

## Article

# Dynamics Modeling and Redundant Force Optimization of Modular Combination Parallel Manipulator

Aimin Jiang <sup>1,2</sup>, Hasiaoqier Han <sup>1,2,\*</sup>, Chunyang Han <sup>1</sup>, Shuai He <sup>1</sup>, Zhenbang Xu <sup>1</sup> and Qingwen Wu <sup>1</sup><sup>1</sup> Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, No. 3888, Dong Nanhu Road, Changchun 130033, China<sup>2</sup> Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

\* Correspondence: aoqier@ciomp.ac.cn

**Abstract:** The limb-driving force mutation of the modular combination parallel manipulator (MCPM) affects the alignment process of optical axis. In this paper, a novel optimization method based on the force mutation penalty term is proposed to solve the problem of driving force mutation. The kinematics and dynamics models of the manipulator are established using a modularization idea, reducing the complexity of the modeling process, and verified using co-simulation. Moreover, particle swarm optimization (PSO) is applied as an optimization tool. The effectiveness of the proposed method is confirmed by comparing it with the minimize-the-maximum and Moore–Penrose (M–P) methods, which are widely used to solve parallel manipulators with redundant drives.

**Keywords:** parallel manipulator; force optimization; module; dynamics; redundant drives; PSO



**Citation:** Jiang, A.; Han, H.; Han, C.; He, S.; Xu, Z.; Wu, Q. Dynamics Modeling and Redundant Force Optimization of Modular Combination Parallel Manipulator. *Machines* **2023**, *11*, 247. <https://doi.org/10.3390/machines11020247>

Academic Editor: Zheng Chen

Received: 26 December 2022

Revised: 1 February 2023

Accepted: 5 February 2023

Published: 7 February 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

The alignment of the optical axis of a small aperture space optical remote sensor in space environment simulator is achieved with the Stewart parallel manipulator. With the increase in the aperture, the requirements for the size of the moving platform and load capacity of the Stewart parallel manipulator increase. The traditional six-point support structure will lead to large deformation of the moving platform, severe stress concentration at the support point, excessive driving force, and other problems. The load capacity of the parallel manipulator can be increased without increasing the height and driving force by splicing the platforms together.

Many scholars have researched kinematics and dynamics on the parallel manipulator [1–4]. Most of the research on the MCPM focuses on the combination of the drive unit as a module or the parallel manipulator as a module. Arrouk et al. [5] designed a 3-CRS parallel kinematic machine (PRM) consisting of three modular kinematic chains. Tang et al. [6] designed a new modular-based hybrid PRM using the joint as a module. Zhang et al. [7] designed a generalized spherical parallel mechanism with partially decoupled characteristics by combining position and orientation modules. Hao et al. [8] used identical spatial modules to construct a large-range modular XYZ compliant parallel manipulator. Until now, the research of parallel manipulators with redundant actuation focuses on adding one or several redundant drives based on a particular configuration. Wu et al. [9] added an RRR leg to the planar 2-RRR parallel mechanism. Xi et al. [10] added a redundant leg to the three-degrees-of-freedom (DOF) tricept parallel mechanism. Yao et al. [11] added actuation to the middle PRPU branch of the 5UPS parallel platform. Zhao et al. [12] added two PSS legs to the planar 6PSS parallel manipulator. However, there are few studies on the Stewart platform as a modular, redundant unit.

The dynamics equation of parallel manipulator with redundant actuation is non-homogeneous linear equations [13], with multiple solutions to the driving force. Xu et al. [14]

used the Moore–Penrose (M–P) method to solve the driving force of redundant parallel manipulators and discussed whether the calculation results contain internal forces. Xie et al. [15] developed a novel 3DOF parallel mechanism driven by four legs and solved the driving force distribution using the least squares norm method. Lee et al. [16] studied the actuator power consumption of a 3-RRR fully planar parallel manipulator and proposed a driving force optimization method to minimize the electrical energy consumed, finding that 26% of energy could be saved. Jiang et al. [17] designed a novel planar 2-DOF parallel kinematic machine with kinematic redundancy, and a method for redundant force optimization is presented to improve the precision of the machine. Hino et al. [18] used minimize-the-maximum actuator force as the objective function, using the linear matrix inequality method to solve the actuator force. Wu et al. [19] used the minimum actuator torque range into account and proposed a novel torque optimization method of a 3-DOF redundantly actuated parallel manipulator for friction stir welding. The primary methods of redundant force distribution are the M–P method [20–22], minimum norm method [23–26], minimum energy consumption method [27–29], minimum structural deformation method [30,31], maximize manipulator accuracy [32,33], and minimize-the-maximum driving force method [9,34]. The driving force mutation of the modular combination parallel manipulator (MCPM) will affect the alignment process of the optical axis. However, these methods solve the driving force independently at each discrete point of the trajectory; the change in the driving force of the adjacent discrete points is not considered.

In this paper, we proposed a novel optimization method based on the force mutation penalty term to solve force mutation of the MCPM. Moreover, particle swarm optimization (PSO) is applied as optimization tool. The proposed method is compared with the minimize-the-maximum and M–P methods (i.e., least squares method) moving along the same trajectory. The remainder of this paper is organized as follows. In the next section, the structure and inverse kinematics of the MCPM is analyzed. Then, its dynamic model is constructed in Section 3. In Section 4, the general solution of the driving force is deduced. Then, force mutation is minimized by optimizing the distribution of redundant force. Section 5 gives two simulation examples of MCPM to verify the proposed method's effectiveness. Finally, the conclusions of this paper are provided.

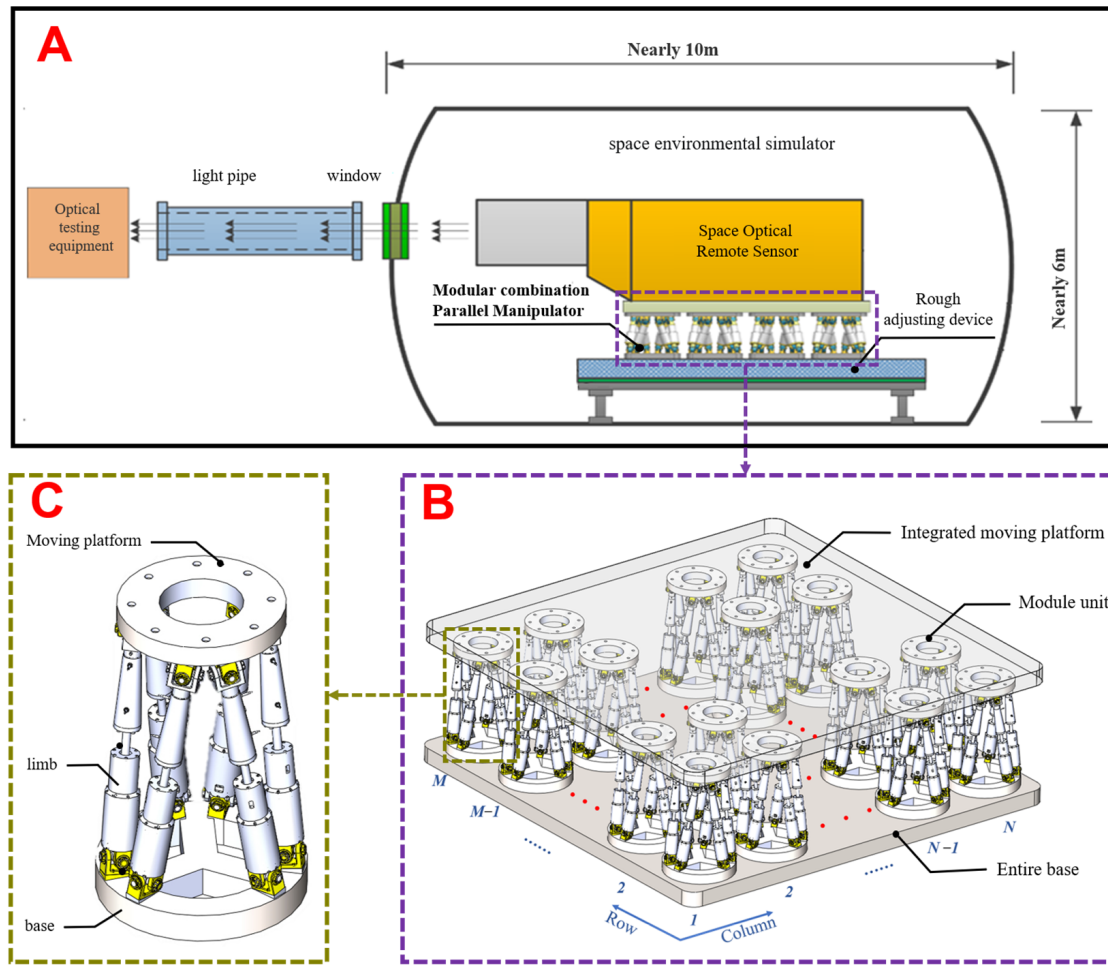
## 2. Structure Description and Inverse Kinematics

### 2.1. Structure Description

A modular combination parallel manipulator (MCPM) is developed for the optical axis alignment of a space optical remote sensor with an aperture over two meters in a space environment simulator. As depicted in Figure 1, the modular combination parallel manipulator (MCPM) comprises  $M \times N$  6-DOF parallel platforms, each module unit consisting of a moving platform, base, and 6-UPU limbs. Each base is connected to the entire base, and each moving platform is connected to the integrated moving platform. The space optical remote sensor is installed above the integrated moving platform. A rough adjustment device and MCPM are used to achieve rough and accurate optical axis alignment in the space environment simulator. First, the space optical remote sensor opens the imaging function and makes the target in the light pipe within the imaging range by adjusting the rough adjusting device. Then, make the target in the middle of the image by adjusting the MCPM.

