

Article

Study on the Kinematic Performances and Optimization for Three Types of Parallel Manipulators

Dan Zhang * and Bin Wei

Department of Mechanical Engineering, York University, Toronto, ON M3J 1P3, Canada; binwei28@yorku.ca

* Correspondence: dzhang99@yorku.ca; Tel.: +1-416-736-2100 (ext. 44049)

Academic Editor: Steven Y. Liang

Received: 20 October 2016; Accepted: 13 December 2016; Published: 16 December 2016

Abstract: The modelling, optimization issues and stiffness for several types of three degrees-of-freedom parallel robotic manipulators, i.e., 3-DOF pure translational, 3-DOF pure rotational and 3-DOF mixed motion types, are studied in this paper. First of all, the kinematics and Jacobian for the robotic manipulators are determined through different approaches; secondly, objective functions modelling are presented, and the associated optimization issues and the geometric parameters' effect on the objective functions for the robotic mechanisms are illustrated and analyzed in detail. Through employing several multi-objective optimization approaches, we manifest an overall process and approach for multi-objective optimization of robotic systems. The correlation among different stiffness models is finally presented. The results indicate that the kinetostatic compliance model is the closest one to the traditional stiffness model.

Keywords: kinematics; modelling; optimization; pure translation; pure rotation; mixed motion; parallel mechanism

1. Introduction

Parallel robotic mechanisms have been broadly employed in the healthcare area [1], agricultural area [2,3], manufacturing area [4–6], sensor applications [7,8], etc. The Steward mechanism, one would say, is one of the most popular parallel robotic mechanisms. The upper moving platform of the Steward mechanism is joined to the base through six actuated limbs, and it possesses three translational degrees-of-freedom and three rotational degrees-of-freedom. The function of the full degrees-of-freedom for a parallel mechanism is not necessary for most applications; rather, three degrees-of-freedom, four degrees-of-freedom or even five degrees-of-freedom mechanisms are more preferable. In addition, full degrees-of-freedom parallel mechanisms have a few drawbacks, i.e., their forward kinematic analysis is usually very inconvenient to solve, and their workspace is very limited. The stiffness performance, in a large number of cases, is one of the critical factors that needs to be taken into consideration in the parallel robotic arena, because stiffness demonstrates how rigid a parallel mechanism can be and sometimes can also represent the general accuracy performance of a robotic mechanism. Stiffness is usually employed to measure how much a mechanism can resist under certain loads, and it is seen by many scholars and engineers as one of the most important attributes for robotic mechanisms, as a large value of stiffness will usually result in good precision when robotic mechanisms are employed as machine tools to manufacture pieces. One usually employs a stiffness matrix to represent the stiffness of a parallel robotic mechanism. Additionally, when a parallel mechanism possesses multiple types of design variables or a certain objective function for a parallel mechanism is complicated, the global stiffness design and its optimization in parallel robotic mechanisms will become hard. The above issue can be effectively addressed by resorting

to the optimization approach that is based on the kinetostatic analysis. The authors will analyze three types of three degrees-of-freedom parallel robotic mechanisms, i.e., a three degrees-of-freedom translational robotic mechanism, a three degrees-of-freedom rotational robotic mechanism and a mixed three degrees-of-freedom (one translational degree-of-freedom and two rotational degrees-of-freedom) robotic mechanism. The authors here will also study the optimization problem of each parallel robotic mechanism, and lastly, we will present and compare a few stiffness/compliance models.

This paper is organized as follows: the kinematic and Jacobian analysis and modelling for the three robotic mechanisms are presented in Section 2; Section 3 establishes the related objective functions; the optimization issues for the robotic mechanisms are conducted in Section 4; Sections 5 and 6 illustrate existing popular stiffness models and compare the compliance among the models, respectively; finally, the conclusion is given in Section 7.

2. Kinematic Analysis and Modelling

2.1. Pure Translational Case

As illustrated in Figure 1, the top moving platform of the 3UPU (U represents universal joint, and P represents prismatic joint) robotic mechanism is joined to the base through three identical UPU type limbs. A coordinate frame $O(X, Y, Z)$ is fixed to the point O , which is the base center, and another coordinate frame $P(x, y, z)$ is fixed to the point P , which is the moving platform center. α_b represents the angle from the X axis of the base frame to OB_1 , and α_p presents the angle from x axis of the moving frame to OP_1 .

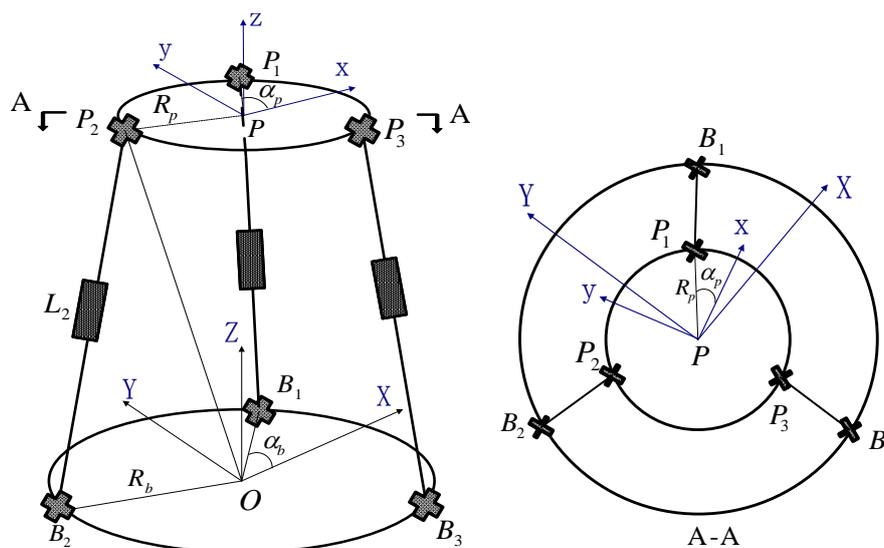


Figure 1. 3UPU robotic manipulator.

