtained with some re-arrangements:

$$(\mathbf{J}_m \Theta_m)^\wedge - (\mathbf{J}_n \Theta_n)^\wedge (\mathbf{g}_n \mathbf{g}_m^{-1}) \approx \mathbf{g}_n \mathbf{g}_m^{-1} - \mathbf{I}_4. \quad (19)$$

In case the joint deviations are small enough, the following approximation can be safely used:

$$(\mathbf{J}_n \Theta_n)^\wedge \mathbf{g}_n \mathbf{g}_m^{-1} \approx (\mathbf{J}_n \Theta_n)^\wedge. \quad (20)$$

Substitution back into Equation (19) yields

$$(\mathbf{J}_m \Theta_m)^\wedge - (\mathbf{J}_n \Theta_n)^\wedge \approx \mathbf{g}_n \mathbf{g}_m^{-1} - \mathbf{I}_4, \quad (21)$$

which up to the first order is equivalent to

$$\mathbf{J}_m \delta \Theta_m - \mathbf{J}_n \delta \Theta_n \approx \log^\vee(\mathbf{g}_n \mathbf{g}_m^{-1}), \quad (22)$$

where $\log : SE(3) \mapsto \mathfrak{se}(3)$ is the inverse operator of $\exp(\cdot)$, and $\log^\vee(\mathbf{g}_n \mathbf{g}_m^{-1})$ abbreviates for $(\log(\mathbf{g}_n \mathbf{g}_m^{-1}))^\vee$. To this point, a differential relationship between joint space and task space for any two chains has been established. To discriminate the active and passive joints, notice that

$$\mathbf{J}_m \delta \Theta_m = \mathbf{J}_{am} \delta \Theta_{am} + \mathbf{J}_{pm} \delta \Theta_{pm}, \quad (23)$$

where Θ_{am} and Θ_{pm} are the active and passive joint displacements of the m -th chain, respectively; \mathbf{J}_{am} and \mathbf{J}_{pm} are the Jacobian matrices corresponding to the active and passive joints of the m -th chain, respectively. In the case of the forward kinematics, the joint

displacements of the actuators are known a priori, namely, $\delta\Theta_{am} = 0$. Hence Equation (22) is equivalent to

$$\mathbf{J}_{pm}\delta\Theta_{pm} - \mathbf{J}_{pn}\delta\Theta_{pn} \approx \log^{\vee}(\mathbf{g}_n\mathbf{g}_m^{-1}). \quad (24)$$

While Equation (24) holds for any two chains, it can be shown that only 5 of which are linearly independent. For a start, it is feasible to fix $m = 1$ and let n iterate over the remaining chains. Alternatively, it is also possible to use the information of adjacent chains, which in matrix form is written as

$$\mathbf{J}_p\delta\Theta_p \approx \delta\mathbf{P}, \quad (25)$$

where

$$\mathbf{J}_p = \begin{bmatrix} \mathbf{J}_{p1} & -\mathbf{J}_{p2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{p2} & -\mathbf{J}_{p3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}_{p3} & -\mathbf{J}_{p4} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{p4} & -\mathbf{J}_{p5} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}_{p5} & -\mathbf{J}_{p6} \end{bmatrix}, \quad (26)$$

$$\delta\Theta_p = [\delta\Theta_{p1}^T \quad \delta\Theta_{p2}^T \quad \delta\Theta_{p3}^T \quad \delta\Theta_{p4}^T \quad \delta\Theta_{p5}^T \quad \delta\Theta_{p6}^T]^T, \quad (27)$$

$$\delta\mathbf{P} = [\log^{\vee}(\mathbf{g}_2\mathbf{g}_1^{-1})^T \quad \log^{\vee}(\mathbf{g}_3\mathbf{g}_2^{-1})^T \quad \log^{\vee}(\mathbf{g}_4\mathbf{g}_3^{-1})^T \quad \log^{\vee}(\mathbf{g}_5\mathbf{g}_4^{-1})^T \quad \log^{\vee}(\mathbf{g}_6\mathbf{g}_5^{-1})^T]^T. \quad (28)$$

Based on Equation (25), an algorithm for solving the forward kinematics of the general Stewart platform is summarized in Algorithm 1. When the Stewart platform is close to singularity, the small deviation in task space amounts to great joint movement, which is undesirable for real applications. To increase the robustness in the case of singularities, the algorithm uses damped least-square method in step 3. The damped least squares method, also known as the Levenberg–Marquardt method, avoids many of the pseudo-inverse method's problems with singularities and can give a numerically stable solution. It was first used for inverse kinematics by Wampler [44] and Nakamura and Hanafusa [45].

Algorithm 1 Forward kinematics of the general Stewart platform.