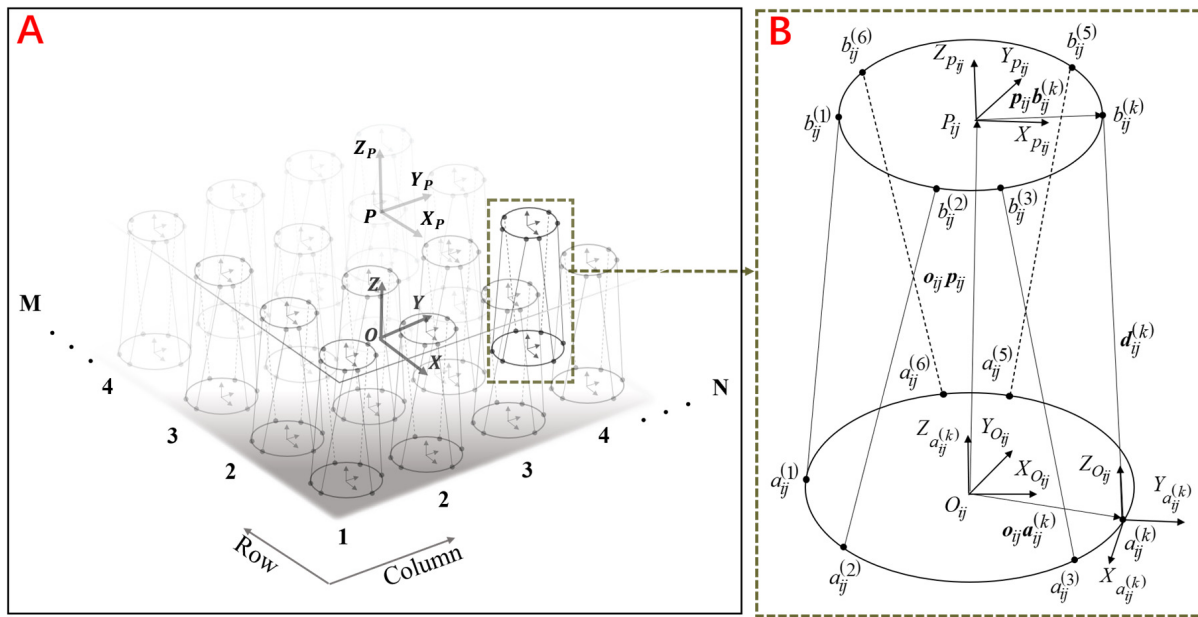
### 2.2. Inverse Kinematics

This section uses a modularization idea to establish the MCPM kinematic model. The modeling process of the modularization idea is as follows. Firstly, the global systems are established on the MCPM level, and the coordinate systems for internal use are established at the module unit level. Then, the kinematic analysis between MCPM and module unit is carried out at the MCPM level. Finally, the kinematics analysis is carried out on the module unit level.



**Figure 1.** Schematic of (A) application scenario, (B) modular combination parallel manipulator (MCPM), and (C) module unit.

As illustrated in Figure 2A, an integrated moving coordinate system  $\{P\} \{P - X_P Y_P Z_P\}$  and an entire base coordinate system  $\{O\} \{O - XYZ\}$  are established in the MCPM. The coordinate frame  $\{P\}$  is located at the center point  $P$  of the integrated moving platform. The coordinate frame  $\{O\}$  is attached to the entire base at its center point  $O$ . As depicted in Figure 2B, a module moving coordinate system  $\{P_{ij}\} \{P_{ij} - X_{P_{ij}} Y_{P_{ij}} Z_{P_{ij}}\}$ , module base coordinate system  $\{O_{ij}\} \{O_{ij} - X_{O_{ij}} Y_{O_{ij}} Z_{O_{ij}}\}$ , and six limbs coordinate system are established in the module unit  $ij$ , which is located in row  $i$  and column  $j$ . The coordinate frame  $\{P_{ij}\}$  is located at the center point  $P_{ij}$  of the moving platform. The coordinate frame  $\{O_{ij}\}$  is attached to the base at its center point  $O_{ij}$ . Six limb coordinate systems  $\{A_{ij}^{(k)}\} \{a_{ij}^{(k)} - X_{a_{ij}^{(k)}} Y_{a_{ij}^{(k)}} Z_{a_{ij}^{(k)}}\}$  are fixed to each limb and located at the center of the joint  $a_{ij}^{(k)}$  connecting the base and the limb. The  $Z_{a_{ij}^{(k)}}$  axis points from  $a_{ij}^{(k)}$  to  $b_{ij}^{(k)}$ ; the  $Y_{a_{ij}^{(k)}}$  axis is parallel to the cross product of two unit vectors of the  $Z_{O_{ij}}$  and  $Z_{a_{ij}^{(k)}}$  axes; the  $X_{a_{ij}^{(k)}}$  axis is defined by the right hand. The frame  $\{O\}$  and  $\{O_{ij}\}$  are in parallel. The frame  $\{P\}$  and  $\{P_{ij}\}$  are also in parallel.



**Figure 2.** Kinematics model of (A) MCPM and (B) module unit.

### 2.2.1. Kinematic Analysis between MCPM and Module Unit

Velocity transformation between the integrated moving platform and moving platform of the module unit  $ij$  in frame  $\{O\}$  can be written as

$$v_{P_{ij}} = T_{P_{ij}} \dot{x}_P \quad (1)$$

where  $T_{P_{ij}} = [E_{3 \times 3} - P\tilde{P}_{ij}]$  is a transformation matrix.

According to the transformation formula between different coordinate systems, moving platform's velocity  $v_{P_{ij}}$  can be transferred to the frame  $\{O_{ij}\}$  as follows:

$${}^{O_{ij}}v_{P_{ij}} = {}^O_{O_{ij}}R v_{P_{ij}} \quad (2)$$

where  ${}^O_{O_{ij}}R$  is the rotation matrix of frame  $\{O\}$  relative to frame  $\{O_{ij}\}$ .

Substituting Equation (1) into Equation (2) yields

$${}^{O_{ij}}v_{P_{ij}} = {}^P_{P_{ij}}J \dot{x}_P \quad (3)$$

where  ${}^P_{P_{ij}}J = {}^O_{O_{ij}}R T_{P_{ij}}$ .

### 2.2.2. Module Unit Inverse Kinematics

As depicted in Figure 2B, the closed-loop vector equation for each limb of module unit  $ij$  can be written as

$${}^{O_{ij}}d_{ij}^{(k)} + {}^{O_{ij}}o_{ij}a_{ij}^{(k)} = {}^{O_{ij}}o_{ij}p_{ij} + {}^{P_{ij}}_{P_{ij}}R {}^{P_{ij}}p_{ij}b_{ij}^{(k)} \quad (4)$$

where  ${}^{P_{ij}}_{P_{ij}}R$  is the rotation matrix of frame  $\{P_{ij}\}$  relative to frame  $\{O_{ij}\}$ .

The velocity of joint  $b_{ij}^{(k)}$  is found by the time derivative of Equation (4).

$${}^{O_{ij}}v_{b_{ij}^{(k)}} = {}^{b_{ij}^{(k)}}_{P_{ij}}T {}^{P_{ij}}_{P_{ij}} \dot{x}_{P_{ij}} \quad (5)$$



where  ${}_{P_{ij}}^{b_{ij}^{(k)}} T = \begin{bmatrix} E_{3 \times 3} - \tilde{O_{ij}} P_{ij}^{b_{ij}^{(k)}} \end{bmatrix}$  is a transformation matrix.

The velocity of joint  $b_{ij}^{(k)}$  can be transferred to frame  $\{A_{ij}^{(k)}\}$ .

$$A_{ij}^{(k)} v_{b_{ij}^{(k)}} = {}_{O_{ij}}^{A_{ij}^{(k)}} R O_{ij} v_{b_{ij}^{(k)}} \quad (6)$$

where  ${}_{O_{ij}}^{A_{ij}^{(k)}} R$  is the rotation matrix of frame  $\{O_{ij}\}$  relative to frame  $\{A_{ij}^{(k)}\}$ .

Substituting Equation (5) into (6) yields

$$A_{ij}^{(k)} v_{b_{ij}^{(k)}} = {}_{P_{ij}}^{b_{ij}^{(k)}} J \dot{O_{ij}} x_{P_{ij}} \quad (7)$$

where  ${}_{P_{ij}}^{b_{ij}^{(k)}} J = {}_{O_{ij}}^{A_{ij}^{(k)}} R {}_{P_{ij}}^{b_{ij}^{(k)}} T$ .

In frame  $\{A_{ij}^{(k)}\}$ , the angular velocity of the  $k$ th limb can be written as

$$A_{ij}^{(k)} \omega = \frac{1}{d_{a_{ij}^{(k)}}} \left( A_{ij}^{(k)} s \times A_{ij}^{(k)} v_{b_{ij}^{(k)}} \right) \quad (8)$$

where  $d_{a_{ij}^{(k)}}$  is the distance between joints  $a_{ij}^{(k)}$  and  $b_{ij}^{(k)}$ , and  $A_{ij}^{(k)} s = [0 \ 0 \ 1]$  is the unit vector of the  $k$ th limb in frame  $\{A_{ij}^{(k)}\}$ .

Substituting Equation (7) into (8) yields

$$A_{ij}^{(k)} \omega = {}_{P_{ij}}^{a_{ij}^{(k)}} J \dot{O_{ij}} x_{P_{ij}} \quad (9)$$

where  ${}_{P_{ij}}^{a_{ij}^{(k)}} J \omega = \frac{1}{d_{a_{ij}^{(k)}}} A_{ij}^{(k)} \tilde{s} {}_{P_{ij}}^{b_{ij}^{(k)}} J$ .

In the  $k$ th limb, the velocity of the cylinder's centroid and piston's centroid can be expressed as

$$A_{ij}^{(k)} v_{a_{ij}^{(k1)}} = l_1 A_{ij}^{(k)} \omega \times A_{ij}^{(k)} s \quad (10)$$

$$A_{ij}^{(k)} v_{a_{ij}^{(k2)}} = \left( d_{a_{ij}^{(k)}} - l_2 \right) A_{ij}^{(k)} \omega \times A_{ij}^{(k)} s + d_{a_{ij}^{(k)}} \dot{A_{ij}^{(k)}} s \quad (11)$$

where  $l_1$  is the distance between joint  $a_{ij}^{(k)}$  and the  $k$ th cylinder's centroid, and  $l_2$  is between joint  $b_{ij}^{(k)}$  and the  $k$ th piston's centroid.

Substituting Equations (7) and (8) into Equations (10) and (11) yields

$$A_{ij}^{(k)} v_{a_{ij}^{(k1)}} = {}_{P_{ij}}^{a_{ij}^{(k1)}} J \dot{O_{ij}} x_{P_{ij}} \quad (12)$$

$$A_{ij}^{(k)} v_{a_{ij}^{(k2)}} = {}_{P_{ij}}^{a_{ij}^{(k2)}} J \dot{O_{ij}} x_{P_{ij}} \quad (13)$$

where

$${}_{P_{ij}}^{a_{ij}^{(k1)}} J = -\frac{l_1}{d_{a_{ij}^{(k)}}} A_{ij}^{(k)} \tilde{s} A_{ij}^{(k)} \tilde{s} {}_{P_{ij}}^{b_{ij}^{(k)}} J,$$

$${}_{P_{ij}}^{a_{ij}^{(k2)}} J = \begin{bmatrix} A_{ij}^{(k)} s - \frac{d_{a_{ij}^{(k)}} - l_2}{d_{a_{ij}^{(k)}}} \tilde{A}_{ij}^{(k)} \tilde{s} A_{ij}^{(k)} \tilde{s} \\ b_{ij}^{(k)} \\ p_{ij} \end{bmatrix} J.$$

### 3. Inverse Dynamics

In this section, a modularization idea is used to establish the MCPM dynamical model. The modeling process of the modularization idea is as follows. Firstly, analyze the applied and inertia force of MCPM. Then, analyze the applied and inertia force of the module unit. Finally, the dynamic model of the entire system is derived using the virtual work principle.