The limb length can be determined as follows:

$$L_i = |p_i + q - b_i| \tag{1}$$

where b_i represents the coordinate of point B_i relative to the base frame, $q = [x, y, z]^T$ represents the vector of point P relative to the base frame and p_i represents the coordinate of point P_i relative to the moving frame. The point P_i 's velocity can be determined as follows:

$$v_{P_i} = \omega_i \times s_i + \dot{L}_i s_i \tag{2}$$

where s_i denotes the unit vector along the i -th limb and ω_i denotes the angular velocity for the i -th limb relative to the frame O . If one multiplies s_i on both sides of Equation (2), the following can be derived:

$$Jv_P = \dot{q} \tag{3}$$

where v_P denotes the point P 's velocity, and the Jacobian matrix is determined as follows:

$$J = \left[s_1^T \quad s_2^T \quad s_3^T \right]^T \tag{4}$$

2.2. Pure Rotational Case

Figure 2 illustrates the 3SPS-S (S represents spherical joint, and P represents prismatic joint) parallel mechanism. The frames $P(x, y, z)$ and $B(X, Y, Z)$ are fixed to the point W . q_i represents a unit vector along $\overline{B_iP_i}$. L_i represents the vector of the length for prismatic actuators, and their lengths are represented by l_i . The distance from point W to the base is denoted as h_B , and the distance from point W to the top platform is denoted as h_p . The radius of the base is denoted as r_B , and the radius of the top platform is denoted as r_p .

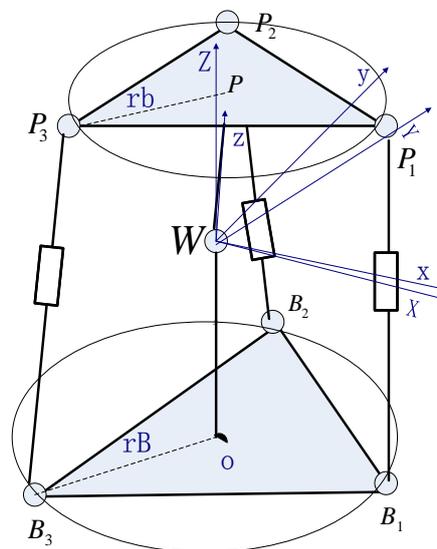


Figure 2. 3SPS-S robotic manipulator.

$$l_i^2 = L_i L_i = (l_i q_i) (l_i q_i) = (Q^P P_i - B_i) (Q^P P_i - B_i) \tag{5}$$

where ${}^P P_i$ represents the position vector of points P_i relative to the moving coordinate system and B_i represents the position vector of points B_i ($i = 1, 2, 3$) relative to the fixed coordinate system B . By taking the time derivative of the above formulation, the following can be derived:

$$\dot{l}_i = \frac{L_i}{l_i} v_{pi} = q_i v_{pi} \tag{6}$$

The points' P_i velocity can be determined as follows:

$$v_{pi} = v + \omega \times b_i = \omega \times b_i \tag{7}$$

If one substitutes Equation (7) to (6), the Jacobian matrix can be obtained as follows:

$$J = (b_i \times q_i)^T \tag{8}$$

where $b_i = Q^P P_i$, and $q_i = L_i/l_i$ ($i = 1, 2, 3$).

2.3. Mixed Motion Case

Figure 3 shows the 4UPS-PU (U represents universal joint, P represents prismatic joint, and S represents spherical joint) parallel mechanism. It contains four identical limbs and one passive limb. The purpose of employing the passive limb is to make the robotic mechanism only have three DOFs. A frame (XYZ) is fixed to the base center, and another frame (xyz) is fixed to the moving platform center. The angle from the x axis of the moving frame to a_i is denoted as ϕ'_i .

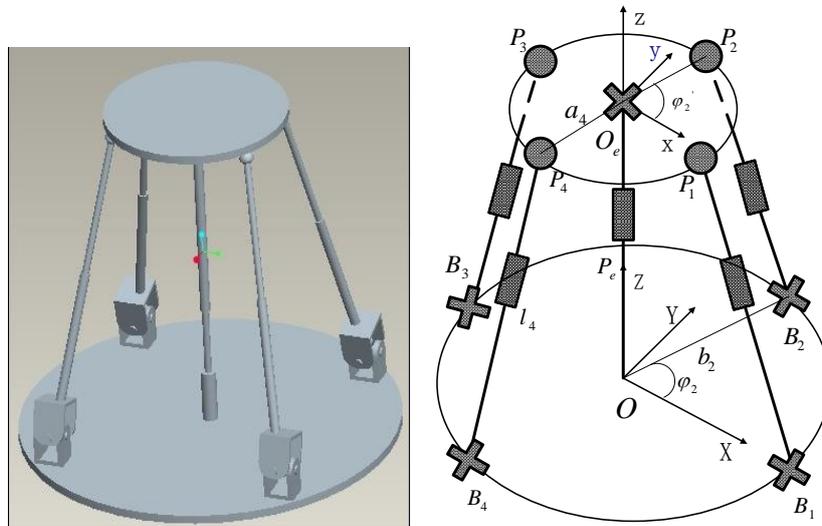


Figure 3. 4UPS-PU robotic manipulator.

Firstly, l_i can be determined as follows:

$$l_i = p_e + a_i - b_i \tag{9}$$

where $a_i = R a'_i$, $b_i = [r \cos \varphi_i \quad r \sin \varphi_i \quad 0]^T$, $a'_i = [c \cos \varphi'_i \quad c \sin \varphi'_i \quad 0]^T$, the radius of the base is denoted as r and the radius of the moving platform is denoted as c .

The i -th limb length can be determined as follows:

$$q_i^2 = l_i^T l_i \tag{10}$$

If one differentiates Equation (10) with respect to time, the Jacobian can be determined as follows:

$$J = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix}^{-1} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \tag{11}$$

where $a_{i1} = -rc \sin \varphi_i \cos \varphi_i \cos \varphi \sin \psi + rc \sin \varphi \sin^2 \varphi_i + hc \cos \varphi_i \sin \varphi \sin \psi + hc \sin \varphi_i \cos \varphi$, $a_{i2} = rc \cos^2 \varphi_i \sin \psi - rc \sin \varphi_i \cos \varphi_i \sin \varphi \cos \psi - hc \cos \varphi_i \cos \varphi \cos \psi$ and $a_{i3} = h - c \cos \varphi_i \sin \psi \cos \varphi + c \sin \varphi_i \sin \varphi$.