Input:

- Twist coordinates of all chains, $\zeta_i^j, i, j = 1 \dots 6$;
- Initial rigid displacements of all chains, $\mathbf{g}_i(0), i = 1 \dots 6$;
- Initial joint displacements of all joints for all chains, $\theta_i^j(0), i, j = 1 \dots 6$;
- Absolute tolerance, ϵ ;
- Damping factor, λ .

Output:

- Tool displacement \mathbf{g}_{st} ;
 - 1: Calculate tool frame pose \mathbf{g}_i and spatial Jacobian \mathbf{J}_i of each chain according to Equations (10) and (13);
 - 2: Calculate \mathbf{J}_p and $\delta\mathbf{P}$ according to Equations (26)–(28);
 - 3: Solve Equation (25), $\delta\Theta_p = (\mathbf{J}_p^T\mathbf{J}_p + \lambda^2)^{-1}\delta\mathbf{P}$;
 - 4: Update passive joint values, $\Theta_p = \Theta_p + \delta\Theta_p$;
 - 5: If $\|\delta\Theta_p\| > \epsilon$, go to step 1;
 - 6: **return** \mathbf{g}_{st} as arbitrary $\mathbf{g}_i, i = 1, 2, \dots 6$.
-

4. Simulations and Experiments

In this section, simulations and experiments are conducted to demonstrate the performance of the proposed algorithm in terms of correctness, robustness, convergence ability, and execution time.

4.1. Simulation Setups

Figure 3a shows the CAD model of the general Stewart platform used in the simulation. For each chain shown in Figure 3b, six joints can be identified, where (a) ζ_{i1} and ζ_{i2} are two revolute joints corresponding to the lower U-joint; (b) ζ_{i3} is the linear actuation joint; (c) ζ_{i4} is a screw joint; and (d) ζ_{i5} and ζ_{i6} are two revolute joints corresponding to the upper U-joint. Different from the traditional U-joint, the two shafts of both the lower and the upper U-joints do not intersect with each other.

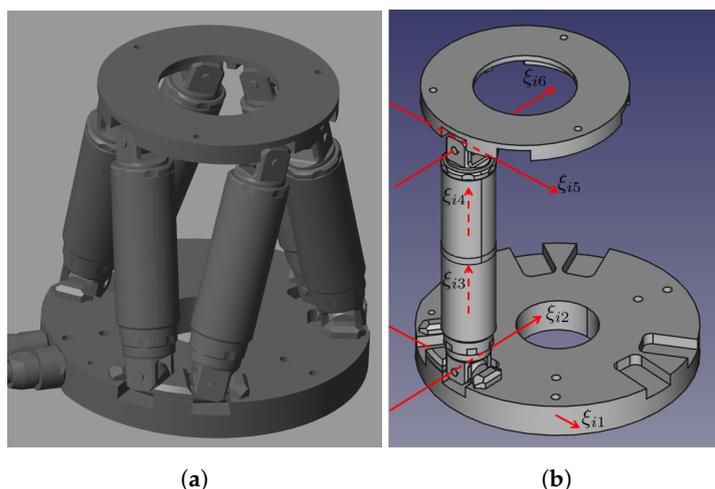


Figure 3. The Stewart platform: (a) the CAD model; (b) twist definitions and initial pose of the i -th chain.

The home position of the i -th chain is also defined in Figure 3b, where (a) ζ_{i2} is parallel to the base plane; (b) ζ_{i3} and ζ_{i4} are perpendicular to ζ_{i1} and ζ_{i2} ; (c) ζ_{i5} is parallel to ζ_{i2} ; and (d) ζ_{i6} is parallel to ζ_{i1} . The moving platform is determined such that it is parallel to the base. As in Figure 4, to describe the initial twist coordinates and rigid displacement of the i -th chain, a set of 11 geometric parameters are needed. These parameters are tabulated in Table 1. The twist coordinates can then be calculated with respect to the chain kinematic parameters. The initial rigid displacement of each chain can be determined in a similar way, as in Table 2.