#### 3.1. MCPM Applied and Inertia Force

Assume that the mass of load is  $m_z$ , the mass of integrated moving platform is  $m_p$ , and the integrated moving platform is subjected to the external force  $F_{ij}$  acting on point  $O_{ij}$  by module unit  $ij$ . The applied and inertia forces imposed on the load's centroid  $P_z$  and integrated moving platform's centroid  $P$  can be expressed as Equations (14) and (15).

$$F_z = \begin{bmatrix} m_z g - m_z \ddot{v}_z \\ -I_z \dot{\omega}_p - \omega_p \times (I_z \omega_p) \end{bmatrix} \quad (14)$$

$$F_p = \begin{bmatrix} m_p g - m_p \ddot{v}_p \\ -I_p \dot{\omega}_p - \omega_p \times (I_p \omega_p) \end{bmatrix} \quad (15)$$

#### 3.2. Module Unit Applied and Inertia Force

Assuming that module unit  $ij$  is subjected to the external force  $-F_{ij}$  acting on point  $O_{ij}$  by the integrated moving platform. In frame  $\{O_{ij}\}$ , the applied and inertia force  ${}^{O_{ij}}F_{P_{ij}}$  imposed on the center of moving platform can be expressed as Equation (16). In frame  $\{A_{ij}^{(k)}\}$ , the applied and inertia forces imposed on the  $k$ th cylinder's centroid and piston's centroid can be expressed as Equations (17) and (18).

$${}^{O_{ij}}F_{P_{ij}} = \begin{bmatrix} m_{p_{ij}O} {}^{O_{ij}}Rg - m_{p_{ij}O} \ddot{v}_{P_{ij}} \\ -{}^{O_{ij}}I_{P_{ij}} \dot{\omega}_{P_{ij}} - {}^{O_{ij}}\omega_{P_{ij}} \times ({}^{O_{ij}}I_{P_{ij}} {}^{O_{ij}}\omega_{P_{ij}}) \end{bmatrix} \quad (16)$$

$${}^{A_{ij}^{(k)}}F_{a_{ij}^{(k1)}} = \begin{bmatrix} m_{a_{ij}^{(k1)}O_{ij}} {}^{A_{ij}^{(k)}}Rg - m_{a_{ij}^{(k1)}O_{ij}} \ddot{v}_{a_{ij}^{(k1)}} \\ -{}^{A_{ij}^{(k)}}I_{a_{ij}^{(k1)}} \dot{\omega}_{a_{ij}^{(k1)}} - {}^{A_{ij}^{(k)}}\omega_{a_{ij}^{(k1)}} \times ({}^{A_{ij}^{(k)}}I_{a_{ij}^{(k1)}} {}^{A_{ij}^{(k)}}\omega_{a_{ij}^{(k1)}}) \end{bmatrix} \quad (17)$$

$${}^{A_{ij}^{(k)}}F_{a_{ij}^{(k2)}} = \begin{bmatrix} m_{a_{ij}^{(k2)}O_{ij}} {}^{A_{ij}^{(k)}}Rg - m_{a_{ij}^{(k2)}O_{ij}} \ddot{v}_{a_{ij}^{(k2)}} \\ -{}^{A_{ij}^{(k)}}I_{a_{ij}^{(k2)}} \dot{\omega}_{a_{ij}^{(k2)}} - {}^{A_{ij}^{(k)}}\omega_{a_{ij}^{(k2)}} \times ({}^{A_{ij}^{(k)}}I_{a_{ij}^{(k2)}} {}^{A_{ij}^{(k)}}\omega_{a_{ij}^{(k2)}}) \end{bmatrix} \quad (18)$$

### 3.3. Dynamics Equation

The dynamics equation of the entire system is established based on the principle of virtual work.

$$\begin{aligned}
 & (\delta \mathbf{x}_z)^T \mathbf{F}_z + (\delta \mathbf{x}_p)^T \mathbf{F}_p + \sum_{j=1}^N \sum_{i=1}^M \left( \delta \mathbf{x}_{P_{ij}} \right)^T \mathbf{F}_{ij} + \dots \\
 & \sum_{j=1}^N \sum_{i=1}^M \left[ \sum_{k=1}^6 \left( \left( \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{b_{ij}^{(k)}} \right)^T \mathbf{A}_{ij}^{(k)} \boldsymbol{\tau} + \left( \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{a_{ij}^{(k1)}} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k1)}} + \dots \right. \right. \\
 & \left. \left. \left( \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{a_{ij}^{(k2)}} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k2)}} \right) + \left( \mathbf{O}_{ij} \delta \mathbf{x}_{P_{ij}} \right)^T \mathbf{O}_{ij} \mathbf{F}_{P_{ij}} + (\delta \mathbf{x}_p)^T (-\mathbf{F}_{ij}) \right] = 0,
 \end{aligned} \quad (19)$$

where  ${}^H \delta \mathbf{x}_e$  means a virtual displacement of point  $e$  in frame  $\{\mathbf{H}\}$ , and  $\mathbf{A}_{ij}^{(k)} \boldsymbol{\tau}$  is the driving force of the  $k$ th limb in frame  $\{\mathbf{A}_{ij}^{(k)}\}$ .

Based on the previous kinematics analysis, the virtual displacement can be expressed as follows

$$\begin{cases}
 \mathbf{O}_{ij} \delta \mathbf{x}_{P_{ij}} = \mathbf{T}_{ij} \delta \mathbf{x}_p \\
 \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{b_{ij}^{(k)}} = \mathbf{T}_{ij}^{(k)} \mathbf{O}_{ij} \delta \mathbf{x}_{P_{ij}} \\
 \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{a_{ij}^{(k1)}} = \mathbf{T}_{ij}^{(k1)} \mathbf{O}_{ij} \delta \mathbf{x}_{P_{ij}} \\
 \mathbf{A}_{ij}^{(k)} \delta \mathbf{x}_{a_{ij}^{(k2)}} = \mathbf{T}_{ij}^{(k2)} \mathbf{O}_{ij} \delta \mathbf{x}_{P_{ij}}
 \end{cases} \quad (20)$$

where  $\mathbf{T}_{ij}$ ,  $\mathbf{T}_{ij}^{(k)}$ ,  $\mathbf{T}_{ij}^{(k1)}$  and  $\mathbf{T}_{ij}^{(k2)}$  are virtual displacement transformation matrices.

Assuming that the load is fixed on the integrated moving platform, its virtual displacement equation can be easily obtained using the inverse kinematics.

$$\delta \mathbf{x}_z = \mathbf{T}_L \delta \mathbf{x}_p \quad (21)$$

where  $\mathbf{T}_L$  is the transformation matrix of  $\delta \mathbf{x}_p$  relative to  $\delta \mathbf{x}_z$ .

Substituting Equations (20) and (21) into Equation (19) yields

$$\begin{aligned}
 & (\delta \mathbf{x}_p)^T \left\{ (\mathbf{T}_L)^T \mathbf{F}_z + \mathbf{F}_p + \dots \right. \\
 & \left. \sum_{j=1}^N \sum_{i=1}^M (\mathbf{T}_{ij})^T \left[ \sum_{k=1}^6 \left( \left( \mathbf{T}_{ij}^{(k)} \right)^T \mathbf{A}_{ij}^{(k)} \boldsymbol{\tau} + \left( \mathbf{T}_{ij}^{(k1)} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k1)}} + \left( \mathbf{T}_{ij}^{(k2)} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k2)}} \right) \right. \right. \\
 & \left. \left. \dots + \mathbf{O}_{ij} \mathbf{F}_{P_{ij}} \right] \right\} = 0
 \end{aligned} \quad (22)$$

Because Equation (22) applies to any  $\delta \mathbf{x}_p$ , it can be rewritten as

$$\begin{aligned}
 & (\mathbf{T}_L)^T \mathbf{F}_z + \mathbf{F}_p + \dots \\
 & \sum_{j=1}^N \sum_{i=1}^M (\mathbf{T}_{ij})^T \left[ \sum_{k=1}^6 \left( \left( \mathbf{T}_{ij}^{(k)} \right)^T \mathbf{A}_{ij}^{(k)} \boldsymbol{\tau} + \left( \mathbf{T}_{ij}^{(k1)} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k1)}} + \left( \mathbf{T}_{ij}^{(k2)} \right)^T \mathbf{A}_{ij}^{(k)} \mathbf{F}_{a_{ij}^{(k2)}} \right) \right. \\
 & \left. \dots + \mathbf{O}_{ij} \mathbf{F}_{P_{ij}} \right] = 0
 \end{aligned} \quad (23)$$

After variable substitution, Equation (23) can be simplified as

$$\mathbf{J}_z \mathbf{F}_z + \mathbf{F}_H + \mathbf{J} \boldsymbol{\tau} + \mathbf{J}_1 \mathbf{F}_1 + \mathbf{J}_2 \mathbf{F}_2 = 0, \quad (24)$$

where  $\mathbf{J}_z = (\mathbf{T}_L)^T$ ,

$$\mathbf{F}_H = \mathbf{F}_p + \sum_{j=1}^N \sum_{i=1}^M (\mathbf{T}_{ij})^T \mathbf{O}_{ij} \mathbf{F}_{P_{ij}},$$

$$\begin{aligned}
J &= \left[ J_{11}^{(1)} \cdots J_{11}^{(6)} J_{21}^{(1)} \cdots J_{ij}^{(k)} \cdots J_{MN}^{(6)} \right]^T \text{ where } J_{ij}^{(k)} = (T_{ij})^T (T_{ij}^{(k)})^T, \\
\tau &= \left[ A_{11}^{(1)} \tau \cdots A_{11}^{(6)} \tau A_{21}^{(1)} \tau \cdots A_{ij}^{(k)} \tau \cdots A_{MN}^{(6)} \tau \right], \\
F_1 &= \left[ A_{11}^{(1)} F_{a_{11}^{(11)}} \cdots A_{11}^{(6)} F_{a_{11}^{(61)}} A_{12}^{(1)} F_{a_{12}^{(11)}} \cdots A_{ij}^{(k)} F_{a_{ij}^{(k1)}} \cdots A_{MN}^{(6)} F_{a_{MN}^{(61)}} \right], \\
J_1 &= \left[ J_{11}^{(1)} \cdots J_{11}^{(6)} J_{21}^{(1)} \cdots J_{ij}^{(k)} \cdots J_{MN}^{(6)} \right]^T \text{ where } J_{ij}^{(k)} = (T_{ij})^T (T_{ij}^{(k1)})^T, \\
F_2 &= \left[ A_{11}^{(1)} F_{a_{11}^{(12)}} \cdots A_{11}^{(6)} F_{a_{11}^{(62)}} A_{12}^{(1)} F_{a_{12}^{(12)}} \cdots A_{ij}^{(k)} F_{a_{ij}^{(k2)}} \cdots A_{MN}^{(6)} F_{a_{MN}^{(62)}} \right], \\
J_2 &= \left[ J_{11}^{(1)} \cdots J_{11}^{(6)} J_{21}^{(1)} \cdots J_{ij}^{(k)} \cdots J_{MN}^{(6)} \right]^T \text{ where } J_{ij}^{(k)} = (T_{ij})^T (T_{ij}^{(k2)})^T.
\end{aligned}$$

where  $J_z$ ,  $J$ ,  $J_1$ , and  $J_2$  are transformation matrices, which could convert  $F_z$ ,  $\tau$ ,  $F_1$ , and  $F_2$  to frame  $\{O\}$  and translate point of force action to point  $P$  at the same time,  $F_1$  and  $F_2$  are vectors of resultant of applied and inertia wrenches exerted at cylinder's and piston's centroid of all limbs.  $\tau$  is a vector of the driving force of all limbs.  $F_H$  is a vector of the resultant of applied and inertia wrenches exerted at the integrated moving platform's centroid and moving platform's centroid of all module units.

