

Research Article

Modified PSO Method for Robust Control of 3RPS Parallel Manipulators

Tung-Sheng Zhan and Chih-Cheng Kao

Department of Electrical Engineering, Kao Yuan University, No. 1821, Jhongshan Road, Lujhu Township, Kaohsiung County 821, Taiwan

Correspondence should be addressed to Chih-Cheng Kao, cckao@cc.kyu.edu.tw

Received 2 February 2010; Revised 14 July 2010; Accepted 21 September 2010

Academic Editor: Horst Ecker

Copyright © 2010 T.-S. Zhan and C.-C. Kao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an effective method to design a modified particle swarm optimization (MPSO) singularity control method for a fully parallel robot manipulator. By adopting MPSO to obtain simple and effective estimated damping values, the result automatically adjusts the damping value around a singular point and greatly improves the accuracy of system responses. This method works by damping accelerations of the end effector, so that accelerations along the degenerated directions are zero at a singular point. These velocities, however, may not be zero in some situations, in which case, fluctuations will be encountered around a singular point. To overcome this drawback, we propose a control scheme that uses both damped acceleration and damped velocity, called the hybrid damped resolved-acceleration control (HDRAC) scheme. The MPSO optimization method can immediately provide optimal damping factors when used in an online application. Our proposed approach offers such superior features as easy implementation, stable convergence characteristics, and good computational efficiency. The main advantage of the HDRAC with MPSO in the 3RPS parallel manipulator control system is that it is not necessary for the system to plan its path for avoiding the singular point; thus, the workspace can be improved. Illustrative examples are provided to show the effectiveness of this HDRAC in practical applications, and experimental results verifying the utility of the proposed control scheme are presented.

1. Introduction

Parallel robotics offers the advantages of heavy load capacity, high rigidity, and low inertia. The triaxial RPS parallel manipulator, which is equipped with revolute joints on the base platform and spherical joints on the moving platform, has been widely applied to electronic gaming machines, virtual reality movies, simulations of aircraft and vehicles, and so on. In their research on 3RPS parallel robotics, Lee and Shah [1] used loop closure equations to deduce kinematic equations for the relationship between link lengths and the position of the

moving platform. In a separate study, they also deduced the dynamic equations that included the drive forces and reacting forces of 3RPS parallel robotics in the case of inverse dynamics [2, 3]. Following their work, Fang and Huang [4] simplified the mathematic models and established the relationship among three input-output variables. Li and Xu [5, 6] deal with the kinematic characteristics in terms of workspace and dexterity vary with differences in the arrangement of actuators are investigated in detail.

Generally, there are three types of singularity of parallel mechanisms, that is, inverse, direct, and combined singularities. In this study, we deal with the direct kinematic singularities. Therefore, the author divided the procedure of finding “direct kinematic singularities” of 3RPS platform mechanism. In horizontal position of the moving platform, to begin with, he assumed the direction of the moving platform and determined its horizontal position at this moment, that is, its projection on the fixed base. Then, since the direction of the moving platform has been assumed, also, a moving platform remaining in this direction can only translate vertically, it is possible to locate the limiting position while this direction has always been kept; this is the boundary of workspace, which makes the determinant of Jacobian matrix equal to zero and thus is the “direct kinematic singularities”.

There may exist singular configurations emerging from the motions of mechanisms, and resolved-rate control and resolved-acceleration control [7–9] are the two general approaches used to control robots. In the neighborhood of singularities, very high speeds are required to produce even small changes in the position or orientation of the end effector. In this paper, singular value decomposition (SVD) is used in order to show that fluctuations around the singular point are due to unnecessary nonzero end-effector velocities along the degenerated directions when the manipulator is at a singularity. This paper proposes a hybrid damped resolved-acceleration control (HDRAC) scheme, which considers both damped acceleration and velocity as a way to remove these unnecessary velocities.

Particle swarm optimization (PSO) is a stochastic population-based optimization approach and was first published by Kennedy and Eberhart in 1995 [10, 11]. PSO has been shown to be an efficient, roust, and simple optimization algorithm. Most studies of the PSO are empirical with only a few theoretical analyses that concentrate on understanding particle trajectories. A large amount of research has been proposed to improve the performance of PSO. From these studies, much effort has been invested to obtain a better understanding of its convergence properties. These studies concentrated mostly on a better understanding of the basic PSO control parameters, namely, the acceleration coefficients, inertia weight, velocity clamping, and swarm size [12–14]. In this paper, a modified PSO algorithm is proposed to improve the searching ability and prevent from being trapped in a local optimum. The main difference of the MPSO from the PSO is its fitness function which considers the “distance” to avoid converging to a local optimum. From these empirical studies, it can be concluded that the MPSO is sensitive to control parameter choices, specifically the inertia weight, acceleration coefficients, and velocity clamping [15]. However, wrong initialization of these parameters may lead to divergent or cyclic behavior.

In this study, singular value decomposition (SVD) is used to show that the abovementioned fluctuation is due to unnecessary nonzero end-effector velocities along the degenerated directions when the manipulator is at a singularity. The hybrid damped resolved-acceleration control scheme (HDRAC), which considers both damped acceleration and velocity, is proposed to remove these unnecessary velocities. Numerical simulations and the results of experiments are compared in order to demonstrate the applicability and effectiveness of the proposed algorithms at a singularity for a 3RPS parallel manipulator.

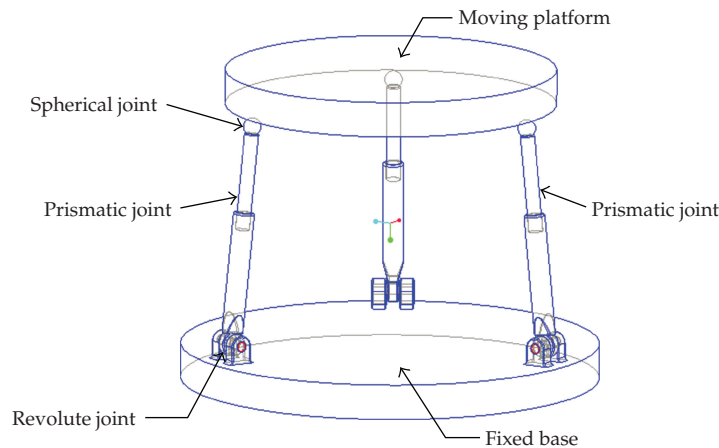


Figure 1: The 3RPS parallel manipulator.