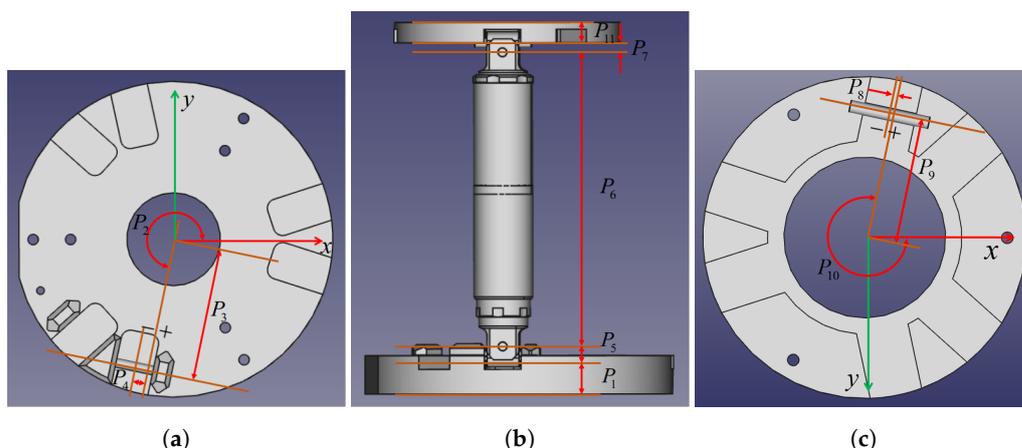


Figure 4. Geometric parameters for one chain: (a) top view of the mounting base; (b) side view of the linear actuator; (c) bottom view of the moving platform.

Table 1. Geometric parameters of all chains.

	P_1 (mm)	P_2 (deg)	P_3 (mm)	P_4 (mm)	P_5 (mm)	P_6 (mm)	P_7 (mm)	P_8 (mm)	P_9 (mm)	P_{10} (deg)	P_{11} (mm)
chain 1	17.00	18.00	57.00	−5.06	4.20	85.00	4.20	0.56	39.00	42.00	9.00
chain 2	17.00	102.00	57.00	5.06	4.20	85.00	4.20	−0.56	39.00	78.00	9.00
chain 3	17.00	138.00	57.00	−5.06	4.20	85.00	4.20	0.56	39.00	162.00	9.00
chain 4	17.00	222.00	57.00	5.06	4.20	85.00	4.20	−0.56	39.00	198.00	9.00
chain 5	17.00	258.00	57.00	−5.06	4.20	85.00	4.20	0.56	39.00	282.00	9.00
chain 6	17.00	342.00	57.00	5.06	4.20	85.00	4.20	−0.56	39.00	318.00	9.00

Table 2. Initial rigid displacement of each chain.

	Translation			Orientation
	x (mm)	y (mm)	z (mm)	(deg)
chain 1	18.51	1.28	119.40	Rotz(−24.00)
chain 2	−8.14	16.67	119.40	Rotz(24.00)
chain 3	−10.37	15.39	119.40	Rotz(−24.00)
chain 4	−10.37	−15.39	119.40	Rotz(24.00)
chain 5	−8.14	−16.67	119.40	Rotz(−24.00)
chain 6	18.51	1.28	119.40	Rotz(24.00)

4.2. Correctness of the Proposed Algorithm

The proposed algorithm naturally solves the displacements of all passive joints, the correctness of which can therefore be verified by comparing the forward kinematics of all six chains. If the results from all chains differ within an acceptable range, the forward kinematics solution along with all the passive joint displacements constitute a valid configuration of the general Stewart platform. As there exists no bi-invariant distance between two elements $\mathbf{g}_0, \mathbf{g}_1$ of SE(3), the spatial distance and rotational distance shall be treated separately. For any two rigid body displacements \mathbf{g}_0 and \mathbf{g}_1 ,

$$\delta\mathbf{g} = \begin{bmatrix} 0 & -\delta z & \delta y & \delta\alpha \\ \delta z & 0 & -\delta x & \delta\beta \\ -\delta y & \delta x & 0 & \delta\gamma \\ 0 & 0 & 0 & 0 \end{bmatrix} = \log(\mathbf{g}_0\mathbf{g}_1^{-1}). \quad (29)$$