3. Optimization Discussion and Modelling

3.1. Pure Translational Case

It is known that the stiffness matrix can be obtained as follows:

$$K = J^T K_J J = k J^T J \tag{12}$$

Every single component on the leading diagonal of the stiffness matrix can reflect each moving directional stiffness. When a robotic mechanism moves, these diagonal components will also change. If one employs these diagonal components to conduct the optimization, one can just obtain the optimum stiffness value at a single spot. Thus, a factor that can reflect the average stiffness [9] is utilized:

$$GSI_i = \frac{\int \int \int K_i dx dy dz}{\int \int \int dx dy dz} \quad (i = x, y, z) \tag{13}$$

As a case study, here, the authors set $\alpha_b = 30^\circ$, $\alpha_p = 15^\circ$; the distributions of the X directional stiffness and Z directional stiffness with respect to the radius of the base and radius of the top platform are illustrated in Figure 4. Note that the distributions for X and Y are the same, so here, we only plotted one case.

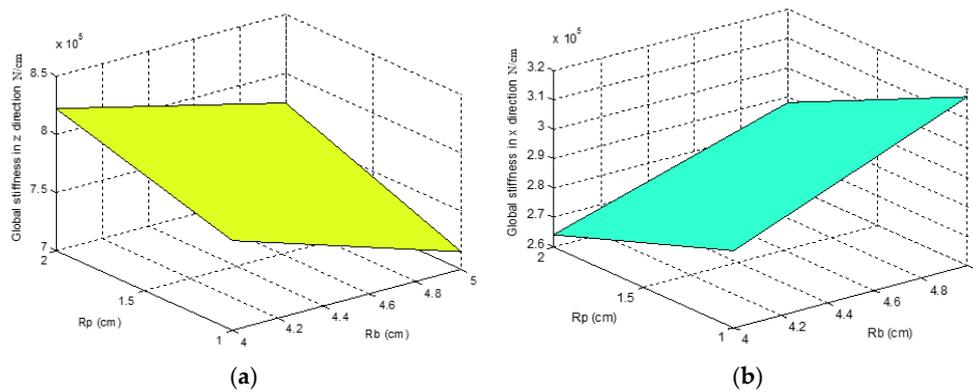


Figure 4. Stiffness distribution in: (a) Z; (b) X.

Similarly, for a different case, the authors set $R_b = 3$ cm, $R_p = 2$ cm, the distributions of X directional stiffness and Z directional stiffness with respect to the attachment points angles are illustrated in Figure 5. Note that the distributions for X and Y are the same, so here, we only plotted one case.

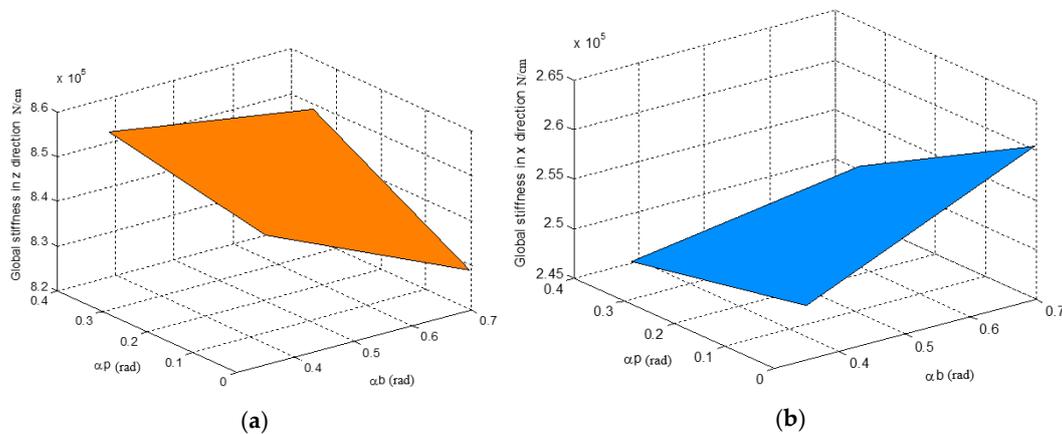


Figure 5. Stiffness distribution in: (a) Z; (b) X.

It is observed that the stiffness depends on the radius of the base, the radius of the top platform and the angles of the attachment points on the base and top platform. It is also observed that the summation of these three stiffnesses is constant. Thus, if one employs the summation of these three stiffnesses to conduct the optimization, it will be unsuitable. A scholar suggested that the Z directional stiffness is able to reflect a robotic mechanism's stiffness [10]. Here, the authors will test if the Z directional stiffness is an objective function and see the outcome.