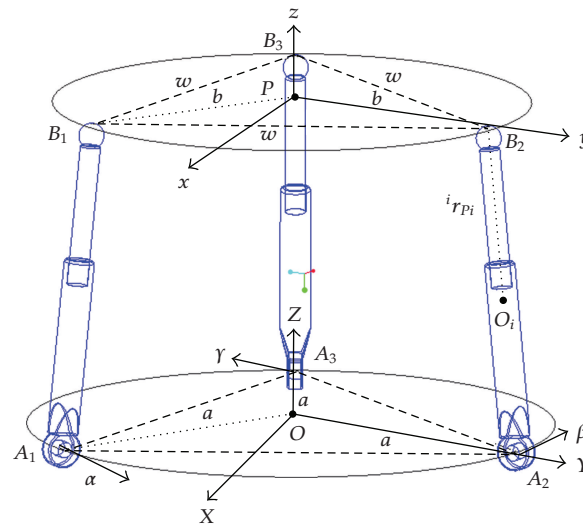


Figure 2: Coordinate systems for the 3RPS manipulator.

2. Singularity Analysis of the 3RPS Parallel Manipulator

As shown in Figure 1, one end of the 3RPS parallel manipulator is a moving platform and the other end is a fixed base. The moving platform is connected to the fixed base by three limbs via three spherical joints (B_1 , B_2 , and B_3) and three revolute joints (A_1 , A_2 , and A_3). Each limb consists of two links connected in a series by a prismatic joint. Figure 2 shows that the origin P of the coordinate system $[x \ y \ z]^T$ is located at the mass center of the base moving platform. The Cartesian coordinate vector $[X \ Y \ Z \ \alpha \ \beta \ \gamma]^T$ denotes the fixed coordinates, where $[X \ Y \ Z]^T$ and $[\alpha \ \beta \ \gamma]^T$ denote, respectively, the vectors of position and orientation. The fixed and moving platforms of the 3 RPS are circles with radius a and b , respectively. The distance between the three spherical joints on the moving platform is w .

2.1. Dynamics of the 3PRS Parallel Manipulator

Let us focus on the mass center O_i of the i th link, as shown in Figure 2. There are three components of the weight force of the moving platform that act on the i th link as ${}^iF_{pix}$, ${}^iF_{piy}$, and ${}^iF_{piz}$. The weight force of the i th link is W_i , the acting force of the separately excited DC servomotor on the revolution parts is ${}^iF_{1motor}$, and the acting force on each vertical leg is ${}^iF_{2motor}$. The 3RPS parallel manipulator dynamic motion of the links and moving platform is described by using Newton-Euler equations, which are formulated by separately considering the free-body diagrams of the links and the moving platform [16–19]. Through reduction and incorporation, we can calculate the acting force on each vertical leg by using the following force equation for the moving vertical leg [19]:

$${}^iF_{2motor} = {}^iF_{pix} + \mathbf{R}_i^T \mathbf{W}_{lix}, \quad (2.1)$$

where

$$\mathbf{R}_i = \begin{bmatrix} \cos \phi \cos \psi \cos \theta - \sin \phi \sin \psi & -\cos \phi \cos \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta \\ \sin \phi \cos \psi \cos \theta + \cos \phi \sin \psi & -\sin \psi \cos \theta \sin \phi + \cos \psi \cos \phi & \sin \phi \sin \theta \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{bmatrix}, \quad (2.2)$$

W_i is the weight of the i th link, ${}^iF_{pix} = \mathbf{D}^{-1} \mathbf{Q}$, and matrices \mathbf{D} and \mathbf{Q} are given in Appendix A.

With regard to the revolution parts, we have the equation for the first motor as follows:

$$l_{a1} \dot{i}_{a1} + r_{a1} i_{a1} = V_T - k_{b1} \omega, \quad \tau = K_{t1} i_{a1}, \quad (2.3)$$

where K_{t1} is the torque coefficient, K_{b1} is the back emf of the motor, l_{a1} is the armature inductance, r_{a1} is the armature resistance, V_T is the terminal voltage, ω is the rotor speed, and i_{a1} is the armature current.

2.2. Deduction of Direct Kinetic Singularities

In this section, we adopt a method [16–19] to formulate the direct singular position of the 3RPS parallel manipulator. The distance between any two points of B_1 , B_2 , and B_3 is always w , which gives us the following relationships:

$$\begin{aligned} f_1 &= (\overline{B_1 B_2})^2 - w^2 = 0, \\ f_2 &= (\overline{B_2 B_3})^2 - w^2 = 0, \\ f_3 &= (\overline{B_3 B_1})^2 - w^2 = 0. \end{aligned} \quad (2.4)$$

By differentiating (2.4) with respect to time, and writing them in matrix form, we get

$$\begin{Bmatrix} \frac{df_1}{dt} \\ \frac{df_2}{dt} \\ \frac{df_3}{dt} \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 0 & a_{22} & a_{23} \\ a_{31} & 0 & a_{33} \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} + \begin{bmatrix} b_{11} & b_{12} & 0 \\ 0 & b_{22} & b_{23} \\ b_{31} & 0 & b_{33} \end{bmatrix} \begin{Bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{Bmatrix} = \mathbf{A} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} + \mathbf{B} \begin{Bmatrix} \dot{d}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}, \quad (2.5)$$

where matrix \mathbf{A} and \mathbf{B} are given in Appendix B.