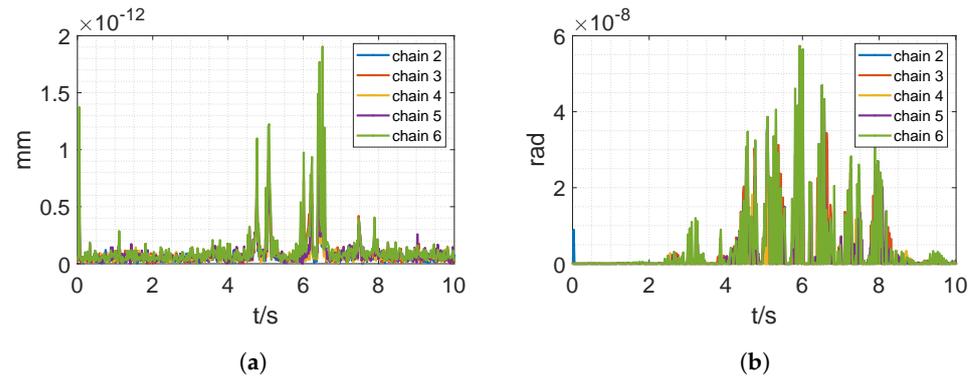
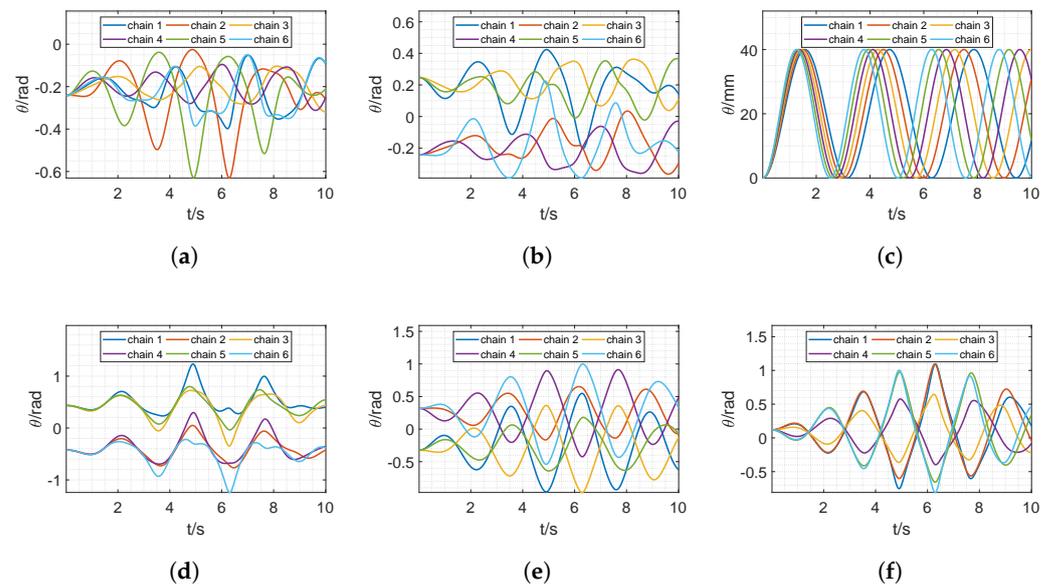
The position error vector is identified as $v = [\delta x \ \delta y \ \delta z]^T$, whose norm will be adopted as the position error. Similarly, the orientation error vector is identified as $\omega = [\delta\alpha \ \delta\beta \ \delta\gamma]^T$, whose norm will be used as the orientation error.

The standard sine wave trajectory is imposed on each actuator, i.e., $L_i = A_i \sin(\omega_i t + \phi_i) + B_i$. To simulate varying poses, different angular velocities are specified for each chain, as in Table 3. The motion profile lasts 10 s, with forward kinematics evaluated every 10 ms. The stopping threshold is set to 10^{-6} , such that the maximum error will be less than 10^{-6} mm or 10^{-6} rad in all directions. Forward kinematics from the first chain is used as the return value of our algorithm, with which the results from every other chain will be compared. The resulting position and orientation errors are shown in Figure 5, where the position errors and the orientation errors are on the order of 10^{-12} mm and 10^{-8} rad, respectively, far less than the specified threshold. Hence the forward kinematic solution is accurate.

If the above solution is feasible for real applications, the trajectories of all joints should also be smooth. Figure 6 shows the trajectories of all joints during the simulation. For better visualization, joints are grouped by their orders in each chain. For example, Figure 6a plots the first joint, which is passive, of each chain, and Figure 6c corresponds to the trajectories of the active joint of each chain. Clearly, all the joint trajectories are smooth, therefore the Stewart platform transits to the final configuration smoothly.

Table 3. Sine trajectory parameters of each chain.

	A_i (mm)	ω_i (rad/s)	ϕ (rad)	B_i (mm)
chain 1	20.00	2.00	0.00	20.00
chain 2	20.00	2.10	0.00	20.00
chain 3	20.00	2.20	0.00	20.00
chain 4	20.00	2.30	0.00	20.00
chain 5	20.00	2.40	0.00	20.00
chain 6	20.00	2.50	0.00	20.00

**Figure 5.** Position and orientation error compared with chain 1; (a) position error; (b) orientation error.**Figure 6.** Joint displacements vs. time for all chains; (a) joint 1; (b) joint 2; (c) joint 3; (d) joint 4; (e) joint 5; (f) joint 6.

4.3. Convergence and Robustness of Proposed Algorithm

To test the convergence ability and robustness of the proposed algorithm, random actuator commands are sent to the Stewart platform at home position. The maximum allowed speed of the linear actuator is around 30 mm/s, which means the joint deviation from last pose is less than 0.3 mm, assuming the cycle time being 10 ms. To test the robustness of proposed algorithm, this deviation is enlarged by a factor of 10. Randomly generated deviations uniformly distributed over $[-3, 3]$ mm are set to all the actuators for one million times. The threshold for termination is 10^{-6} , the same as that in the previous simulation.