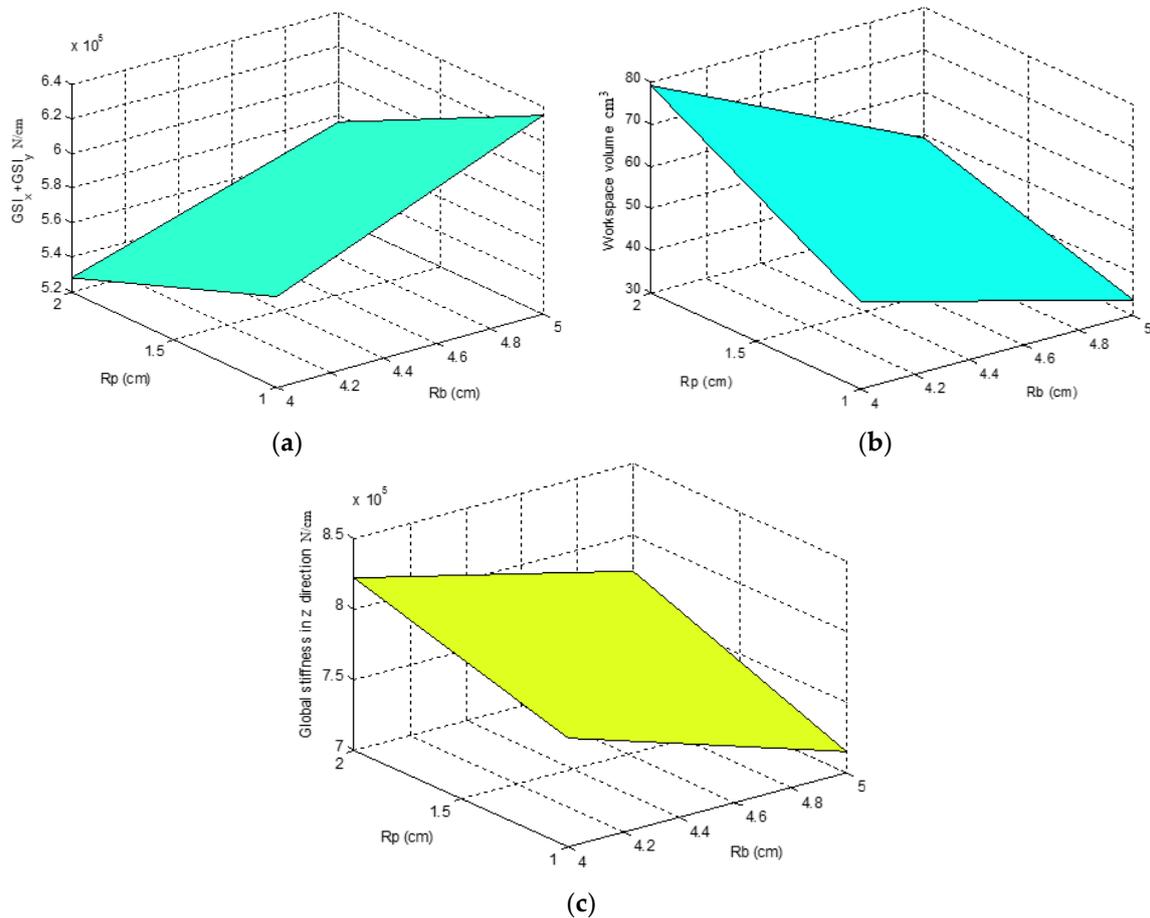


Figure 6. (a) X and Y directional stiffness; (b) workspace volume; (c) Z directional stiffness.

It is observed from Figure 6 that the distributions of the workspace volume have the same changing fashion with respect to the Z directional stiffness case, whereas it has the opposite changing fashion with respect to the X directional stiffness case and Y directional stiffness case. Additionally, the summation of the three stiffnesses is unchanging, and also, the distributions of the X and Y directional stiffnesses have the opposite changing fashion with respect to the Z directional stiffness. Optimizing the Z directional stiffness will affect the X and Y directional stiffnesses. Hence, in practical applications, one needs to select the appropriate objective function based on one's own needs. In the scenario where the Z directional stiffness is vital, one then optimizes the Z directional stiffness. In the scenario where the X and Y directional stiffnesses are vital, one then uses the X and Y directional stiffnesses as an objective function. Here, for a case study, the authors assume that the X and Y directional stiffnesses are vital. The X and Y directional stiffnesses are optimized together with the workspace performance. One can employ the fast search approach [11] to determine the workspace volume.

3.2. Pure Rotational Case

In the compliance matrix, the leading diagonal segments' mean value can reflect the overall compliance. The leading diagonal segments' standard deviation can reflect the compliance variation. A small mean value means that a robotic mechanism has better stiffness performance, and a small standard deviation value means the compliance is evenly distributed. Here, the mean value and standard deviation are utilized for conducting the stiffness optimization.

By resorting to the virtual work principle, the Cartesian compliance matrix of the robotic manipulator is able to be obtained:

$$C_c = J_{serial}(AJ_{serial})^{-1}BCB^T(AJ_{serial})^{-T}J_{serial}^T \quad (14)$$

where $mean = \frac{\sum_{i=1}^6 C_c(i,i)}{6}$ represents the mean value and $std = \sqrt{\frac{\sum_{i=1}^6 (C_c(i,i) - mean)^2}{6}}$ represents the standard deviation.

There are numerous methodologies being put forward for workspace optimization. However, it was noticed that parallel robotic manipulators that are geared to the largest workspace can sometimes produce unwanted kinematic performances (e.g., bad dexterity). Thus, one scholar suggested to employ a global condition index [12] to handle the mentioned issue. The authors will resort to this index as formulated in Equation (15) to represent the performance of the manipulator's workspace.

$$\eta = \int_W \frac{1}{k} dW \quad (15)$$

where k represents the Jacobian's condition number. Determining the above Equation (15) can be difficult, which compels the authors to employ a numerical solution approach (e.g., the Monte Carlo methodology). The approach can be summarized as four major steps: first of all, we select a tremendous amount of points n_{total} inside the predicted workspace region; secondly, these selected points need to be checked to see whether they do indeed fall within the workspace boundary. By computing every single limb length, one is able to determine the above condition. If the limb length complies with its maximum and minimum length, it means the condition is satisfied. After that, one needs to calculate the kinematics condition index. This index times the predicted workspace volume and this product then divided by n_{total} , one finally has the following:

$$\eta = 2\pi(\sqrt{l_2^2 + R_p^2})^3 \cdot \sum_i \frac{1}{k_i} / 3n_{total}$$

3.3. Mixed Motion Case

Based on the virtual work principle, the manipulator's Cartesian stiffness matrix can be derived as follows:

$$K = A^T B^{-T} K_J B^{-1} A + (J_{serial}')^{-T} K_{n+1} (J_{serial}')^{-1} \quad (16)$$

The compliance matrix can therefore be obtained as follows:

$$C = K^{-1} \quad (17)$$

Thus:

$$M = \frac{\sum_{i=1}^6 C_c(i,i)}{6}, S = \sqrt{\frac{\sum_{i=1}^6 (C_c(i,i) - mean)^2}{6}} \quad (18)$$

where M represents the mean value and S represents the standard deviation.