If the lengths of the three limbs are represented by d_i ($i = 1, 2, 3$) and the link's length is divided into two components, which are parallel and normal to the base and represented by d_{ip} and d_{in} , respectively, then we have

$$\begin{aligned} (d_{1p}, d_{2p}, d_{3p}) &= (d_1 \cos \theta_1, d_2 \cos \theta_2, d_3 \cos \theta_3), \\ (d_{1n}, d_{2n}, d_{3n}) &= (d_1 \sin \theta_1, d_2 \sin \theta_2, d_3 \sin \theta_3), \end{aligned} \quad (2.6)$$

where θ_1 , θ_2 , and θ_3 are the rotating angles of the revolute joint on the first, second, and third limbs, respectively.

This study begins with projecting the moving platform onto the fixed base, where the projections of B_1 , B_2 , and B_3 are represented by B'_1 , B'_2 , and B'_3 , respectively. The length $A_1B'_1$ is $d_{1p} = d_1 \cos(180 - \theta_1)$, and the lengths OB'_1 , OB'_2 , and OB'_3 are represented by e_1 , e_2 , and e_3 , respectively, which gives us

$$(e_1, e_2, e_3) = (d_{1p} + 1, d_{2p} + 1, d_{3p} + 1). \quad (2.7)$$

For convenience of calculation, another coordinate system (h_1, h_2, h_3) is defined as follows [19]:

$$(h_1, h_2, h_3) = (d_{3n} - d_{2n}, d_{1n} - d_{3n}, d_{2n} - d_{1n}). \quad (2.8)$$

We must note that this manipulator is capable of translating upward or downward without changing the direction of the moving platform. The approach proposed here is to start by assuming the direction of the moving platform, which is $d_{2n} - d_{1n}$, $d_{3n} - d_{2n}$, $d_{1n} - d_{3n}$, and equating (2.6) and (2.7) with (2.4) to solve the position of the three spherical joints, e_1 , e_2 , and e_3 . The elevation of the moving platform can be represented by the levels of three spherical joints, as follows:

$$d_{\text{ave}} = \frac{(d_{1n} + d_{2n} + d_{3n})}{3}. \quad (2.9)$$

We then need to find a value for $d_{1n} + d_{2n} + d_{3n}$ that makes the determinant of matrix \mathbf{A} equal to zero. We have $\det(\mathbf{A}) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31}$, while using

$$(d_{1n}, d_{2n}, d_{3n}) = d_{\text{ave}} + \frac{(h_2 - h_3, h_3 - h_1, h_1 - h_2)}{3}, \quad (2.10)$$

and this determinant is a cubic function of d_{ave} . Calling this function $F(d_{\text{ave}})$, our purpose is then to find d_{ave} , such that [19]

$$F(d_{\text{ave}}) = \det(\mathbf{A}) = \Omega_1 d_{\text{ave}}^3 + \Omega_2 d_{\text{ave}}^2 + \Omega_3 d_{\text{ave}} + \Omega_4 = 0, \quad (2.11)$$

where the coefficients are given in Appendix C. The process of finding a solution begins with assuming (h_1, h_2, h_3) values, and obtaining (e_1, e_2, e_3) values and coefficients of the cubic polynomial equation, so that finally the roots of d_{ave} that satisfy $\det(\mathbf{A}) = 0$ can be found.

3. Optimization Method: Modifying Particle Swarm Optimization

3.1. Particle Swarm Optimization

Birds (particles) flocking optimizes a certain objective function in a PSO system. Each agent knows its best value so far (*pbest*) and its position. This information is an analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (*gbest*) among *pbests*. This information is an analogy of knowledge of how the other agents around them have performed. The PSO concept [10, 11] consists of changing the velocity of each particle toward its *pbest* and *gbest* locations. In the PSO, each particle moves to a new position according to new velocity and the previous positions of the particle. This is compared with the best position generated by previous particles in the fitness function, and the best one is kept; so each particle accelerates in the direction of not only the local best solution but also the global best position. If a particle discovers a new probable solution, other particles will move closer to it to explore the region more completely in the process.

In general, there are three attributes, current position x_j , current velocity v_j , and past best position $pbest_j$, for particles in the search space to present their features. Each particle in the swarm is iteratively updated according to the aforementioned attributes. For example, [10–14], the j th particle is represented as $x_j = (x_{j,1}, x_{j,2}, \dots, x_{j,g})$ in the g -dimensional space. The best previous position of the j th particle is recorded and represented as $pbest_j = (pbest_{j,1}, pbest_{j,2}, \dots, pbest_{j,g})$. The index of best particle among all particles in the group is represented by the $gbest_g$. The rate of the position change (velocity) for particle j is represented as $v_j = (v_{j,1}, v_{j,2}, \dots, v_{j,g})$. The modified velocity and position of each particle can be calculated using the current velocity and distance from $pbest_{j,g}$ to $gbest_{j,g}$ as shown in the following formulas [14]:

$$\begin{aligned} v_{j,g}^{(t+1)} &= w \cdot v_{j,g}^{(t)} + c_1 \text{Rand}() \cdot (pbest_{j,g}^{(t)} - x_{j,g}^{(t)}) \\ &\quad + c_2 \text{Rand}^*() \cdot (gbest_{j,g}^{(t)} - x_{j,g}^{(t)}), \quad j = 1, 2, \dots, n; \quad g = 1, 2, \dots, m, \quad (3.1) \\ x_{j,g}^{(t+1)} &= x_{j,g}^{(t)} + v_{j,g}^{(t+1)}, \end{aligned}$$

where n is the number of particles in a group; m is the number of members in a particle; t is the pointer of iterations (generations); $v_{j,g}^{(t)}$ is the velocity of the particle j at iteration t , $V_g^{\min} \leq v_{j,g}^{(t)} \leq V_g^{\max}$; w is the inertia weighting factor; c_1, c_2 are the acceleration constants; $Rand(), Rand^*(\cdot)$ are random numbers between 0 and 1; $x_{j,g}^{(t)}$ is the current position of particle j at iteration t ; $pbest_j$ is the $pbest$ of particle j ; $gbest_g$ is the $gbest$ of the group g .