In all test cases, the algorithm converges with maximum iteration being 4. To visualize the convergence speed, Figure 7 shows the errors at each iteration for 100 randomly selected cases. As can be seen from the figure, the error magnitude almost squared after each iteration and terminated with a maximum of 4 iterations. Therefore, the proposed algorithm possesses a near-quadratic convergence speed, which is not surprising for Levenberg–Marquardt algorithms.

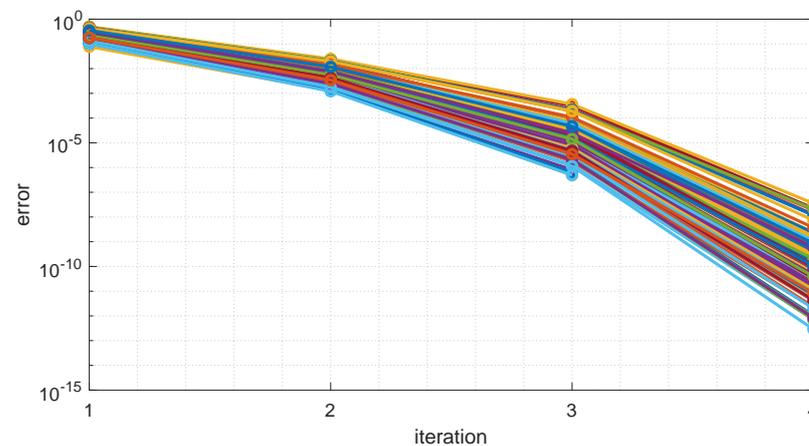


Figure 7. Error vs. iteration of proposed algorithm of 100 randomly selected cases.

4.4. On-Board Implementation for Real-Time Control

To test the execution time in real applications, the proposed algorithm is implemented on a motion controller platform (iDEABOX 3 Pro intelligent controller), equipped with an Intel Bay Trail J1900 processor at 2.0 GHz. The cycle time of the controller is set to 10 ms. Figure 8 shows the hardware setup for our experiment. The same joint trajectory as in Section 4.2 is commanded to the controller, with termination threshold ϵ set to 10^{-6} and damping factor λ set to 10^{-8} . The incremental method proposed in Wang [24] is implemented here as a comparison, using the same termination threshold. The maximum iteration counts for both algorithms are set to 20. There are some clarification to be made clear

1. The testing Stewart platform violates Equation (1), as it has non-intersecting upper and lower universal joints. Therefore the incremental method can't be used. However, since P_5 and P_7 in Table 1 are quite small, in our implementation we manually set them to 0 such that the incremental method become feasible. Although such modification changes the kinematic model and is not applicable in practise, the results should still provide insights regarding the robustness and speed of the proposed algorithm.
2. To address the initial value problem, the Stewart platform is set to start at home pose for the first control cycle, namely $\Theta = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$. Accordingly, the following initial end-effector displacement is used

$$g(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 114.75 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (30)$$

From the second cycle and onwards, the last cycle pose is used as the initial value.

Figure 9 plots the execution times and iteration counts of both algorithms, and Table 4 summarizes some key statistics. Since the incremental method doesn't always converge, the statistics of convergent cases are also listed.

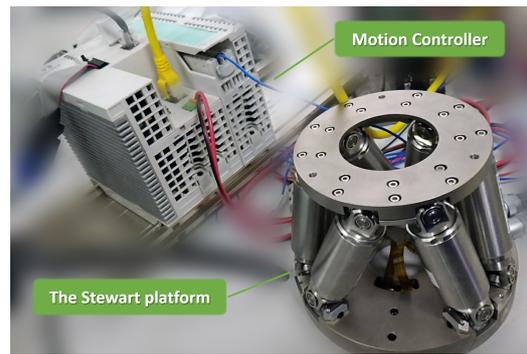


Figure 8. The experiment set-up.