#### 4. Redundant Driving Force Optimization

##### 4.1. General Solution of Driving Force

For a given load and trajectory, the dynamics equation of the MCPM is a non-homogeneous linear equation. There are multiple solutions to the limb-driving force. For Equation (24), based on the solution method of the non-homogeneous linear equations, the solution of limb-driving forces can be written as

$$F = C_1 X + C_2, \quad (25)$$

where  $C_2$  is the M-P solution which is given in Appendix A,  $C_1 = N(J)$  is the null space matrix of  $J$ ,  $X = [x_1 \cdots x_i \cdots x_{6 \times (M \times N - 1)}]$  is  $1 \times (6 \times (M \times N - 1))$  constant matrix, and  $x_i \in (-\infty, +\infty)$ . By taking different  $x_i$  values, we can obtain an infinite number of solutions.

##### 4.2. Driving Force Optimization Method

The driving force mutation of MCPM affects optical axis alignment. However, the traditional optimization method does not consider the change in the driving force of the adjacent discrete points, causing force mutation during MCPM movement. Consequently, given that the driving force in actual work is a bounded variable, too large a driving force causes the driver element to fail—a smaller driving force is more optimal. Therefore, this paper designs an improved minimize-the-maximum driving force optimization method. The improved optimization method's objective function is to minimize the maximum value of all driving forces, with the change in the driving force multiplied by the penalty coefficient added as the penalty term. The objective optimization function of the improved optimization method can be expressed as

$$f(x) = \min_x \left[ \max_t (|F_t| + w \times |F_t - F_{t-1}|) \right], \quad (26)$$

where  $w$  is the penalty coefficient and  $w \geq 0$ ,  $w \times |F_t - F_{t-1}|$  is the penalty term,  $|F_t|$  is the absolute value of  $F$  at time  $t$  in a given trajectory,  $|F_t - F_{t-1}|$  is the absolute value of  $F$  change between time  $t$  and  $t - 1$  in the given trajectory.

The optimization results can be changed by adjusting the penalty coefficient. When the penalty coefficient  $w = 0$ , it equals the minimize-the-maximum optimization. The optimization result is minimal driving force but with significant force mutation. When the penalty coefficient is infinitely great, it equals the minimum-force-mutation optimization.

tion. The optimization result is the minimum-force mutation but a sizeable driving force. Therefore, the reduction in force mutation is at the cost of increasing the driving force.

Minimize-the-maximum optimization is a special optimization problem. It uses the maximum value of each group to form a new group and selects the minimum value from the new group. Most previous studies on the minimize-the-maximum optimization problem are conducted under the condition of one redundant driving force. Moreover, the problem can be solved by geometric methods. However, the geometric method is no longer suitable when there are too many redundant driving forces. Therefore, this paper proposes applying PSO as the optimization tool under the condition of multiple redundant driving forces. PSO can quickly converge to the global optimal solution. For a given load and trajectory, the optimization process is depicted in Figure 3.

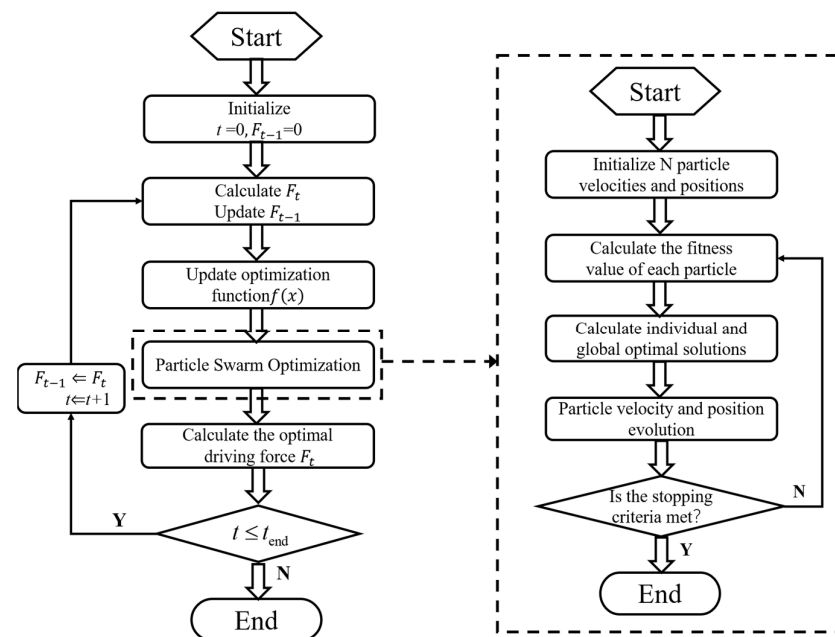


Figure 3. Flowchart of particle swarm optimization (PSO).

In the process of optical axis adjustment, the appropriate penalty coefficient value must be selected to reduce the torque of the driving element under the premise of meeting the requirements of the force mutation index. For a given load and trajectory, the procedure for determining the value of penalty coefficient can be described as follows:

**Step 1.** Calculate the maximum force mutation value  $K$  that the MCPM can tolerate.

**Step 2.** Set the initial value of  $w = 0$  and the search step length  $\Delta w = 1$ .

**Step 3.** For the given  $w$ , use the method shown in Figure 3 to optimize the driving force. After optimization, the maximum driving force  $H$  and maximum force mutation  $U$  are obtained.

**Step 4.** If  $U > K$ , set  $w = w + \Delta w$  and go to step 3; else, output the  $w$  value and  $H$  value and end the program.

An appropriate penalty coefficient can be obtained after calculation by this algorithm. At this time, the maximum driving force value is minimized while the force mutation requirement is met.

## 5. Case Study

In order to verify the effectiveness of the dynamic model and improved optimization method presented in this research, two examples of MCPM ( $M = 1, N = 2$  and  $M = 2, N = 2$ ) are given in this section.



### 5.1. Inverse Kinematics and Dynamics Simulation

This section uses an example of MCPM ( $M = 1, N = 2$ ) to prove the correctness of the dynamics modeling. Firstly, using ADAMS software to establish the dynamics model. Then, the distribution results of limb driving force in ADAMS software are obtained through the co-simulation of MATLAB and ADAMS. Finally, the simulation results obtained by two different software are the same, which verifies the correctness of the dynamic modeling.

#### 5.1.1. ADAMS Model

ADAMS software is used to build a dynamics model of the MCPM ( $M = 1, N = 2$ ). The dynamics model is depicted in Figure 4. The geometrical and inertia parameters of the module unit are presented in Table 1. The geometrical and inertia parameters of the MCPM are presented in Table 2.

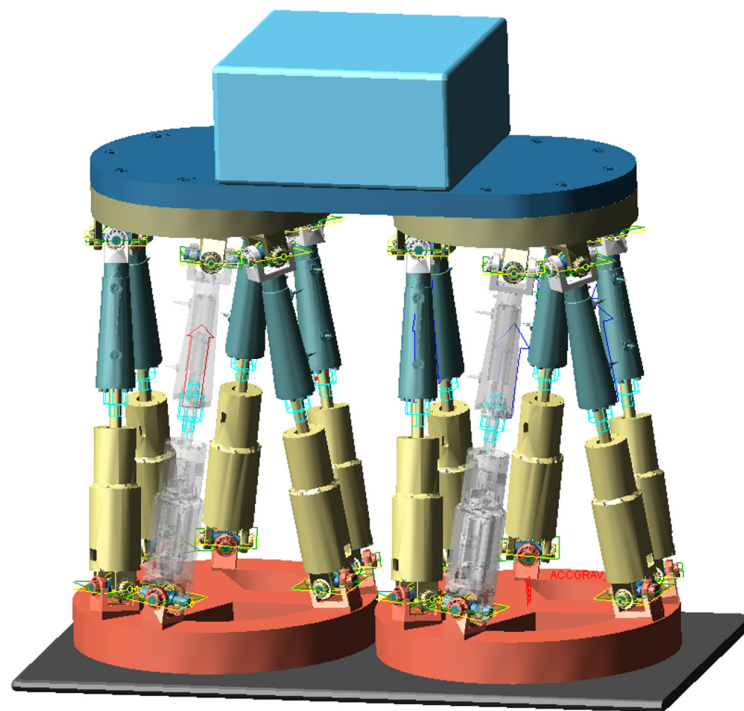


Figure 4. Dynamics model in ADAMS.