In the above procedures, the parameter V_g^{\max} determines the resolution or fitness, with which regions are searched between the present position and the target position. If V_g^{\max} is too high, particles might fly past good solutions. If V_g^{\max} is too low, particles may not explore sufficiently beyond local solutions.

The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward $pbest$ and $gbest$ positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in an abrupt movement toward or past target regions.

Suitable selection of inertia weighting factor w provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run. In general, the inertia weighting factor w is set according to the following equation [11, 12]:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times \text{iter}, \quad (3.2)$$

where iter_{\max} is the maximum number of iterations (generations), and iter is the current number of iterations.

3.2. Modified Particle Swarm Optimization

The main point of the MPSO differs from the PSO to consider the "distance" in its fitness function to avoid converging to a local optimum. Assign a rank (i.e., the number place 1, 2, 3, ..., etc.) RE_k to the calculated error of each new individual, v_k , $k = 1, \dots, PS$, PS is the population size. A combined population with $2 \times PS$ individuals is formed. Unlike previously developed statistic methods, the concept of "distance" is added to the fitness function to prevent from being trapped in a local minimum. The fitness score of the k th individual is modified by [20, 21]

$$F_k = RE_k + \rho \times RD_k, \quad k = 1, \dots, 2 \times PS, \quad (3.3)$$

where ρ is an adaptive decay scale, ρ_{\max} is set as 0.7, and ρ_{\min} is set as 0.005 in this paper. RD_k is the rank of D_k assigned to the k th individual, where D_k is the distance from the individual to the current best solution vector and is given by

$$D_k = \|v_k - v_{best}\|, \quad (3.4)$$

where v_k is the vector of the k th individual in the combined population, and v_{best} is the current best solution vector.

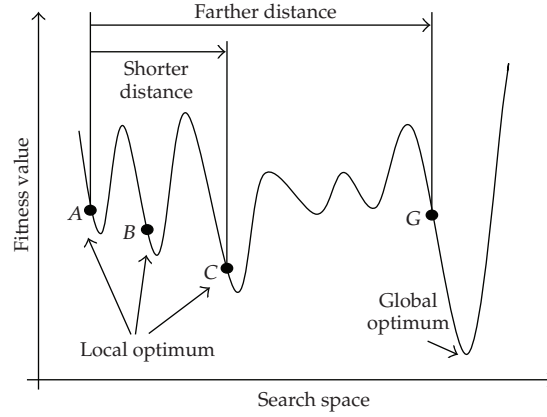


Figure 3: The concept of distances.

An adaptive scheme is defined as [20]

$$\rho_{\Delta} = R \times \frac{(\rho_{\max} - \rho_{\min})}{g_{\max}},$$

$$\rho(g+1) = \begin{cases} \rho(g) - \rho_{\Delta}, & F_{\min}(g) = F_{\min}(g-1), \\ \rho(g), & F_{\min}(g) < F_{\min}(g-1), \end{cases} \quad (3.5)$$

$$\rho(g+1) = \rho_{\min} \quad \text{if } \rho(g) - \rho_{\Delta} < \rho_{\min},$$

where ρ_{Δ} is the step size; F_{\min} is the minimum value of fitness functions; R is the regulating scale and is set as 1.25 in this paper, and g_{\max} is the maximum allowable number of iterations.

Individuals will be ranked ascending according to their fitness scores by a sorting algorithm. The PS individuals are transcribed along with their fitness for the next generation. If the new population does not include the current best solution, the best solution must be replaced with the last individual in the new population. In addition, a gradually decreased decay scale can satisfy a successive statistic searching process by first using the diversification (bigger ρ) to explore more regions and then the intensification (smaller ρ) to exploit the neighborhood of an elite solution. The current best solution (point A) for a minimum fitness problem as shown in the Figure 3 may not reach the global optimum [20, 21], and there are three electable solutions. Generally, solutions with slightly better fitness (point C or B) prevailed, so the solution was trapped into the valley prematurely. The more attractive solution (point G) is relatively far away from point A, but it nears the global optimal. To prevent prematurity, point G with slightly worse fitness than C, it needs a higher rank to be selected. That is, a higher RD_k is awarded to a longer D_k .

Stopping Criteria

Stopping criteria is given in the following order:

- (1) maximum allowable number of iterations reached.
- (2) number of iterations reached without improving the current best solution.

Figure 4 shows the flow chart of the proposed algorithm.

4. Singularity Control Scheme

4.1. Resolved-Acceleration Control

The relationship between the velocities of the mass center P of the moving platform and the joint velocity $\dot{\mathbf{q}}$ for parallel manipulators can be represented as

$$\dot{\mathbf{x}} \equiv \begin{bmatrix} \dot{r} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}\dot{\mathbf{q}}, \quad \mathbf{q} = \begin{bmatrix} X \\ Y \\ Z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}, \quad \dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \begin{bmatrix} v_X \\ v_Y \\ v_Z \\ \omega_\alpha \\ \omega_\beta \\ \omega_\gamma \end{bmatrix}, \quad (4.1)$$

where \dot{r} and $\boldsymbol{\omega}$ are the linear and angular velocities, respectively. The Cartesian coordinate vector $\mathbf{q} = (X, Y, Z, \alpha, \beta, \gamma)^T$ with six variables is chosen to describe the position and orientation of the moving base, that is, $\dot{\mathbf{q}}$ is the velocity and \mathbf{J} is the Jacobian matrix. Differentiating (4.1) gives

$$\mathbf{a} \equiv \begin{bmatrix} \ddot{r} \\ \boldsymbol{\alpha} \end{bmatrix} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}, \quad (4.2)$$

where \ddot{r} and $\boldsymbol{\alpha}$ are the linear and angular accelerations, respectively.