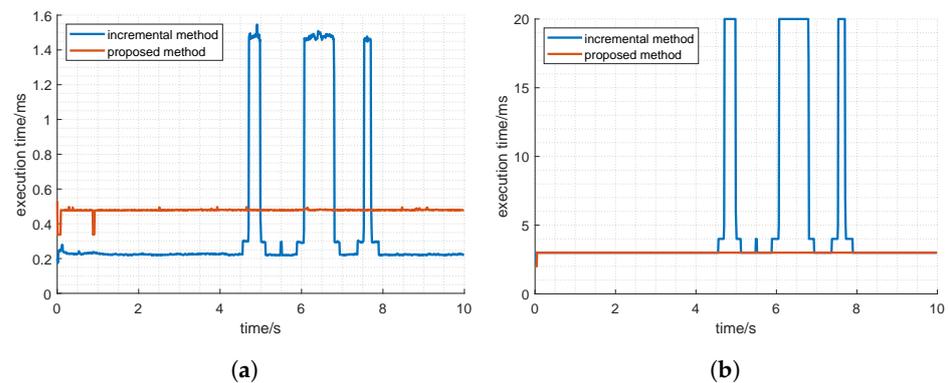


Figure 9. Comparison of iteration count and execution time with the incremental method; (a) execution time; (b) iteration count.

Table 4. Statistics of forward kinematics of two methods.

	Average Iteration Count	Max Iteration Count	Average Execution Time	Max Execution Time
incremental method (all cases)	5.12	20	0.38	1.54
incremental method (convergent cases)	3.13	17	0.23	1.25
proposed method	2.99	3	0.48	0.53

From the results we can see that compared with the incremental method, which only applies to the 6-SPS model, the proposed algorithm is more robust, with a comparative execution time of 0.48 ms. Specifically,

1. The proposed algorithm converges at all command points, while the incremental method fails in multiple cases. Further experiment shows that the incremental method converges in all points if the magnitude of all joint trajectories decreases to 15 mm. In other words, the proposed algorithm has a larger convergent area than the incremental method, and thus is more robust. One reason that might account for the difference is that the incremental method is a Newton–Raphson method, and the proposed method belongs to Levenberg–Marquardt, which is generally considered more robust.
2. In cases where both algorithms converge, average iteration counts are 3.13 and 2.99, respectively, indicating that the two algorithms share basically the same convergence ability. This can be explained as the Levenberg–Marquardt degenerating to a Newton–Raphson when using a very small damping factor.

3. In case the incremental method converges, the execution time is around half of the proposed method's. Since the iteration counts for two algorithms are almost identical, we conclude that the complexity of the iteration equation leads to the difference. Indeed, although J_p is quite sparse, it is a 30-by-30 matrix, whereas the counterpart in the incremental method is 6-by-6. To this point, we would like to mention that the extra complexity is not for nothing. For one thing, the incremental method is constructed for the ideal 6-SPS platform, so it can't be applied directly to other variants such as the 6-RSS. Moreover, subject to manufacturing ability, a calibration process is usually needed to increase the accuracy. For this purpose, an error model is indispensable, which might break Equation (1) by considering imperfect ball joints. The incremental method is infeasible in both cases. Whereas the proposed method can still be adopted by simply modifying the corresponding twist coordinates.

5. Conclusions

In this paper, a geometric algorithm was proposed for real-time forward kinematics of the general Stewart platform, where no geometric constraints, such as perpendicularity, parallelism, and intersection, are presumed between joints. Specifically, a geometric approach, which describes the rigid body motion as exponential mapping of twist was used to formulate the problem. A very neat and simple differential relationship between task space and joint space was therefore established. The proposed algorithm is derived for the general Stewart platform, and can work for other existing spatial manipulators, including but not limited to 6-RUS and 6-RSS. Experiments and simulations were carried out to demonstrate the effectiveness of the proposed algorithm. The results showed that the proposed algorithm is accurate, robust, and fast. In all test cases, the proposed algorithm terminates with maximum iteration being 4 and converges with near-quadratic speed. The proposed algorithm is also implemented on an embedded controller. Compared with the incremental method, which only applies to the 6-SPS model, the proposed algorithm is more robust, with a comparative execution time of 0.48 ms. Overall, the efficiency of proposed algorithm meets the requirements of most applications, such as kinematic calibration, motion simulation, and real-time control. Some of the future work are summarized as follows.

1. As the damping factor λ has great influence on the efficiency of the Levenberg–Marquardt method, instead of using a fixed damping factor, a more effective strategy can be explored considering manipulability and other factors.
2. Calibration can greatly increase the accuracy of the end-effector. Most existing algorithms considered the errors from actuator lengths, ball joint location, and motion errors; developing a calibration model that applies to the general Stewart platform is necessary.