Table 1. Module unit parameters.

| Parameters                       | Value                                                                                                       | Parameters                       | Value                                                                                                       |
|----------------------------------|-------------------------------------------------------------------------------------------------------------|----------------------------------|-------------------------------------------------------------------------------------------------------------|
| $o_{ij}a_{ij}^{(1)}$             | $(-0.1488, -0.1416, 0.1) \text{ m}$                                                                         | $o_{ij}b_{ij}^{(1)}$             | $(-0.1530, 0.04837, 0.6651) \text{ m}$                                                                      |
| $o_{ij}a_{ij}^{(2)}$             | $(-0.04818, -0.1997, 0.1) \text{ m}$                                                                        | $o_{ij}b_{ij}^{(2)}$             | $(0.3463, -0.1567, 0.6651) \text{ m}$                                                                       |
| $o_{ij}a_{ij}^{(3)}$             | $(0.1970, -0.05810, 0.1) \text{ m}$                                                                         | $o_{ij}b_{ij}^{(3)}$             | $(0.1184, -0.1084, 0.6651) \text{ m}$                                                                       |
| $o_{ij}a_{ij}^{(4)}$             | $(0.1961, 0.05810, 0.1) \text{ m}$                                                                          | $o_{ij}b_{ij}^{(4)}$             | $(0.1184, 0.1084, 0.6651) \text{ m}$                                                                        |
| $o_{ij}a_{ij}^{(5)}$             | $(-0.04818, 0.1997, 0.1) \text{ m}$                                                                         | $o_{ij}b_{ij}^{(5)}$             | $(0.03463, 0.1567, 0.6651) \text{ m}$                                                                       |
| $o_{ij}a_{ij}^{(6)}$             | $(-0.1488, 0.1416, 0.1) \text{ m}$                                                                          | $o_{ij}b_{ij}^{(6)}$             | $(-0.1530, 0.04837, 0.6651) \text{ m}$                                                                      |
| $o_{ij}p_{ij}$                   | $(0, 0, 0.7288) \text{ m}$                                                                                  | $m_{p_{ij}}$                     | 30.68 kg                                                                                                    |
| $l_1$                            | 0.1495 m                                                                                                    | $m_{a_{ij}^{(k1)}}$              | 3.089 kg                                                                                                    |
| $l_2$                            | 0.1221 m                                                                                                    | $m_{a_{ij}^{(k2)}}$              | 1.825 kg                                                                                                    |
| $o_{ij}I_{p_{ij}}$               | $\begin{bmatrix} 0.404 & 0 & 0 \\ 0 & 0.404 & 0 \\ 0 & 0 & 0.779 \end{bmatrix} \text{ kg}\cdot\text{m}^2$   | $A_{ij}^{(k)} I_{a_{ij}^{(k1)}}$ | $\begin{bmatrix} 0.0413 & 0 & 0 \\ 0 & 0.0414 & 0 \\ 0 & 0 & 0.002 \end{bmatrix} \text{ kg}\cdot\text{m}^2$ |
| $A_{ij}^{(k)} I_{a_{ij}^{(k2)}}$ | $\begin{bmatrix} 0.0147 & 0 & 0 \\ 0 & 0.0150 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \text{ kg}\cdot\text{m}^2$ |                                  |                                                                                                             |

**Table 2.** MCPM parameters.

| Parameters           | Value                                                                                                    | Parameters           | Value                                                                                                    |
|----------------------|----------------------------------------------------------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------|
| $m_z$                | 1000 kg                                                                                                  | $m_p$                | 103.5 kg                                                                                                 |
| $P_z$                | (0, 0, 0.9776) m                                                                                         | $P$                  | (0, 0, 0.7776) m                                                                                         |
| $g$                  | (0, 0, −10) N/kg                                                                                         | $I_P$                | $\begin{bmatrix} 1.26 & 0 & 0 \\ 0 & 6.08 & 0 \\ 0 & 0 & 7.31 \end{bmatrix} \text{ kg} \cdot \text{m}^2$ |
| $ p_{(i-1)j}p_{ij} $ | 0.51 m                                                                                                   | $ p_{i(j-1)}p_{ij} $ | 0.51 m                                                                                                   |
| $I_z$                | $\begin{bmatrix} 43.4 & 0 & 0 \\ 0 & 36.5 & 0 \\ 0 & 0 & 58.3 \end{bmatrix} \text{ kg} \cdot \text{m}^2$ |                      |                                                                                                          |

### 5.1.2. Co-Simulation

MATLAB and ADAMS software are used for co-simulation. the simulation flow chart is depicted in Figure 5. In ADAMS, the limb of module unit 11 is driven by force, and the limb of module unit 12 is driven by motion. The penalty coefficient  $w$  is set to 500, and the given trajectory of the integrated moving platform is defined by Equation (27), where  $x_p$ ,  $y_p$ , and  $z_p$  are the translational motion in frame  $\{O\}$  with time,  $\varphi_{x_p}$ ,  $\varphi_{y_p}$ ,  $\varphi_{z_p}$  are the Tait–Bryan angles extrinsic rotations motion in frame  $\{O\}$  with time. The rotation is defined as follows: a rotation of  $\varphi_{x_p}$  about the fixed X axis followed by a rotation of  $\varphi_{y_p}$  about the fixed Y axis and a rotation of  $\varphi_{z_p}$  about the fixed Z axis. The given position and attitude are illustrated in Figure 6A,C. The position and attitude results of the ADAMS simulation output are illustrated in Figure 6B,D. Based on Figure 6, the given trajectory and ADAMS outputs trajectory are almost identical. This finding confirms that the inverse kinematics is correct. The MATLAB and ADAMS simulation results of the limb-driving force of the module unit 12 are depicted in Figure 7. Based on Figure 7, the output driving force of each limb is identical from different software, verifying correct inverse dynamics.

$$\begin{cases} x_p = 0.010 \times \sin(0.1 \times t) & (\text{m}) \\ y_p = 0.007 \times \sin(0.1 \times t) & (\text{m}) \\ z_p = 0.003 \times \sin(0.1 \times t) + 0.778 & (\text{m}) \\ \varphi_{x_p} = 0.10 \times \sin(0.1 \times t) & (\text{rad}) \\ \varphi_{y_p} = 0.07 \times \sin(0.1 \times t) & (\text{rad}) \\ \varphi_{z_p} = 0.03 \times \sin(0.1 \times t) & (\text{rad}) \end{cases} \quad \begin{matrix} t \in (0, 20\pi) \\ \Delta t = 0.1(\text{s}) \end{matrix} \quad (27)$$

### 5.2. Redundant-Force-Optimization Method Simulation

This section uses the M-P, minimize-the-maximum, and improved optimization methods to calculate the driving force of MCPM ( $M = 1$ ,  $N = 2$  and  $M = 2$ ,  $N = 2$ ) in the given trajectory to confirm the superiority of the improved optimization method. The penalty coefficients of the improved optimization method are set as 0, 30, 50, 100, 200, 300, 500, and 800 for calculation. When  $w = 0$ , the improved optimization method is equal to the minimize-the-maximum method.

The driving force calculation results of the three types of optimization methods are depicted in Figures 8 and 9 (each optimization result curve takes the maximum of the absolute value of all driving forces at a discrete point-in-time of the trajectory as source data). Based on Figures 8 and 9, for each discrete point of the trajectory, the maximum driving force increases as the penalty coefficient increases, although they are larger than the minimize-the-maximum method but much smaller than the M-P method.

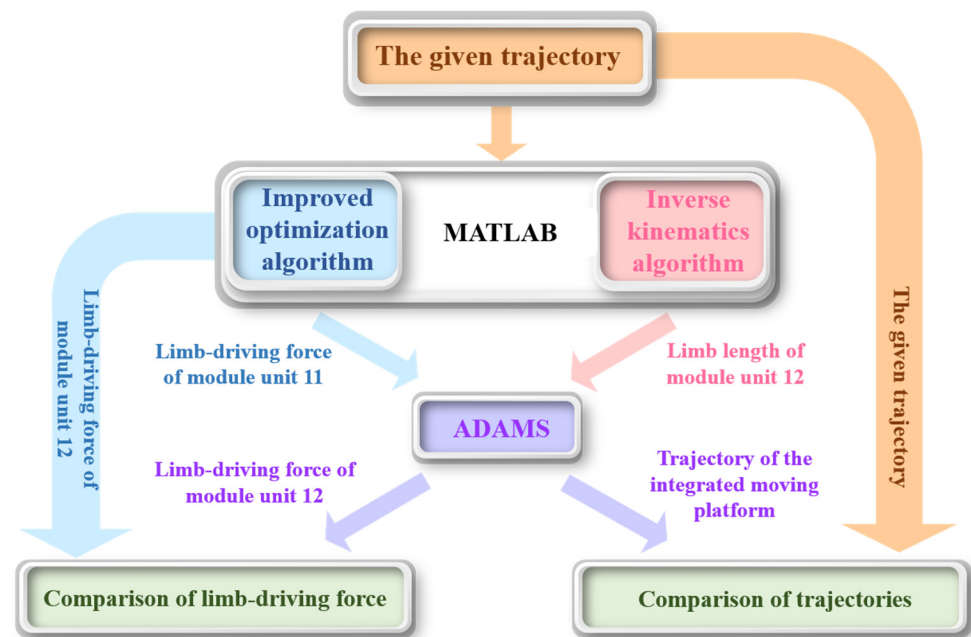


Figure 5. Co-simulation flowchart.