Now, by expanding (2.1)-(2.3), we can obtain the dynamic equation of the manipulator in the matrix form of

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\dot{\mathbf{q}}, \mathbf{q}) + \mathbf{G}(\mathbf{q}) = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}, \mathbf{q}), \quad (4.3)$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix of the manipulator, $\mathbf{N}(\dot{\mathbf{q}}, \mathbf{q})$ contains the Coriolis and centrifugal force/torque components, $\mathbf{G}(\mathbf{q})$ contains the gravity force/torque components, and $\boldsymbol{\tau}$ is the vector of actuator forces. The second-order nonlinear coupled dynamic equation (4.3) can be linearized by inputting the inverse dynamics as follows:

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{f}(\dot{\mathbf{q}}, \mathbf{q}), \quad (4.4)$$

where $\ddot{\mathbf{q}}_d$ is the vector of the desired accelerations, so that $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_d$.

The resolved-acceleration control scheme [19] is

$$\ddot{\mathbf{q}}_d = \mathbf{J}^{-1}(\mathbf{a}_d - \dot{\mathbf{J}}\dot{\mathbf{q}}) = \mathbf{J}^{-1}\mathbf{a}_d^*, \quad (4.5a)$$

with

$$\mathbf{a}_d = \begin{bmatrix} \ddot{r}_d \\ \boldsymbol{\alpha}_d \end{bmatrix} + \mathbf{K}_D \begin{bmatrix} \dot{r}_d - \dot{r} \\ \boldsymbol{\omega}_d - \boldsymbol{\omega} \end{bmatrix} + \mathbf{K}_P \begin{bmatrix} \boldsymbol{\varepsilon}_r \\ \boldsymbol{\varepsilon}_e \end{bmatrix}, \quad \mathbf{a}_d^* = (\mathbf{a}_d - \dot{\mathbf{J}}\dot{\mathbf{q}}), \quad (4.5b)$$

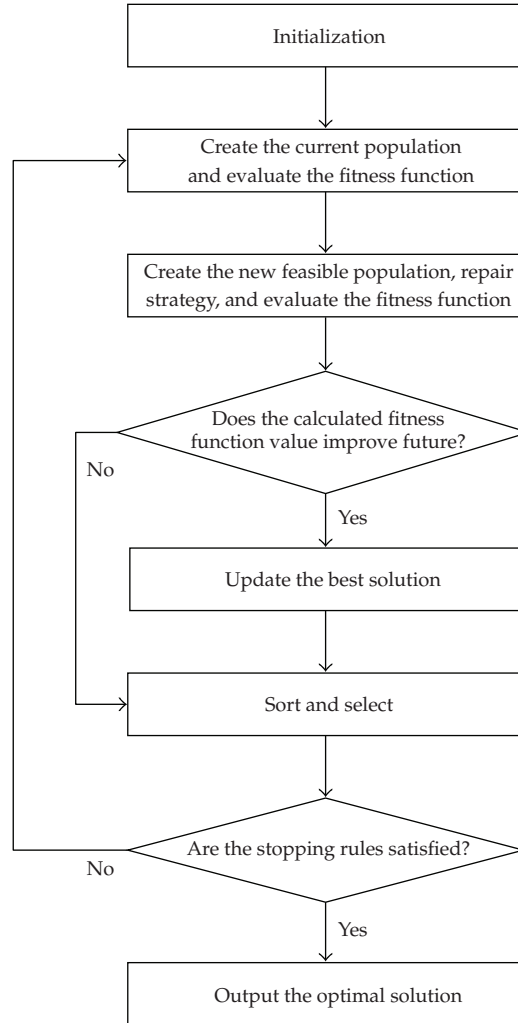


Figure 4: The flow chart of the MPSO.

where \mathbf{K}_D and \mathbf{K}_P are gain matrices, subscript d denotes the desired value, $\varepsilon_r = \varepsilon_d - r$ is the positional error, and ε_e is the orientation error. Unfortunately, this control scheme (4.5a) breaks down when \mathbf{J}^{-1} does not exist, which occurs at a singular configuration.

4.2. Damped Least-Square Method

In this section, the damped least-square method for the inverse problem of (4.1) is applied to solve the following optimization problem [19]:

$$\min_{\dot{q}_\rho} \left(\|\mathbf{J}\dot{q}_\rho - \dot{x}\|^2 + \rho^2 \|\dot{q}_\rho\|^2 \right), \quad (4.6)$$

where ρ is the damping factor. The solution of (4.6) is

$$\dot{\mathbf{q}}_\rho = \mathbf{J}_\rho^* \dot{\mathbf{x}}, \quad (4.7)$$

where

$$\mathbf{J}_\rho^* = \left(\mathbf{J}^T \mathbf{J} + \rho^2 \mathbf{I} \right)^{-1} \mathbf{J}^T, \quad (4.8)$$

which always exists for $\rho \neq 0$. This solution is a compromise between the residual error, $\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}$, and the velocity $\dot{\mathbf{q}}$ of the point P .

Singular value decomposition (SVD) can provide insight into the singularities of the inverse Jacobian [19, 22, 23]. This study deals with nonredundant manipulators, and there are two orthogonal matrices $\mathbf{U} = [u_1 \cdots u_6]$ and $\mathbf{V} = [v_1 \cdots v_6]$, such that

$$\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^6 \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \quad (4.9)$$

where $\mathbf{\Sigma} = \text{diag}[\sigma_1, \dots, \sigma_6]$ and σ_i are the singular values of \mathbf{J} . The vectors \mathbf{u}_i and \mathbf{v}_i are the i th left and the i th right singular vectors, respectively. Substituting (4.9) into (4.8) gives

$$\mathbf{J}_\rho^* = \mathbf{V}\mathbf{\Sigma}_\rho^* \mathbf{U}^T = \sum_{i=1}^6 \frac{\sigma_i}{\sigma_i^2 + \rho^2} \mathbf{v}_i \mathbf{u}_i^T, \quad (4.10)$$

where $\mathbf{\Sigma}_\rho^* = \text{diag}[\sigma_1/(\sigma_1^2 + \rho^2), \dots, \sigma_6/(\sigma_6^2 + \rho^2)]$. When the manipulator is far away from a singular point (i.e., $\sigma_i \gg \rho$), we have $\sigma_i/(\sigma_i^2 + \rho^2) \approx 1/\sigma_i$, which implies that $\mathbf{J}_\rho^* \approx \mathbf{J}^{-1}$. When the manipulator is near a singular point, it can be seen from (4.10) that the solutions for the velocity $\dot{\mathbf{q}}$ have finite values.