Author Contributions: F.Y.: conceptualization, investigation, methodology, writing—original draft, writing—review & editing; X.T.: conceptualization, methodology, writing—review & editing; Z.W.: investigation, software, validation; Z.L.: data interpretation, writing—review & editing; T.H.: methodology, writing—review & editing, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0909030005, Guangdong Basic and Applied Basic Research Foundation (2021A1515110354, 2021A1515110576), and the Fundamental Research Funds for the Central Universities, Sun Yat-sen University (22qntd1704, 22qntd1708).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stewart, D. A platform with six degrees of freedom. *Proc. Inst. Mech. Eng.* **1965**, *180*, 371–386. [\[CrossRef\]](#)
2. Dasgupta, B.; Mruthyunjaya, T. The Stewart platform manipulator: A review. *Mech. Mach. Theory* **2000**, *35*, 15–40. [\[CrossRef\]](#)
3. Griffis, M.; Duffy, J. A forward displacement analysis of a class of Stewart platforms. *J. Robot. Syst.* **1989**, *6*, 703–720. [\[CrossRef\]](#)
4. Lin, W.; Griffis, M.; Duffy, J. Forward displacement analyses of the 4–4 Stewart platforms. *J. Mech. Des.* **1992**, *114*, 444–450. [\[CrossRef\]](#)
5. Lin, W.; Crane, C.; Duffy, J. Closed-form forward displacement analysis of the 4–5 in-parallel platforms. *J. Mech. Des.* **1994**, *116*, 47–53. [\[CrossRef\]](#)
6. Innocenti, C.; Parenti-Castelli, V. Direct Kinematics in Analytical Form of a General Geometry 5–4 Fully-Parallel Manipulator. In *Computational Kinematics*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 141–152.
7. Innocenti, C.; Parenti-Castelli, V. Direct kinematics of the 6–4 fully parallel manipulator with position and orientation uncoupled. In *Robotic Systems*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 3–10.
8. Wen, F.; Liang, C. Displacement analysis of the 6-6 Stewart platform mechanisms. *Mech. Mach. Theory* **1994**, *29*, 547–557. [\[CrossRef\]](#)
9. Sreenivasan, S.; Waldron, K.; Nanua, P. Closed-form direct displacement analysis of a 6-6 Stewart platform. *Mech. Mach. Theory* **1994**, *29*, 855–864. [\[CrossRef\]](#)
10. Mruthyunjaya, B.D.T. A canonical formulation of the direct position kinematics problem for a general 6-6 Stewart platform. *Mech. Mach. Theory* **1994**, *29*, 819–827.
11. Faugère, J.; Lazard, D. Combinatorial classes of parallel manipulators. *Mech. Mach. Theory* **1995**, *30*, 765–776. [\[CrossRef\]](#)
12. Wampler, C. Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mech. Mach. Theory* **1996**, *31*, 331–337. [\[CrossRef\]](#)
13. Nag, A.; Safar, V.; Bandyopadhyay, S. A uniform geometric-algebraic framework for the forward kinematic analysis of 6-6 Stewart platform manipulators of various architectures and other related 6-6 spatial manipulators. *Mech. Mach. Theory* **2021**, *155*, 104090. [\[CrossRef\]](#)
14. Innocenti, C. Forward kinematics in polynomial form of the general Stewart platform. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, New York, NY, USA, 13–16 September 1998; Volume 80302, p. V01AT01A013.
15. Huang, X.; Liao, Q.; Wei, S. Closed-form forward kinematics for a symmetrical 6-6 Stewart platform using algebraic elimination. *Mech. Mach. Theory* **2010**, *45*, 327–334. [\[CrossRef\]](#)
16. Zhou, W.; Chen, W.; Liu, H.; Li, X. A new forward kinematic algorithm for a general Stewart platform. *Mech. Mach. Theory* **2015**, *87*, 177–190. [\[CrossRef\]](#)
17. Merlet, J.P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int. J. Robot. Res.* **2004**, *23*, 221–235. [\[CrossRef\]](#)
18. Raghavan, M. The Stewart platform of general geometry has 40 configurations. *AMSE J. Mech. Des.* **1993**, *115*, 277–282. [\[CrossRef\]](#)
19. Merlet, J.P. Direct kinematics of parallel manipulators. *IEEE Trans. Robot. Autom.* **1993**, *9*, 842–846. [\[CrossRef\]](#)
20. Yang, C.F.; Zheng, S.T.; Jin, J.; Zhu, S.B.; Han, J.W. Forward kinematics analysis of parallel manipulator using modified global Newton-Raphson method. *J. Cent. South Univ. Technol.* **2010**, *17*, 1264–1270. [\[CrossRef\]](#)
21. Puglisi, L.J.; Saltaren, R.; Garcia, C.; Cardenas, P.; Moreno, H. Implementation of a generic constraint function to solve the direct kinematics of parallel manipulators using Newton-Raphson approach. *J. Control. Eng. Appl. Inf.* **2017**, *19*, 71–79.
22. Xie, B.; Dai, S.; Liu, F. A Lie Group-Based Iterative Algorithm Framework for Numerically Solving Forward Kinematics of Gough–Stewart Platform. *Mathematics* **2021**, *9*, 757. [\[CrossRef\]](#)
23. Yang, C.; He, J.; Han, J.; Liu, X. Real-time state estimation for spatial six-degree-of-freedom linearly actuated parallel robots. *Mechatronics* **2009**, *19*, 1026–1033. [\[CrossRef\]](#)
24. Wang, Y. A direct numerical solution to forward kinematics of general Stewart–Gough platforms. *Robotica* **2007**, *25*, 121–128. [\[CrossRef\]](#)
25. Parikh, P.J.; Lam, S.S. A hybrid strategy to solve the forward kinematics problem in parallel manipulators. *IEEE Trans. Robot.* **2005**, *21*, 18–25. [\[CrossRef\]](#)
26. Bonev, I.A.; Ryu, J. A new method for solving the direct kinematics of general 6-6 Stewart platforms using three linear extra sensors. *Mech. Mach. Theory* **2000**, *35*, 423–436. [\[CrossRef\]](#)
27. Chiu, Y.J.; Perng, M.H. Forward kinematics of a general fully parallel manipulator with auxiliary sensors. *Int. J. Robot. Res.* **2001**, *20*, 401–414. [\[CrossRef\]](#)
28. Cheng, S.; Liu, Y.; Wu, H. A new approach for the forward kinematics of nearly general Stewart platform with an extra sensor. *J. Adv. Mech. Des. Syst. Manuf.* **2017**, *11*, JAMDMS0032. [\[CrossRef\]](#)
29. Morell, A.; Acosta, L.; Toledo, J. An artificial intelligence approach to forward kinematics of Stewart platforms. In Proceedings of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, 3–6 July 2012; pp. 433–438.
30. Haykin, S.; Network, N. A comprehensive foundation. *Neural Netw.* **2004**, *2*, 41.
31. Parikh, P.J.; Lam, S.S. Solving the forward kinematics problem in parallel manipulators using an iterative artificial neural network strategy. *Int. J. Adv. Manuf. Technol.* **2009**, *40*, 595–606. [\[CrossRef\]](#)