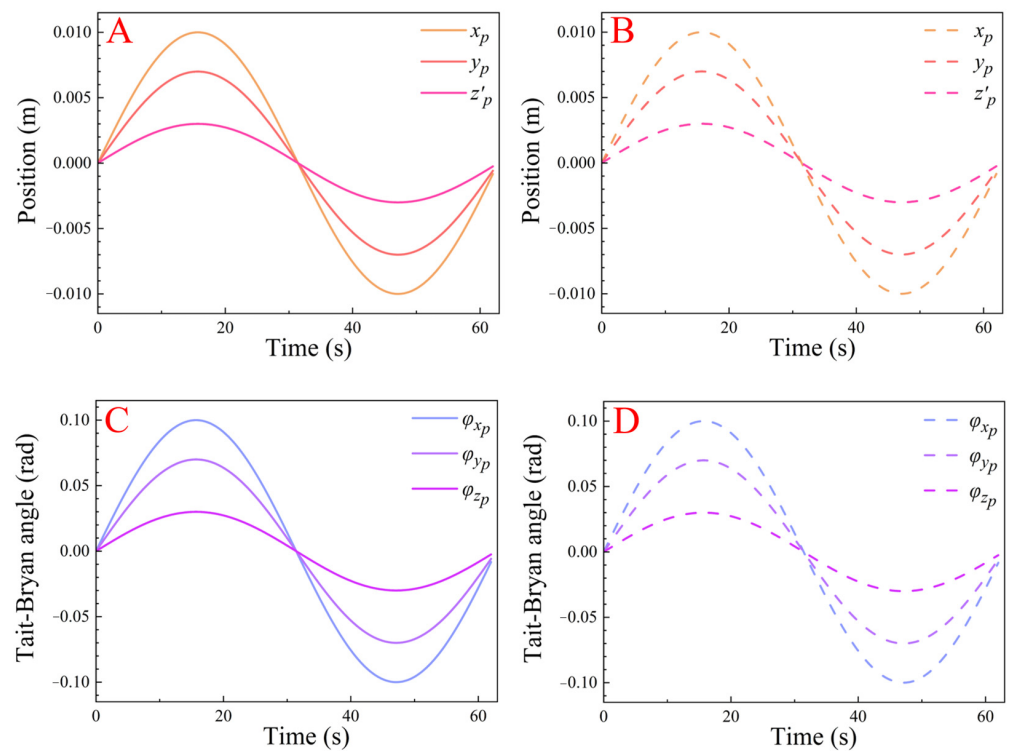
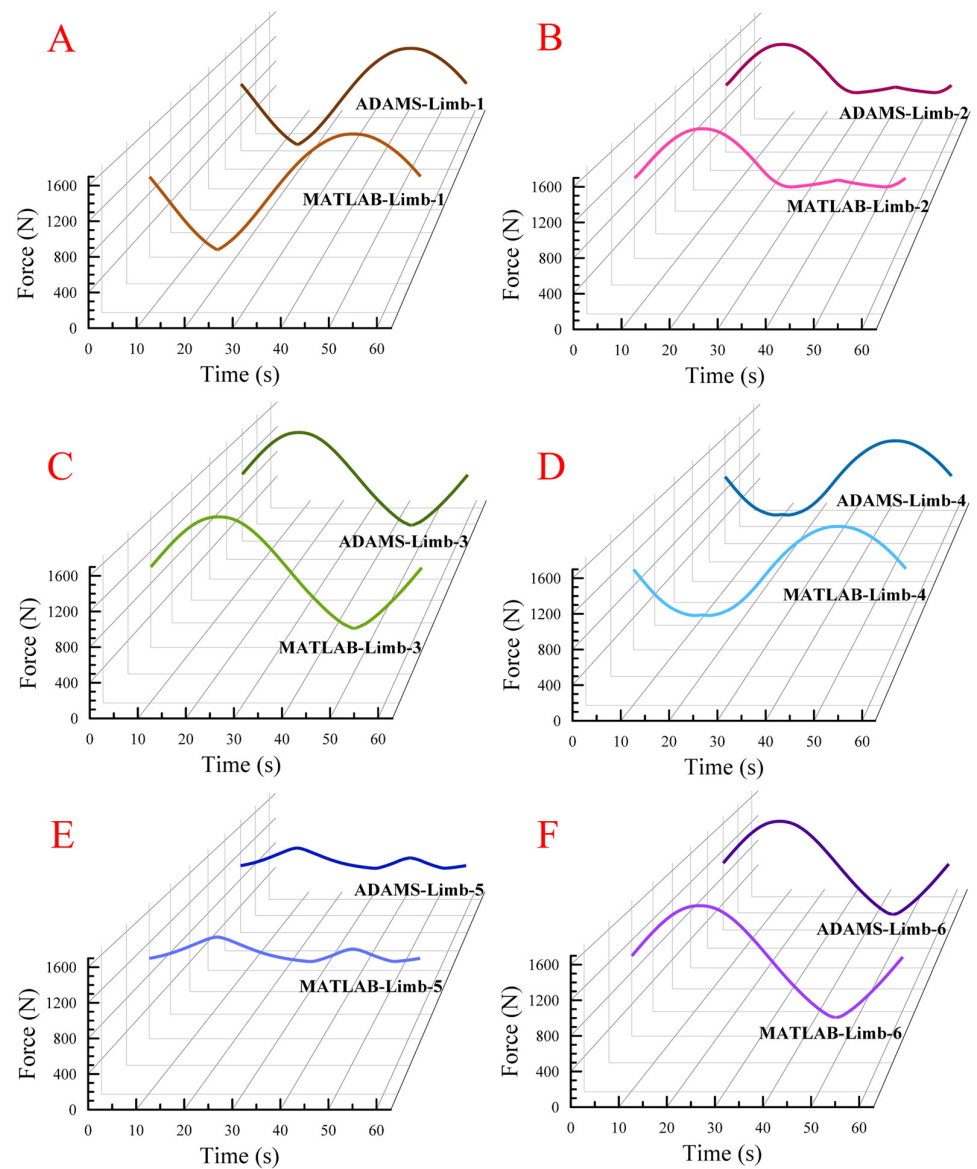
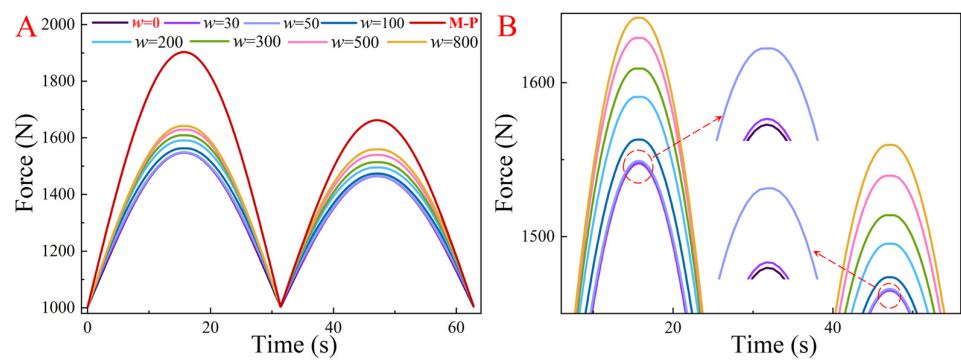


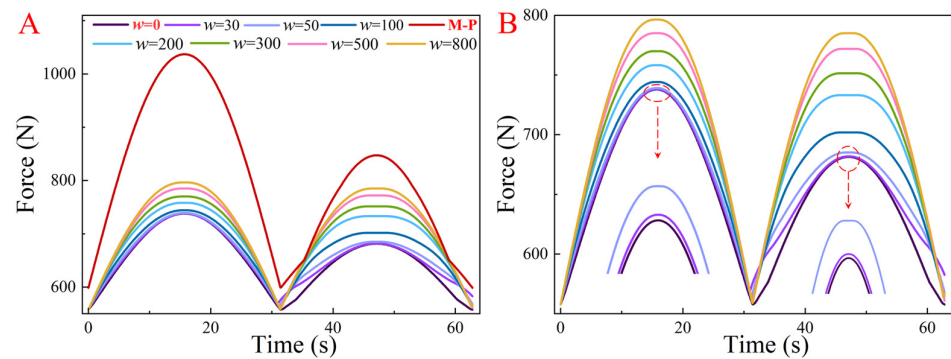
Figure 6. Comparison of trajectories: (A) given position trajectory, (B) ADAMS outputs position trajectory, (C) given attitude trajectory, and (D) ADAMS outputs attitude trajectory.



**Figure 7.** Simulation results of limb-driving force of the module unit 12 from MATLAB and ADAMS software: (A) 1 th limb, (B) 2 th limb, (C) 3 th limb, (D) 4 th limb, (E) 5 th limb and (F) 6 th limb.

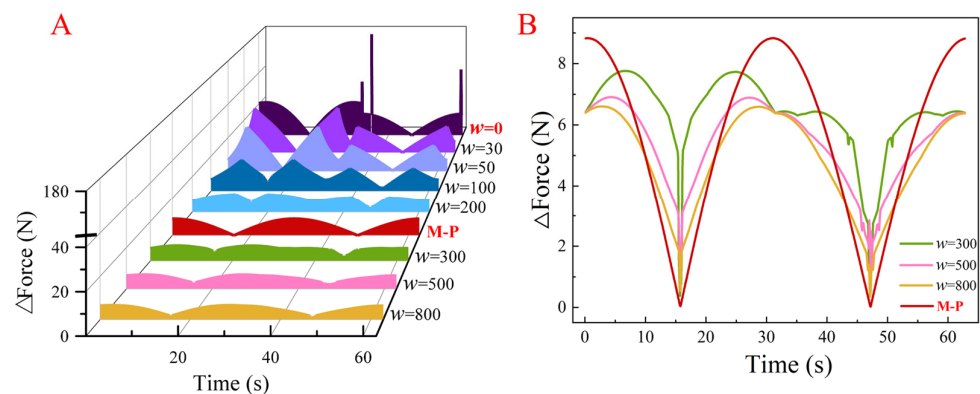


**Figure 8.** Driving force optimization results ( $M = 1$ ,  $N = 2$ ) using different optimization methods: (A) overall view and (B) partial enlarged view.

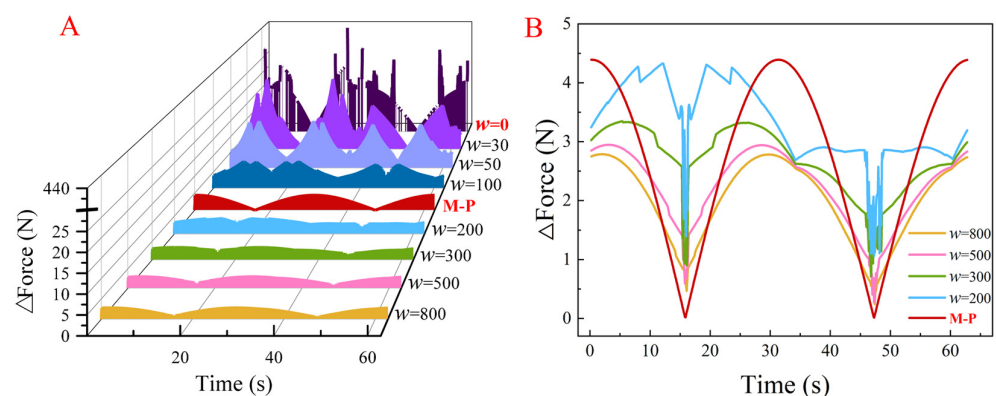


**Figure 9.** Driving force optimization results of MCPM ( $M=2, N=2$ ) using different optimization methods: (A) overall view and (B) partial enlarged view.

The force mutations of the three types of optimization methods are depicted in Figure 10 (each force mutation curve takes the maximum of the absolute value of all force mutation at a discrete point-in-time of the trajectory as source data). As depicted in Figures 10A and 11A, for each discrete point of the trajectory, the force mutation value decreases with the increase in the penalty coefficient. Moreover, they are all much smaller than the minimize-the-maximum method. As depicted in Figures 10B and 11B, the force mutation is better than the M-P method when the penalty coefficient is greater than a specific value.

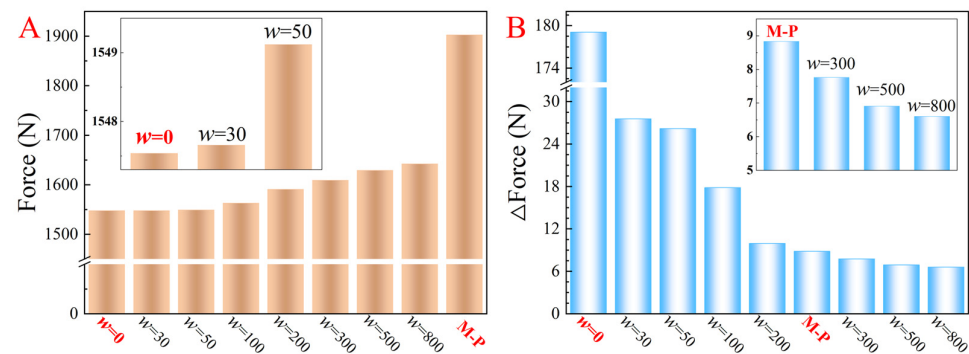


**Figure 10.** Driving force mutation of MCPM ( $M=1, N=2$ ) using different optimization methods: (A) overall 3D view and (B) comparison of the M-P method and improved method in 2D view.