In terms of the workspace, similarly, the authors here will employ the global condition index as described in Section 3.2. Regarding the mixed motion mechanism, first of all, we select a tremendous amount of points n_{total} inside the predicted workspace region; second, by computing every single limb length, one is able to determine if the limb length complies with its maximum and minimum length. After that, one needs to calculate the kinematics condition index. This index times the predicted workspace volume and this product then divided by n_{total} , one finally has the following:

$$\eta = \frac{\pi \cdot R_p^2 \cdot (h_{\max} - h_{\min}) \cdot KCI}{n_{total}}$$

4. Optimization Results

4.1. Pure Translational Case

One regards particle swarm optimization as one of the computational intelligent methodologies [13]. Figure 7 illustrates the general working principle of particle swarm optimization. The authors here use the workspace volume, which is denoted as V for its optimization, and the summation of X and Y directional stiffnesses, which is denoted as f , to reflect stiffness performance.

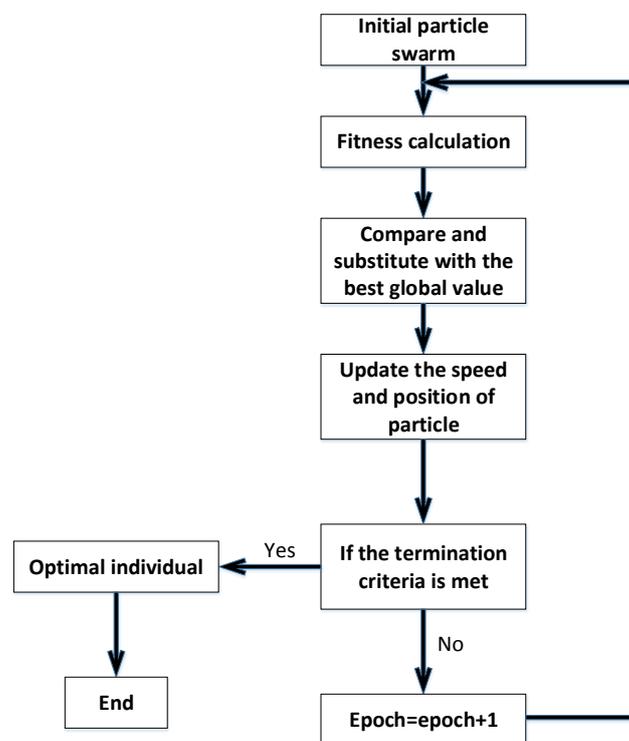


Figure 7. Working principle of PSO.

By combining two objective functions into one, the final objective function is listed below:

$$G_{best} = \min \frac{1}{0.5f + 0.5V} \quad (19)$$

R_b , R_p , α_b and α_p are the design variables, and $R_b \in [2, 5]$ cm, $R_p \in [1, 3]$ cm, $\alpha_b \in [20^\circ, 40^\circ]$ and $\alpha_p \in [0^\circ, 20^\circ]$ are set as the corresponding constraints for each design variable. Figure 8 illustrates this optimization result. It is observed that the objective function is improved approximately two times.

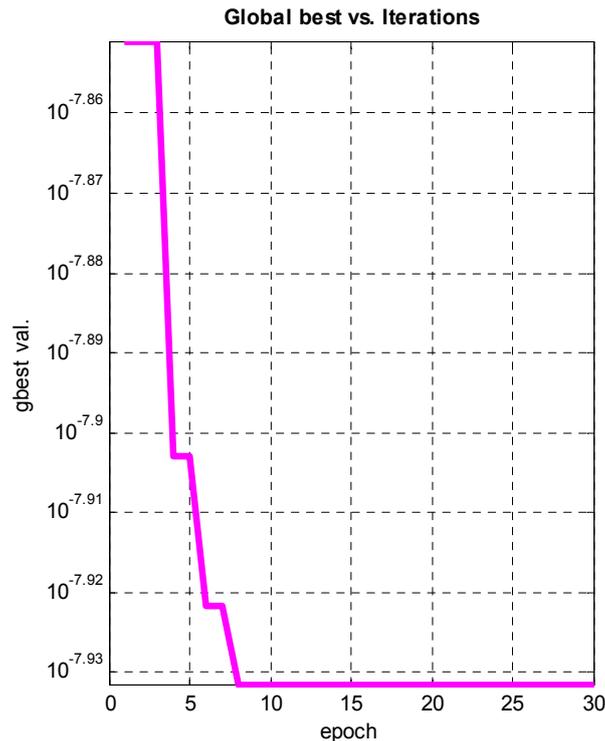


Figure 8. Optimization result.

4.2. Pure Rotational Case

If one is making the summation of leading diagonal segments' mean value and standard deviation be the smallest one, one can have a better stiffness performance. However, doing this can deteriorate the workspace performance. The authors will utilize the multi-optimization genetic algorithm solver in MATLAB. The first objective function is to reflect the stiffness performance, and the second objective function is to reflect the workspace performance. Thus, the objective function that reflects the stiffness performance is $f_1 = mean + std$, and the objective function that reflects the workspace performance is $f_2 = 2\pi(\sqrt{l_2^2 + R_p^2})^3 \cdot \sum_i \frac{1}{k_i} / 3n_{total}$.

$s = [R_p, R_b, l_1, l_2, \theta_1, \theta_2, \theta_3]$ are set as the design variables, and $R_p \in [0.40, 0.50]$ m, $R_b \in [0.70, 0.80]$ m, $l_1 \in [0.40, 0.50]$ m, $l_2 \in [0.20, 0.30]$ m, $\theta_1 \in [-30^\circ, 30^\circ]$, $\theta_2 \in [60^\circ, 120^\circ]$, and $\theta_3 \in [-30^\circ, 30^\circ]$ are their corresponding constraints. The optimization is conducted through utilizing the gamultiobj solver in MATLAB. The optimization parameters are selected as shown in Table 1. After running the optimization, the optimization result is illustrated in Figure 9.

Table 1. Optimization parameters.