4.3. The Hybrid Damped Resolved-Acceleration Control Scheme

In this paper, we propose a hybrid damped resolved-acceleration control scheme (HDRAC) that restricts velocity as well as acceleration. The HDRAC is used to solve the following optimization problem:

$$\min_{\ddot{\mathbf{q}}_{hd}} \left(\|\mathbf{J}\ddot{\mathbf{q}}_{hd} - \mathbf{a}_d^*\|^2 + \rho^2 \|\ddot{\mathbf{q}}_{hd} + \rho_{hd}\dot{\mathbf{q}}\|^2 \right), \quad (4.11)$$

where \mathbf{a}_d^* is defined in (4.2), ρ is the damping factor, and ρ_{hd} is an additional damping factor which is dependent on the smallest singular value of the Jacobian matrix. The joint acceleration commands $\ddot{\mathbf{q}}_{hd}$ have to minimize (4.11). When the manipulator is far away from the singular point, there is no unnecessary joint velocity, so ρ_{hd} will be zero. If the manipulator is at a singular point, the unnecessary joint velocity will have to be removed by joint acceleration commands, so ρ_{hd} must be specified. The redundant components of the joint acceleration commands $\ddot{\mathbf{q}}_{hd}$ will be equal to the unnecessary components of $\dot{\mathbf{q}}$ multiplied by ρ_{hd} to minimize (4.11).

Equation (4.11) can be rewritten as

$$\min_{\dot{\mathbf{q}}_{hd}} \left\| \begin{bmatrix} \mathbf{J} \\ \rho I_6 \end{bmatrix} \dot{\mathbf{q}}_{hd} - \begin{bmatrix} \mathbf{a}_d^* \\ \rho \dot{\mathbf{q}} \end{bmatrix} \right\|^2, \quad (4.12)$$

where I_6 denotes a 6×6 identity matrix. In order to obtain the solutions for this optimization problem in a closed form, the discrete-time technique will be used. With discrete-time control, the back difference is always used to approximate the derivative, so that

$$\ddot{\mathbf{q}}_{hd}(k) = \frac{1}{\Delta t} (\dot{\mathbf{q}}_{hd}(k) - \dot{\mathbf{q}}_{hd}(k-1)), \quad (4.13)$$

where Δt is the sampling time. Through reduction and incorporation, the solution of the optimization problem is [19]

$$\ddot{\mathbf{q}}_{hd} \equiv \ddot{\mathbf{q}}_{da} + \ddot{\mathbf{q}}_{hdc} = \mathbf{J}_\rho^* \mathbf{a}_d^* - \rho_{hd} \rho^2 (\mathbf{J}^T \mathbf{J} + \rho^2 I_6)^{-1} \dot{\mathbf{q}}. \quad (4.14)$$

Meanwhile, we have denoted the second term as $\ddot{\mathbf{q}}_{hdc}$.

The SVD theory can be reformulated as

$$\ddot{\mathbf{q}}_{hdc} = \rho_{hd} \left[\sum_{i=1}^6 \frac{-\rho^2}{\sigma_i^2 + \rho^2} (\mathbf{v}_i^T \dot{\mathbf{q}}) \mathbf{v}_i \right]. \quad (4.15)$$

The size of the deceleration region of the HDRAC can be adjusted by a linear function ρ_{hd} , which is chosen as

$$\rho_{hd} = \begin{cases} \frac{1}{\Delta t} \left(1 - \frac{\sigma_{\min}}{\delta_{hd}} \right), & \text{for } 0 \leq \sigma_{\min} < \delta_{hd}, \\ 0, & \text{for } \sigma_{\min} \geq \delta_{hd}, \end{cases} \quad (4.16)$$

where σ_{\min} is the smallest singular value or singular parameter [19] of the Jacobian matrix. δ_{hd} is a user-defined threshold value that represents the deceleration region. Unnecessary joint velocities will be partially removed when σ_{\min} is in this region and wholly removed at the next sampling interval when $\rho_{hd} = 1/\Delta t$. Note that the maximum value of ρ_{hd} is $1/\Delta t$. If the manipulator is at a singular point of $\sigma_i \gg \rho > 0$, $i = 1, \dots, k$, and $j = k+1, \dots, 6$, then (4.15) can be reduced to

$$\ddot{\mathbf{q}}_{hdc} \approx \frac{-1}{\Delta t} \sum_{i=k+1}^6 (\mathbf{v}_i^T \dot{\mathbf{q}}) \mathbf{v}_i. \quad (4.17)$$

We will next prove that this term removes unnecessary joint velocities at the next sampling interval. Multiplying both sides of (4.13) by $\mathbf{v}_i \mathbf{v}_i^T$ yields

$$\frac{1}{\Delta t} (\mathbf{v}_i^T \dot{\mathbf{q}}_{hd}) \mathbf{v}_i = (\mathbf{v}_i^T \ddot{\mathbf{q}}_{hd}) \mathbf{v}_i + \frac{1}{\Delta t} (\mathbf{v}_i^T \dot{\mathbf{q}}) \mathbf{v}_i. \quad (4.18)$$