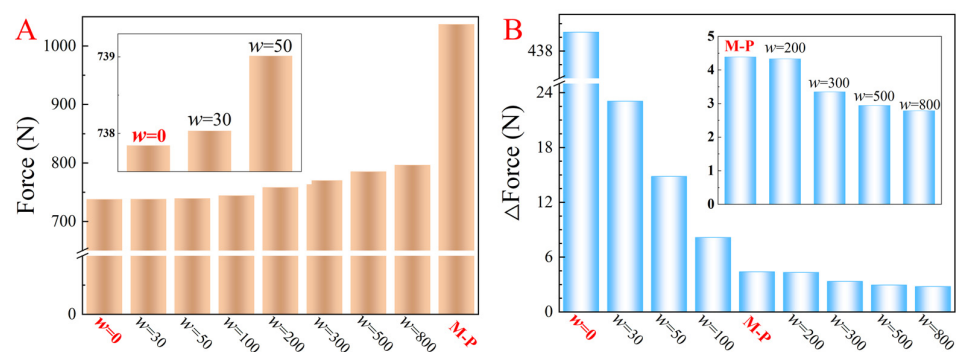


**Figure 11.** Driving force mutation of MCPM ( $M=2, N=2$ ) using different optimization methods: (A) overall 3D view and (B) comparison of the M-P method and improved method in 2D view.

The maximum driving forces and force mutations of the three types of optimization methods are depicted in Figures 12 and 13 (each column takes the maximum of the absolute value of the driving force or force mutation of the entire trajectory as source data). Based on Figures 12 and 13, for the entire trajectory, compared with the minimize-the-maximum driving force method, the maximum force value increases slightly after introducing the penalty term, but the force mutation decreases significantly. Moreover, by choosing appropriate penalty coefficients ( $w = 300, 500$ , and  $800$ ), the maximum driving force and force mutation are both smaller than the M-P method. In particular, the maximum driving force decreased significantly.



**Figure 12.** The entire trajectory's simulation results of MCPM ( $M = 1, N = 2$ ) using different optimization methods: (A) maximum driving force and (B) maximum driving force mutation.



**Figure 13.** The entire trajectory's simulation results of MCPM ( $M = 2, N = 2$ ) using different optimization methods: (A) maximum driving force and (B) maximum driving force mutation.

## 6. Conclusions

This paper investigates the driving force of the limb optimization method of the MCPM by taking force mutation into account. A novel optimization method based on the force mutation penalty term is proposed to solve the problem of force mutation. The PSO has been considered the optimization tool under multiple redundant driving forces. The performance of the proposed method is simulated with a series of  $w$  values. Compared with the minimize-the-maximum driving force method, the maximum force value increases slightly after introducing the penalty term, but the force mutation decreases significantly. Compared with the widely used M-P method, the proposed method can reduce both the maximum driving force and force mutation. Especially, the maximum driving force decreased significantly. The simulation results verify the effectiveness of the proposed optimization method.

However, the difficulty of MCPM motion control lies in the cooperative movement of each module unit, which belongs to the traditional closed-chain system control problem. With the module as the redundant driving unit, the control system with the proposed method will be more complex, which could be the future research direction.



**Author Contributions:** Conceptualization, H.H. and A.J.; methodology, A.J.; software, C.H.; validation, S.H.; formal analysis, A.J.; investigation, Z.X.; resources, Q.W.; data curation, C.H.; writing—original draft preparation, A.J.; writing—review and editing, H.H.; project administration, H.H.; funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Natural Science Foundation of China [Grant No. 52005478, 62235018], the Youth Innovation Promotion Association, Chinese Academy of Sciences [Grant No. 2022215] and the Jilin Scientific and Technological Development Program [Grant No. 20200404204YY].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

**Acknowledgments:** The authors would like to thank Xiaoming Wang, Ang Li, and others of SREC Department for their help in this research, and express sincere appreciation to the reviewers of this paper for their helpful comments.

**Conflicts of Interest:** No conflicts of interest exist in the submission of this manuscript, and the manuscript is approved by all authors for publication.

## Appendix A

- $a_{ij}^{(k)}$  = center of the joint connecting the base and the  $k$  th limb;
- $b_{ij}^{(k)}$  = center of the joint connecting the moving platform and the  $k$ th limb;
- $v_p$  = velocity of integrated moving platform in frame  $\{O\}$ ;
- $\dot{v}_p$  = acceleration of integrated moving platform in frame  $\{O\}$ ;
- $\ddot{v}_z$  = load's centroid acceleration in frame  $\{O\}$ ;
- $\omega_p$  = angular velocity of integrated moving platform in frame  $\{O\}$ ;
- $\dot{\omega}_p$  = angular acceleration of integrated moving platform in frame  $\{O\}$ ;
- $\dot{x}_p = \begin{bmatrix} v_p \\ \omega_p \end{bmatrix}$  generalized velocity of integrated moving platform frame  $\{O\}$ ;
- $v_{p_{ij}}$  = moving platform velocity of the module unit  $ij$  in frame  $\{O\}$ ;
- ${}^{O_{ij}}v_{p_{ij}}$  = moving platform velocity of the module unit  $ij$  in frame  $\{O_{ij}\}$ ;
- ${}^{O_{ij}}\omega_{p_{ij}} = \omega_p$  angular velocity of moving platform of module unit  $ij$  in frame  $\{O_{ij}\}$ ;
- $\dot{{}^{O_{ij}}x_{p_{ij}}} = \begin{bmatrix} {}^{O_{ij}}v_{p_{ij}} \\ {}^{O_{ij}}\omega_{p_{ij}} \end{bmatrix}$  moving platform generalized velocity in frame  $\{O_{ij}\}$ ;
- $\tilde{PP}_{ij}$  = skew symmetric matrix of vector  $PP_{ij}$ ;
- ${}^{P_{ij}}p_{ij}b_{ij}^{(k)}$  = position vector in frame  $\{P_{ij}\}$ ;
- ${}^{O_{ij}}o_{ij}p_{ij}, {}^{O_{ij}}o_{ij}a_{ij}^{(k)}, {}^{O_{ij}}P_{ij}b_{ij}^{(k)}, {}^{O_{ij}}d_{ij}^{(k)}$  = position vector in frame  $\{O_{ij}\}$ ;
- ${}^{O_{ij}}P_{ij}b_{ij}^{(k)}$  = skew symmetric matrix of vector  ${}^{O_{ij}}P_{ij}b_{ij}^{(k)}$ ;
- ${}^{O_{ij}}v_{b_{ij}^{(k)}}$  = velocity of joint  $b_{ij}^{(k)}$  in frame  $\{O_{ij}\}$ ;
- ${}^{A_{ij}^{(k)}}v_{b_{ij}^{(k)}}$  = velocity of joint  $b_{ij}^{(k)}$  in frame  $\{A_{ij}^{(k)}\}$ ;
- $\tilde{A_{ij}^{(k)}}S$  = skew symmetric matrix of vector  $A_{ij}^{(k)}s$ ;
- $\dot{d}_{a_{ij}^{(k)}}$  = derivative of  $d_{a_{ij}^{(k)}}$ ;
- $g$  = vector of gravity;

$F_z$  = resultant of applied and inertia wrenches exerted at the load's centroid;  
 $F_p$  = resultant of applied and inertia wrenches exerted at integrated moving platform's centroid;  
 $I_z$  = inertia matrix of load relative to its center in frame  $\{O\}$ ;  
 $I_p$  = inertia matrix of integrated moving platform relative to its center in frame  $\{O\}$ ;  
 ${}^{O_{ij}}F_{p_{ij}}$  = resultant of applied and inertia wrenches exerted at the center of moving platform of module unit  $ij$  in frame  $\{O_{ij}\}$ ;  
 ${}^{A_{ij}^{(k)}}F_{a_{ij}^{(k1)}}$  = resultant of applied and inertia wrenches exerted at the  $k$ th cylinder's centroid of module unit  $ij$  in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{A_{ij}^{(k)}}F_{a_{ij}^{(k2)}}$  = resultant of applied and inertia wrenches exerted at the  $k$ th piston's centroid of module unit  $ij$  in frame  $\{A_{ij}^{(k)}\}$ ;  
 $m_{p_{ij}}, m_{a_{ij}^{(k1)}}, m_{a_{ij}^{(k2)}}$  = mass of moving platform, cylinder, and piston of module unit  $ij$ ;  
 ${}^{O_{ij}}\dot{v}_{p_{ij}}, {}^{A_{ij}^{(k)}}\dot{v}_{a_{ij}^{(k1)}}, {}^{A_{ij}^{(k)}}\dot{v}_{a_{ij}^{(k2)}}$  = centroid acceleration of moving platform in frame  $\{O_{ij}\}$ , cylinder in frame  $\{A_{ij}^{(k)}\}$ , and piston in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{A_{ij}^{(k)}}\dot{\omega}$  = the angular acceleration of the  $k$ th limb in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{O_{ij}}I_{p_{ij}}$  = inertia matrix of moving platform relative to its center in frame  $\{O_{ij}\}$ ;  
 ${}^{A_{ij}^{(k)}}I_{a_{ij}^{(k1)}}, {}^{A_{ij}^{(k)}}I_{a_{ij}^{(k2)}}$  = inertia matrix of cylinder and piston relative to its centroid in frame  $\{A_{ij}^{(k)}\}$ ;  
 $\delta x_z$  = virtual displacement of the load in frame  $\{O\}$ ;  
 $\delta x_p$  = virtual displacement of the integrated moving platform in frame  $\{O\}$ ;  
 ${}^{O_{ij}}\delta x_{p_{ij}}$  = virtual displacement of moving platform of module unit  $ij$  in frame  $\{O_{ij}\}$ ;  
 ${}^{A_{ij}^{(k)}}\tau$  = driving force of  $k$ th limb in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{A_{ij}^{(k)}}\delta x_{b_{ij}^{(k)}}$  = virtual displacement of  $k$ th limb in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{A_{ij}^{(k)}}\delta x_{a_{ij}^{(k1)}}$  = virtual displacement of  $k$ th cylinder in frame  $\{A_{ij}^{(k)}\}$ ;  
 ${}^{A_{ij}^{(k)}}\delta x_{a_{ij}^{(k2)}}$  = virtual displacement of  $k$ th piston in frame  $\{A_{ij}^{(k)}\}$ ;  
 $E_{3 \times 3}, O_{3 \times 3}$  and  $O_{3 \times 6} = 3 \times 3$  identity matrix,  $3 \times 6$  zero matrix and  $3 \times 6$  zero matrix;  
 $T_{ij} = \begin{bmatrix} P_{ij} J \\ O_{3 \times 3} \quad E_{3 \times 3} \end{bmatrix}$  virtual displacement transformation matrix of  $\delta x_p$  relative to  ${}^{O_{ij}}\delta x_{p_{ij}}$ ;  
 $T_{ij}^{(k)} = \begin{bmatrix} b_{ij}^{(k)} J \\ O_{3 \times 6} \end{bmatrix}$  virtual displacement transformation matrix of  ${}^{O_{ij}}\delta x_{p_{ij}}$  relative to  ${}^{A_{ij}^{(k)}}\delta x_{b_{ij}^{(k)}}$ ;  
 $T_{ij}^{(k1)} = \begin{bmatrix} a_{ij}^{(k1)} J \\ P_{ij} J \\ a_{ij}^{(k1)} J \\ P_{ij} J \end{bmatrix}$  virtual displacement transformation matrix of  ${}^{O_{ij}}\delta x_{p_{ij}}$  relative to  ${}^{A_{ij}^{(k)}}\delta x_{a_{ij}^{(k1)}}$ ;  
 $T_{ij}^{(k2)} = \begin{bmatrix} a_{ij}^{(k2)} J \\ P_{ij} J \\ a_{ij}^{(k2)} J \\ P_{ij} J \end{bmatrix}$  virtual displacement transformation matrix of  ${}^{O_{ij}}\delta x_{p_{ij}}$  relative to  ${}^{A_{ij}^{(k)}}\delta x_{a_{ij}^{(k2)}}$ ;