Options	Parameters
The size of the population	50
Maximum of generations	66
Selection strategy	Tournament
The size of tournament	2
Crossover type	Intermediate
Crossover ratio	1
Pareto front population fraction	0.7

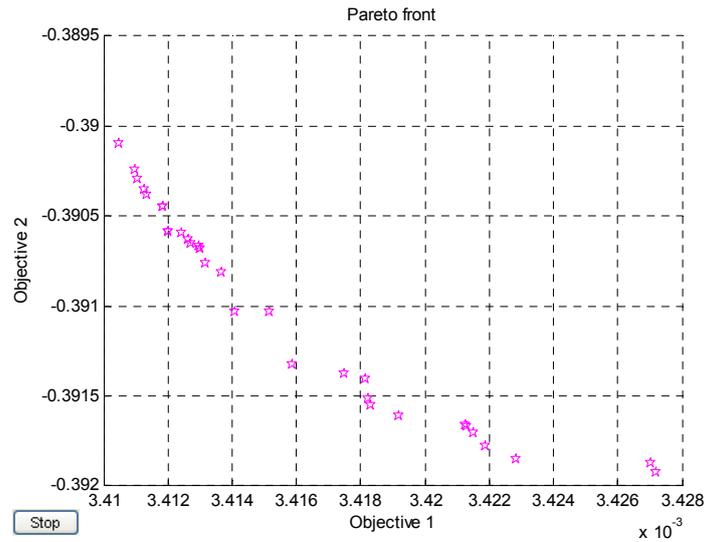


Figure 9. Optimization result for 3SPS-S.

It is observed that the first objective function that reflects the stiffness performance and the second objective function that reflects the workspace performance conflict with each other. The optimization result shows multiple possible solutions, which one calls non-dominant. One is able to select one particular solution based on one’s own situations.

4.3. Mixed Motion Case

Regarding the mixed motion case, one has:

$$f_1 = M + S \tag{20}$$

$$f_2 = \frac{\pi \cdot R_p^2 \cdot (h_{\max} - h_{\min}) \cdot KCI}{n_{total}} \tag{21}$$

$s = [R_p, R_b, \theta_5, \theta_6, z]$ are set as the design variables, and $R_p \in [0.40, 0.50]$ m, $R_b \in [0.70, 0.80]$ m, $\theta_5 \in [-120^\circ, -60^\circ]$, $\theta_6 \in [60^\circ, 120^\circ]$ and $z \in [0.2, 1.2]$ m are their corresponding constraints.

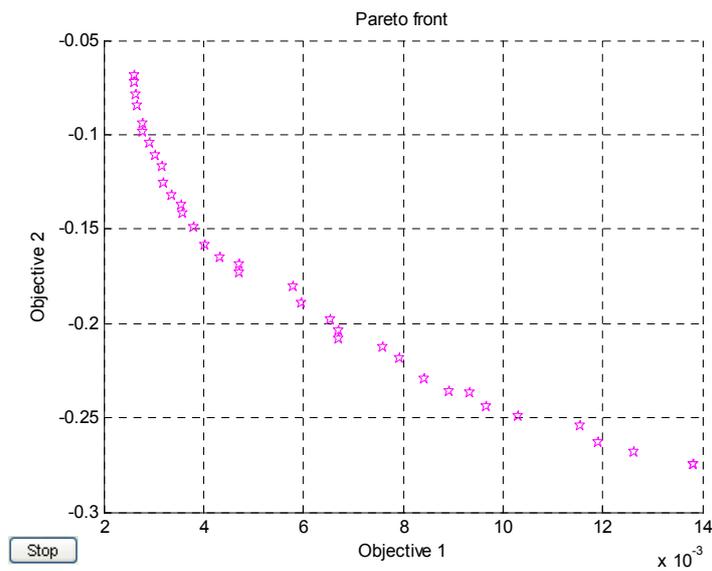


Figure 10. Optimization result for 4UPS-PU

Based on the results as illustrated in Figure 10, one is able to select one particular solution based on one's own requests.

5. Stiffness Models

The traditional stiffness model (TSM), which is very popular and well established, is able to be easily obtained on the basis of a mechanism's Jacobian. One does not take the center limb (if it is applicable) into consideration for the traditional stiffness model. In [14], the author derived a kinetostatic stiffness model under the case where one takes the compliance of the center limb into consideration and a kinetostatic compliance model (KCM) under the case where one considers the center limb as rigid on the basis of the virtual work principle. As demonstrated in [15], the authors suggested to employ a dexterous stiffness model (DSM) to reflect a robotic mechanism's singularity situation. Here, the authors will employ the 3UPS-PU robotic manipulator (as shown in Figure 11) to do a comparison among these models.

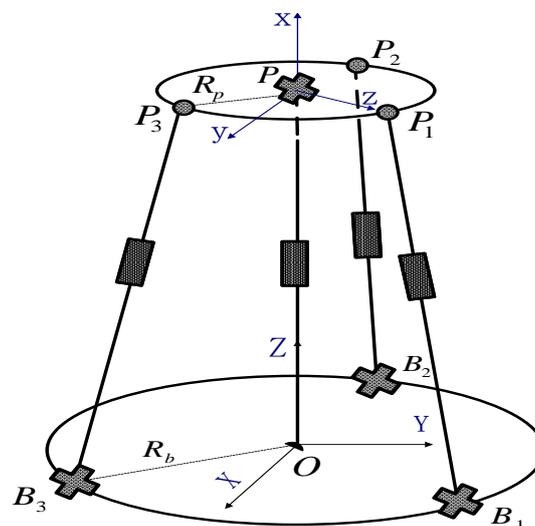


Figure 11. A 3-DOF mechanism.

5.1. TSM Case

One is able to obtain the TSM as follows:

$$K = J_{rigid}^T K_q J_{rigid} \quad (22)$$

where $K_q = diag[k_1, k_2, k_3]$. k_i denotes the joint stiffness. Since the active limbs are identical, one can rewrite Equation (22) as follows:

$$K = k J_{rigid}^T J_{rigid} = 1000 J_{rigid}^T J_{rigid} \quad (23)$$

The above matrix's leading diagonal segments can reflect corresponding moving directional stiffness.

5.2. KCM Case

The author in [14] obtained the KCM under the case where one considers the center limb as rigid. In order to make a comparison, here the authors assume the center limb to be rigid. Furthermore, we make the value of the compliance for the actuators 0.001; this will make the KCM case be consistent

with the TSM case. We can derive the compliance matrix as follows. The detailed derivation process is not illustrated here anymore; interested readers can refer to [14].

$$\Delta c = J_{serial}(AJ_{serial})^{-1}BCB^T(AJ_{serial})^{-T}J_{serial}^T w = C_c w \quad (24)$$

where C_c denotes the compliance matrix.