By using an ideal computed-torque scheme, we force joint velocity \mathbf{v}_i (i.e., $\mathbf{v}_i^T \dot{\mathbf{q}}$) along the degenerated direction to zero at the next sampling interval. The joint acceleration command along \mathbf{v}_i (i.e., the product of $\mathbf{v}_i^T \ddot{\mathbf{q}}_{hd}$) should then be $(-1/\Delta t)(\mathbf{v}_i^T \dot{\mathbf{q}})$, which follows from (4.18). This implies that (4.17) will remove unnecessary joint velocities along all degenerated directions \mathbf{v}_i , $i = k + 1, \dots, 6$. According to (4.18), the desired joint velocities along \mathbf{v}_i , $i = 1, \dots, k$ are also decreased by the amount of $\Delta t \mathbf{v}_i^T \ddot{\mathbf{q}}_{hd}$ (or $[\rho^2/(\sigma_i^2 + \rho^2)]\mathbf{v}_i^T \dot{\mathbf{q}}$). Note that the maximum coefficient of $(\mathbf{v}_i^T \dot{\mathbf{q}})\mathbf{v}_i$ in (4.15) is $-1/\Delta t$ when $\sigma_i = 0$.

5. Stability Analysis

In this section, we will show that the HDRAC is asymptotically stable and discuss its convergent property. Let the orientation error be $\varepsilon_e = f(\theta_e)u_e$, where u_e and θ_e are the rotational axis and the angle between the current orientation and the desired orientation, respectively, and $f(\theta_e) = \theta_e$ for very small values of θ_e .

Suppose that $\rho > 0$ when the manipulator is at a singular point. A sufficient condition for the global asymptotic convergence of the HDRAC in the whole workspace of the manipulator in a situation, where $\ddot{r}_d = \dot{r}_d = 0$ and $\alpha_d = \omega_d = 0$, is that \mathbf{K}_D is a positive-definite matrix and

$$\mathbf{K}_P = \begin{bmatrix} k_{pr}I_3 & 0 \\ 0 & k_{pe}I_3 \end{bmatrix}, \quad (5.1)$$

where k_{pr} and k_{pe} are positive. The equilibrium is at the point where $\dot{\mathbf{q}} = 0$. Let a Lyapunov function be [24]

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}k_{pr}\varepsilon_r^T \varepsilon_r + k_{pe}E + \frac{1}{2}\dot{\mathbf{q}}^T (\mathbf{J}^T \mathbf{J} + \rho^2 \mathbf{I}_6) \dot{\mathbf{q}}, \quad (5.2)$$

where $\varepsilon_r = r_d - r$ and $E = \int_0^{\theta_e} f(\phi) d\phi = (1/2)\theta_e^2$, for $f(\phi) = \phi$. Clearly, $L(\mathbf{q}, \dot{\mathbf{q}}) \geq 0$. It is recognized that $\dot{\theta}_e = u_e(\omega_d - \omega)$. Differentiating E with respect to time, we obtain

$$\dot{E} = \frac{\partial E}{\partial \theta_e} \dot{\theta}_e = \varepsilon_e(\omega_d - \omega). \quad (5.3)$$

Evaluating $\partial L/\partial t$ in terms of the solutions of (5.2) yields

$$\frac{\partial L}{\partial t} = k_{pr}\varepsilon_r^T (\dot{r}_d - \dot{r}) + k_{pe}f(\theta_e)u_e^T (\omega_d - \omega) + \dot{\mathbf{q}}^T (\mathbf{J}^T \mathbf{J} + \rho^2 \mathbf{I}_6) \ddot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{J}^T \mathbf{J} \dot{\mathbf{q}}. \quad (5.4)$$

If the ideal computed-torque control is used, $\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{hd}$, (4.14) can be substituted into (5.4), which gives us [24]

$$\frac{\partial L}{\partial t} = k_{pr}\varepsilon_r^T \dot{r}_d + k_{pe}\varepsilon_e^T \omega_d + \dot{\mathbf{q}}^T \mathbf{J}^T \begin{bmatrix} \dot{r}_d \\ \alpha_d \end{bmatrix} + \dot{\mathbf{q}}^T \mathbf{J}^T \mathbf{K}_D \left(\begin{bmatrix} \dot{r}_d \\ \omega_d \end{bmatrix} - \mathbf{J} \dot{\mathbf{q}} \right) - \rho_{hd} \rho^2 \dot{\mathbf{q}}^T \dot{\mathbf{q}}. \quad (5.5)$$

If $\ddot{r}_d = \dot{r}_d = 0$ and $\alpha_d = \omega_d = 0$, then

$$\frac{\partial L}{\partial t} = -\dot{\mathbf{q}}^T \mathbf{J}^T \mathbf{K}_D \mathbf{J} \dot{\mathbf{q}} - \rho_{hd} \rho^2 \dot{\mathbf{q}}^T \dot{\mathbf{q}}. \quad (5.6)$$

Since \mathbf{K}_D is a positive-definite matrix, $\partial L/\partial t \leq 0$ for all $\dot{\mathbf{q}}$, and $\partial L/\partial t = 0$ only for $\dot{\mathbf{q}} = 0$. Combining the above results and $L(\mathbf{q}, \dot{\mathbf{q}}) \geq 0$, it is shown by the Lyapunov theorem that $\dot{\mathbf{q}}$ is asymptotically stable, and the equilibrium point is $\dot{\mathbf{q}} = 0$. However, for the proposed control scheme, $\partial L/\partial t$ has an additional term $(-\rho_{hd} \rho^2 \dot{\mathbf{q}}^T \dot{\mathbf{q}})$ more than that of the HDRAC, to ensure that $\partial L/\partial t$ is zero only for $\dot{\mathbf{q}} = 0$, that is, the equilibrium point is $\dot{\mathbf{q}} = 0$.