$[0 \ 0 \ C_2 \ 0 \ 0 \ 0]^T = -J^+(J_z F_z + F_H + J_1 F_1 + J_2 F_2)$ —the M-P solution, where  $J^+$  is the M-P inverse matrix of  $J$ .

## References

1. Tsai, L.-W. Solving the inverse dynamics of a Stewart-Gough manipulator by the principle of virtual work. *J. Mech. Des.* **2000**, *122*, 3–9. [\[CrossRef\]](#)
2. Yang, C.; Ye, W.; Li, Q. Review of the performance optimization of parallel manipulators. *Mech. Mach. Theory* **2022**, *170*, 104725. [\[CrossRef\]](#)
3. Wang, D.; Wang, L.; Wu, J. Physics-based mechatronics modeling and application of an industrial-grade parallel tool head. *Mech. Syst. Signal Process.* **2021**, *148*, 107158. [\[CrossRef\]](#)
4. Bourbonnais, F.; Bigras, P. Minimum-time trajectory planning and control of a pick-and-place five-bar parallel robot. *IEEE/ASME Trans. Mechatron.* **2014**, *20*, 740–749. [\[CrossRef\]](#)
5. Arrouk, K.A.; Bouzgarrou, B.C.; Gogu, G. CAD Based Techniques for Workspace Analysis and Representation of the 3CRS Parallel Manipulator. In Proceedings of the 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010), Budapest, Hungary, 24–26 June 2010; pp. 155–160. [\[CrossRef\]](#)
6. Tang, T.; Zhang, J. Conceptual design and kinetostatic analysis of a modular parallel kinematic machine-based hybrid machine tool for large aeronautic components. *Robot. Comput. -Integr. Manuf.* **2019**, *57*, 1–16. [\[CrossRef\]](#)
7. Zhang, J.; Liu, C.; Liu, T.; Qi, K.; Niu, J.; Guo, S. Module combination based configuration synthesis and kinematic analysis of generalized spherical parallel mechanism for ankle rehabilitation. *Mech. Mach. Theory* **2021**, *166*, 104436. [\[CrossRef\]](#)
8. Hao, G.; Kong, X. Design and Modeling of a Large-Range Modular XYZ Compliant Parallel Manipulator Using Identical Spatial Modules. *J. Mech. Robot.* **2012**, *4*, 021009. [\[CrossRef\]](#)
9. Wu, J.; Wang, J.; Wang, L. A comparison study of two planar 2-DOF parallel mechanisms: One with 2-RRR and the other with 3-RRR structures. *Robotica* **2010**, *28*, 937–942. [\[CrossRef\]](#)
10. Qing, J.; Li, J.; Fang, B. Drive optimization of Tricept parallel mechanism with redundant actuation. *J. Mech. Eng.* **2010**, *46*, 8–14. [\[CrossRef\]](#)
11. Yao, J.; Gu, W.; Feng, Z.; Chen, L.; Xu, Y.; Zhao, Y. Dynamic analysis and driving force optimization of a 5-DOF parallel manipulator with redundant actuation. *Robot. Comput. -Integr. Manuf.* **2017**, *48*, 51–58. [\[CrossRef\]](#)
12. Zhao, Y.; Gao, F. Dynamic performance comparison of the 8PSS redundant parallel manipulator and its non-redundant counterpart—The 6PSS parallel manipulator. *Mech. Mach. Theory* **2009**, *44*, 991–1008. [\[CrossRef\]](#)
13. Yan, P.; Huang, H.; Li, B.; Zhou, D. A 5-DOF redundantly actuated parallel mechanism for large tilting five-face machining. *Mech. Mach. Theory* **2022**, *172*, 104785. [\[CrossRef\]](#)
14. Xu, Y.; Yao, J.; Zhao, Y. Inverse dynamics and internal forces of the redundantly actuated parallel manipulators. *Mech. Mach. Theory* **2012**, *51*, 172–184. [\[CrossRef\]](#)
15. Xie, S.; Hu, K.; Liu, H.; Wan, Y. Dynamic modeling and performance analysis of a new redundant parallel rehabilitation robot. *IEEE Access* **2020**, *8*, 222211–222225. [\[CrossRef\]](#)
16. Lee, G.; Sul, S.-K.; Kim, J. Energy-saving method of parallel mechanism by redundant actuation. *Int. J. Precis. Eng. Manuf. -Green Technol.* **2015**, *2*, 345–351. [\[CrossRef\]](#)
17. Jiang, Y.; Li, T.-M.; Wang, L.-P. Dynamic modeling and redundant force optimization of a 2-DOF parallel kinematic machine with kinematic redundancy. *Robot. Comput. -Integr. Manuf.* **2015**, *32*, 1–10. [\[CrossRef\]](#)
18. Hino, T.; Kawanishi, M.; Kanki, H.; Narikiyo, T. Control of parallel mechanism with redundant actuators and non-redundant actuators. In Proceedings of the ICMIT 2007: Mechatronics, MEMS, and Smart Materials, Gifu, Japan, 9 January 2008; pp. 639–644. [\[CrossRef\]](#)
19. Wu, J.; Qiu, J.; Ye, H. Torque optimization method of a 3-DOF redundant parallel manipulator based on actuator torque range. *J. Mech. Robot.* **2022**, *15*, 021005. [\[CrossRef\]](#)
20. Liu, W.; Xu, Y.; Yao, J.; Zhao, Y. The weighted Moore–Penrose generalized inverse and the force analysis of overconstrained parallel mechanisms. *Multibody Syst. Dyn.* **2016**, *39*, 363–383. [\[CrossRef\]](#)
21. Notash, L. On the minimum 2-Norm positive tension for wire-actuated parallel manipulators. In *Computational Kinematics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 1–9.
22. Ding, J.; Liu, X.; Wang, C. Repeatability Analysis of an Overconstrained Kinematic Coupling Using a Parallel- Mechanism-Equivalent Model. *IEEE Access* **2021**, *9*, 82455–82461. [\[CrossRef\]](#)
23. Niu, X.-M.; Gao, G.-Q.; Liu, X.-J.; Bao, Z.-D. Dynamics and control of a novel 3-DOF parallel manipulator with actuation redundancy. *Int. J. Autom. Comput.* **2013**, *10*, 552–562. [\[CrossRef\]](#)
24. Wu, J.; Wang, J.; Wang, L.; Li, T. Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy. *Mech. Mach. Theory* **2009**, *44*, 835–849. [\[CrossRef\]](#)
25. Gosselin, C.; Grenier, M. On the determination of the force distribution in overconstrained cable-driven parallel mechanisms. *Meccanica* **2011**, *46*, 3–15. [\[CrossRef\]](#)
26. Hassan, M.; Khajepour, A. Minimum-Norm Solution for the Actuator Forces in Cable-Based Parallel Manipulators Based on Convex Optimization. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 1498–1503. [\[CrossRef\]](#)

27. Kucuk, S. Energy minimization for 3-RRR fully planar parallel manipulator using particle swarm optimization. *Mech. Mach. Theory* **2013**, *62*, 129–149. [[CrossRef](#)]
28. Pasand, A.M.; Naderi, D. Energy optimization of a planar parallel manipulator with kinematically redundant degrees of freedom. In Proceedings of the 2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Tehran, Iran, 13–15 February 2013; pp. 362–367. [[CrossRef](#)]
29. Wang, W.; Tang, X.; Shao, Z. Study on energy consumption and cable force optimization of cable-driven parallel mechanism in automated storage/retrieval system. In Proceedings of the 2015 Second International Conference on Soft Computing and Machine Intelligence (ISCM), Hong Kong, China, 23–24 November 2015; pp. 144–150. [[CrossRef](#)]
30. Wu, J.; Li, T.; Xu, B. Force optimization of planar 2-DOF parallel manipulators with actuation redundancy considering deformation. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2013**, *227*, 1371–1377. [[CrossRef](#)]
31. Zhang, L.; Jiang, Y.; Li, T. An articular force optimization method of parallel manipulator with actuation redundancy. In Proceedings of the 2014 International Conference on Mechanical Engineering, Automation and Control Systems (MEACS), Tomsk, Russia, 16–18 October 2014; pp. 1–6. [[CrossRef](#)]
32. Kotlarski, J.; Abdellatif, H.; Heimann, B. Improving the pose accuracy of a planar 3RRR parallel manipulator using kinematic redundancy and optimized switching patterns. In Proceedings of the 2008 IEEE international conference on robotics and automation, Pasadena, CA, USA, 19–23 May 2008; pp. 3863–3868. [[CrossRef](#)]
33. Kotlarski, J.; Do Thanh, T.; Heimann, B.; Ortmaier, T. Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 656–661. [[CrossRef](#)]
34. Zhao, Y. Dynamic optimum design of a 3U P S-P RU parallel robot. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 1–13. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.