32. Chauhan, D.K.S.; Vundavilli, P.R. Forward Kinematics of the Stewart Parallel Manipulator Using Machine Learning. *Int. J. Comput. Methods* **2022**, *214*, 2009. [[CrossRef](#)]
33. Joumah, A.A.; Albitar, C. Design optimization of 6-RUS parallel manipulator using hybrid algorithm. *Int. J. Inf. Technol. Comput. Sci. (IJITCS)* **2018**, *10*, 83–95. [[CrossRef](#)]
34. Lau, T.; Zhang, Y. New design of 6-RSS parallel manipulator with large rotation workspace. *Int. J. Mech. Robot. Syst.* **2013**, *1*, 221–239. [[CrossRef](#)]
35. Stoughton, R.; Arai, T. A modified Stewart platform manipulator with improved dexterity. *IEEE Trans. Robot. Autom.* **1993**, *9*, 166–173. [[CrossRef](#)]
36. Pittens, K.; Podhorodeski, R. A family of Stewart platforms with optimal dexterity. *J. Robot. Syst.* **1993**, *10*, 463–479. [[CrossRef](#)]
37. Wapler, M.; Urban, V.; Weisener, T.; Stallkamp, J.; Dürr, M.; Hiller, A. A Stewart platform for precision surgery. *Trans. Inst. Meas. Control.* **2003**, *25*, 329–334. [[CrossRef](#)]
38. Wang, J.; Masory, O. On the accuracy of a Stewart platform. I. The effect of manufacturing tolerances. In Proceedings of the IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; pp. 114–120.
39. Masory, O.; Wang, J.; Zhuang, H. Kinematic modeling and calibration of a Stewart platform. *Adv. Robot.* **1996**, *11*, 519–539. [[CrossRef](#)]
40. Murray, R. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 2017.
41. Hall, B. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*; Springer: New York, NY, USA, 2015; Volume 222.
42. Yang, X.; Wu, L.; Li, J.; Chen, K. A minimal kinematic model for serial robot calibration using POE formula. *Robot.-Comput.-Integr. Manuf.* **2014**, *30*, 326–334. [[CrossRef](#)]
43. Li, C.; Wu, Y.; Li, Z. Identifiability and improvement of adjoint error approach for serial robot calibration. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 31 May–5 June 2014; pp. 1361–1366.
44. Wampler, C.W. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 93–101. [[CrossRef](#)]
45. Nakamura, Y.; Hanafusa, H. Inverse kinematic solutions with singularity robustness for robot manipulator control. *J. Dyn. Syst. Meas. Control.* **1986**, *109*, 163–171. [[CrossRef](#)]