5.3. DSM Case

Under the case where a robotic mechanism is inside the singular boundary, the robotic mechanism's stiffness property can be deteriorated. Under this case, the leading diagonal segments cannot accurately reflect the corresponding stiffness in each moving direction [15]. Thus, in [15], the authors suggested to employ the DSM as illustrated in Equation (25) to handle the above issue. Its matrix is written as follows:

$$K_L' = \frac{1000}{(svd(J_{rigid}))^T \cdot svd(J_{rigid})} J_{rigid}^T J_{rigid} \quad (25)$$

6. Comparisons

6.1. Three Cases

As a case study, the authors here set $R_p = 0.07$, $R_b = 0.16$, $Z = 0.66$, $\theta_2 = 100^\circ$ and $\theta_3 = 0^\circ$. Under the above condition, for the TSM case, the compliance in the Z direction is calculated as 0.0003; the compliance in the θ_x direction is calculated as 0.1461; and the compliance in the θ_y direction is calculated as 0.1472. For the KCM case, the compliance in the θ_x direction is calculated as 0.1428; the compliance in the θ_y direction is calculated as 0.1461; the compliance in the θ_z direction is calculated as 0.0044; the compliance in the X direction is calculated as zero; the compliance in the Y direction is calculated as zero; and the compliance in the Z direction is calculated as 0.0003. For the DSM case, the compliance in the Z direction is calculated as 0.001; the compliance in the θ_x direction is calculated as 0.4218; and the compliance in the θ_y direction is calculated as 0.4249.

6.2. Comparison

Table 2 compares the TSM, KCM and DSM, from which it can be observed that the Z compliance of the TSM case equals that of the KCM case; this indicates that the Z stiffness of the TSM case equals that of the KCM case. The θ_x compliance offset is 0.0033; the θ_y compliance offset is 0.0011; and the sum of the compliance of the TSM case equals that of the KCM case. There is just a tiny bit of offset for both the θ_x and θ_y cases; hence, KCM has been reaffirmed and verified.

Table 2. Comparison table. KCM, kinetostatic compliance model; TSM, traditional stiffness model; DSM, dexterous stiffness model.

	KCM	TSM	DSM
Z Compliance	0.0003	0.0003	0.0010
θ_x Compliance	0.1428	0.1461	0.4218
θ_y Compliance	0.1461	0.1472	0.4249
Sum of compliances	0.2936	0.2936	0.8477

Regarding the TSM case and the DSM case, the θ_x compliance offset is 0.2757; the θ_y compliance offset is 0.2777; the Z compliance offset is 0.0007; and the compliance sum offset is 0.5541. The scale of these offsets is large, and this is expected because as can be observed from Equations (22) and (25), these two cases do show an offset. Based on Figure 12, Figure 13, and Figure 14, it can be observed that the KCM coincides with the TSM; this further reaffirms and verifies the KCM.

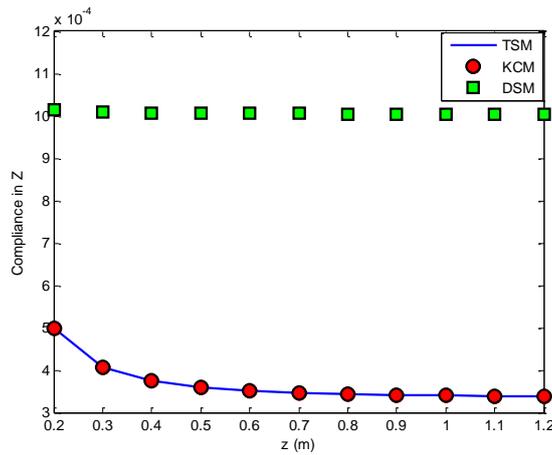


Figure 12. Z Compliance.

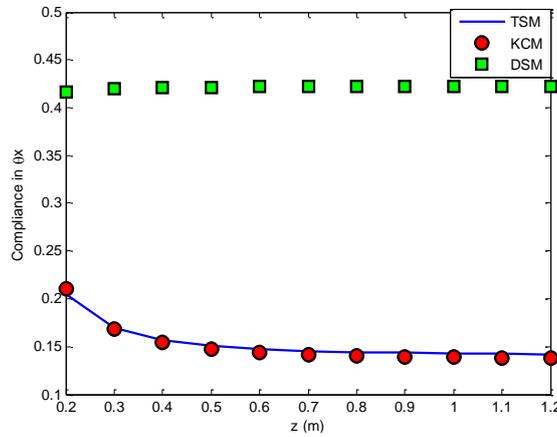


Figure 13. θ_x Compliance.

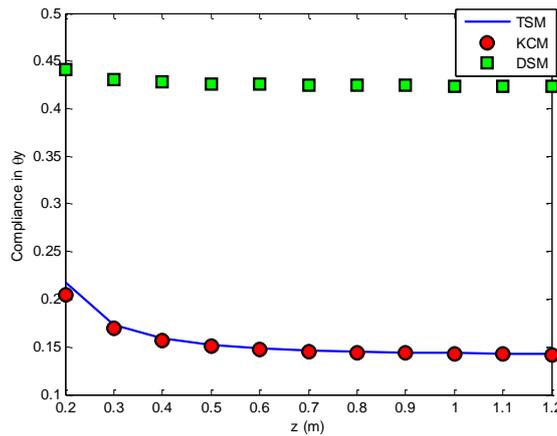


Figure 14. θ_y Compliance.

When we change the coefficient k from 1000 to 2000 in the TSM, the TSM will deviate from the KCM in terms of θ_x compliance, θ_y compliance and Z compliance (illustrated from Figure 15, Figure 16, and Figure 17), and this is expected due to the fact that the actuator stiffness is not consistent with the actuator compliance in the KCM case anymore. Therefore, based on the above analysis and demonstrations, the correlation and agreement of the TSM, KCM and DSM have been verified.

With respect to the four degrees-of-freedom case and the five degrees-of-freedom robotic manipulators case, one has similar outcomes.

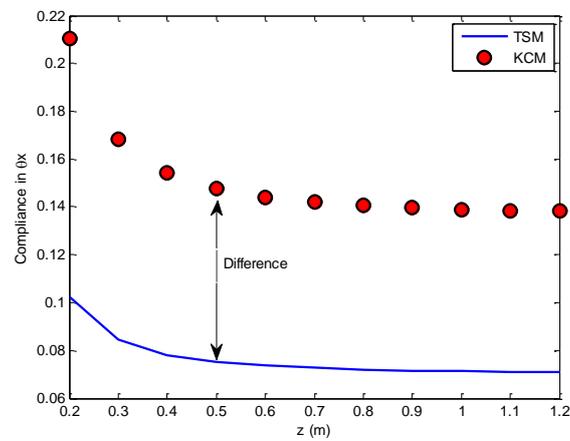


Figure 15. θ_x Compliance.

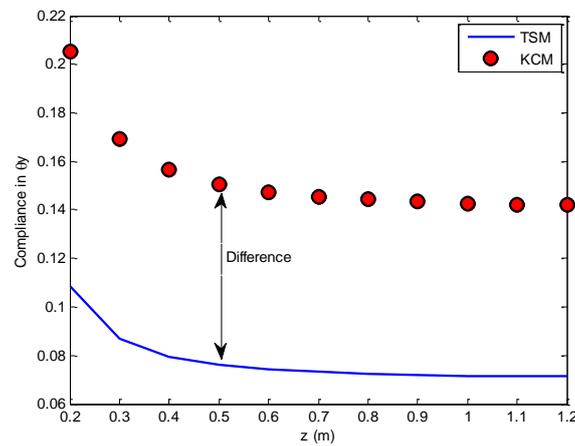


Figure 16. θ_y Compliance.

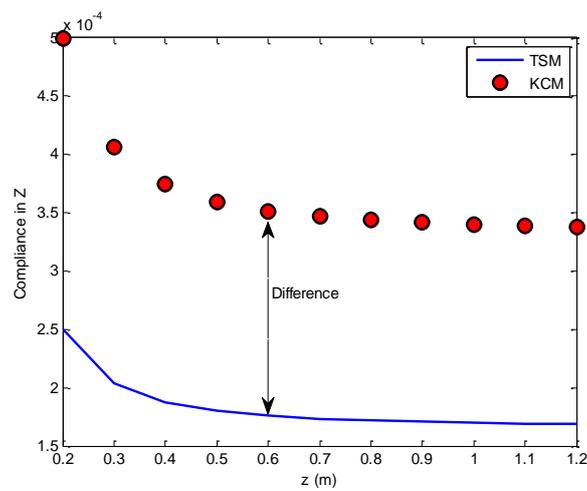


Figure 17. Z Compliance.

7. Conclusions

In this paper, the authors study the kinematic performances and optimization for three types of parallel robotic manipulators. Firstly, the kinematics analysis and Jacobian are determined for the three types of manipulators; secondly, through selecting two indices that are able to reflect the stiffness performance and workspace performance, the authors conduct the optimization analysis towards these two performances for the three types of manipulators and manifest an overall process and approach for multi-objective optimization issues of robotic manipulators; lastly, the correlation and agreement of the TSM, KCM and DSM are presented.

Acknowledgments: The authors would like to thank the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors gratefully acknowledge the financial support from the Canada Research Chairs (CRC) program.

Author Contributions: This work was supervised by Dan Zhang. Under Dan Zhang's supervision, feedback and support, Dan Zhang and Bin Wei together conducted the work.

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

References

1. Li, Y.; Xu, Q. Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation. *IEEE/ASME Trans. Mechatron.* **2007**, *12*, 265–273. [[CrossRef](#)]
2. Sakai, S.; Lida, M.; Umeda, M. Heavy material handling manipulator for agricultural robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002; pp. 1062–1068.
3. Sakai, S.; Osuka, K.; Maekawa, T.; Umeda, M. Robust Control Systems of a Heavy Material Handling Agricultural Robot: A Case Study for Initial Cost Problem. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 1038–1048. [[CrossRef](#)]
4. Zhang, D.; Wei, B. Design and analysis of a collision detector for hybrid robotic machine tools. *Sens. Transducers J.* **2015**, *193*, 67–73.
5. Bohez, E. Five-axis milling machine tool kinematic chain design and analysis. *Int. J. Mach. Tools Manuf.* **2002**, *42*, 505–520. [[CrossRef](#)]
6. Zhang, D.; Wei, B. Payload variation compensation for robotic arms through model reference control approach. *Int. J. Robot. Autom.* **2016**. [[CrossRef](#)]
7. Liang, Q.; Wu, W.; Zhang, D.; Wei, B. Design and analysis of a micromechanical three-component force sensor for characterizing and quantifying surface roughness. *Meas. Sci. Rev.* **2015**, *15*, 248–255. [[CrossRef](#)]
8. Howe, R. Tactile Sensing and Control of Robotic Manipulation. *J. Adv. Robot.* **1994**, *8*, 245–261. [[CrossRef](#)]
9. Gosselin, C.; Jorge, A. A Global Performance Index for the Kinematic Optimization of Robotic Manipulators. *J. Mech. Des.* **1991**, *113*, 220–226. [[CrossRef](#)]
10. Li, B.; Zhi, W.; Ying, H. The Stiffness Calculation Model of the New Typed Parallel Machine Tool. *Mach. Des.* **1999**, *3*, 14–16.
11. Merlet, J.P. Determination of the Orientation Workspace of Parallel Manipulators. *J. Intell. Robot. Syst.* **1995**, *13*, 143–160. [[CrossRef](#)]
12. Stamper, R.E.; Tsai, L.; Walsh, G.C. Optimization of a Three DOF Translational Platform for Well-conditioned Workspace. In Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, USA, 25 April 1997; pp. 3250–3255.
13. Caponetto, R.; Fortuna, L.; Fazzino, S.; Xibilia, M. Chaotic Sequences to Improve the Performance of Evolutionary Algorithms. *IEEE Trans. Evolut. Comput.* **2003**, *7*, 289–304. [[CrossRef](#)]
14. Zhang, D. *Parallel Robotic Machine Tools*; Springer: New York, NY, USA, 2009.
15. Zhang, D.; Gao, Z. Forward kinematics, performance analysis, and multi-objective optimization of a bio-inspired parallel manipulator. *Robot. Comput.-Integr. Manuf.* **2012**, *28*, 484–492. [[CrossRef](#)]