6. Numerical Simulations and Experimental Results

The purpose of inverse dynamics analysis is to determine the required torque of the separately-excited DC servomotor for a given motion trajectory of the moving platform. The method [19] is adopted to analyze the 3RPS parallel manipulator dynamics of the links and moving platform by using Newton-Euler equations of motion, which are formulated by considering the free-body diagrams of the links and moving platform separately.

Let us focus on the mass center O_i of the i th link as shown in Figure 5, where ${}^i F_{pix}$, ${}^i F_{piy}$, and ${}^i F_{piz}$ are three components of the weight force of the moving platform acting on the i th link, W_i is the weight force of the i th link, ${}^i F_{1motor}$ is the acting force of the separately-excited DC servomotor on the revolution parts, and the ${}^i F_{2motor}$ is the acting force on each vertical leg. Each link is placed along the X_i axis; therefore, its inertia tensor about X_i , Y_i and Z_i is a diagonal matrix with diagonal elements I_{ixx} , I_{iyy} , and I_{izz} . The angular momentum [19] of the i th link is

$$\mathbf{H}_i = (I_{ixx} \omega_{ix}) \mathbf{i}_i + (I_{iyy} \omega_{iy}) \mathbf{j}_i + (I_{izz} \omega_{iz}) \mathbf{k}_i. \quad (6.1)$$

The rate change of \mathbf{H}_i is given by

$$\dot{\mathbf{H}}_i = (\dot{\mathbf{H}}_i)_{xyzi} + \boldsymbol{\omega}_i \times \mathbf{H}_i. \quad (6.2)$$

It follows from Figures 2 and 5 that the external moment about the mass center O_i of the i th link is given by

$$\sum \mathbf{M}_{O_i} = {}^i \mathbf{r}_{pi} \times {}^i \mathbf{F}_{pi}, \quad (6.3)$$

where ${}^i \mathbf{r}_{pi}$ is the vector from O_i to B_i with respect to the link frame with unit vector $[i \ j \ k]$.

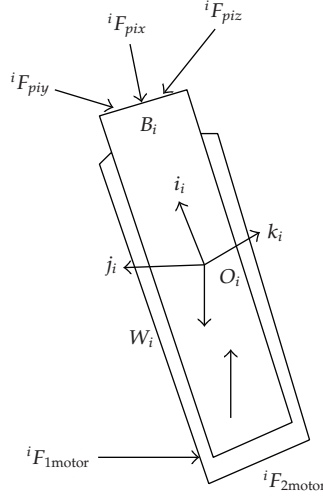


Figure 5: Free-body diagram of acting forces on the i th link.

The equations of rotational motion of the i th link can be obtained by equating the rate of change of \mathbf{H}_i with the external moment. By equating (6.2) with (6.3), we obtain

$$\begin{aligned} (I_{izz}\omega_{iz}\omega_{iy} - I_{iy y}\omega_{iy}\omega_{iz}) + \dot{I}_{ixx}\omega_{ix} + I_{ixx}\dot{\omega}_{ix} &= ({}^i r_{piy} {}^i F_{piz} - {}^i r_{piz} {}^i F_{piy}), \\ -(I_{zz}\omega_{iz}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iz}) + \dot{I}_{iy y}\omega_{iy} + I_{iy y}\dot{\omega}_{iy} &= -({}^i r_{pix} {}^i F_{piz} - {}^i r_{piz} {}^i F_{pix}), \\ (I_{iy y}\omega_{iy}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iy}) + \dot{I}_{zz}\omega_{iz} + I_{iz}\dot{\omega}_{iz} &= ({}^i r_{pix} {}^i F_{piy} - {}^i r_{piy} {}^i F_{pix}). \end{aligned} \quad (6.4)$$

Since $\dot{I}_{ixx} = \dot{I}_{iy y} = \dot{I}_{zz} = 0$, ${}^i F_{pix}$ passing through the mass center O_i do not generate torques, and ${}^i r_{pi}$ only has values in the X_i -axis and couples with the separately-excited DC motor; the above equations (6.4) can be simplified as follows:

$$\begin{aligned} {}^i r_{pix} {}^i F_{piz} &= I_{zz}\omega_{iz}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iz} - I_{iy y}\dot{\omega}_{iy} + J_{m1}\dot{\omega}_{iy} + B_{m1}\omega_{iy} - K_{T1}i_{a1} - K_{T2}i_{a2}, \\ {}^i r_{pix} {}^i F_{piy} &= I_{iy y}\omega_{iy}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iy} + I_{zz}\dot{\omega}_{iz} + J_{m1}\dot{\omega}_{iz} + B_{m1}\omega_{iz} - K_{T1}i_{a1} - K_{T2}i_{a2}, \end{aligned} \quad (6.5)$$

where J_{m1} is the moment of inertia; B_{m1} is the damped coefficient; K_{T1} and K_{T2} are the torque constants; i_{a1} and i_{a2} are the armature currents.

We now process to find the total force acting on the mass center O_i as follows:

$$\begin{aligned} {}^i F_{piy} &= \frac{\{I_{iy y}\omega_{iy}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iy} + I_{zz}\dot{\omega}_{iz} + J_{m1}\dot{\omega}_{iz} + B_{m1}\omega_{iz} - K_{T1}i_{a1} - K_{T2}i_{a2}\}}{{}^i r_{pix}}, \\ {}^i F_{piz} &= \frac{\{I_{zz}\omega_{iz}\omega_{ix} - I_{ixx}\omega_{ix}\omega_{iz} - I_{iy y}\dot{\omega}_{iy} + J_{m1}\dot{\omega}_{iy} + B_{m1}\omega_{iy} - K_{T1}i_{a1} - K_{T2}i_{a2}\}}{{}^i r_{pix}}. \end{aligned} \quad (6.6)$$

Once the force terms ${}^iF_{piy}$ and ${}^iF_{piz}$ have been obtained in (6.6) of link motion, one should take into consideration the force term ${}^iF_{pix}$. The angular momentum [19] of the moving platform is described by

$$\mathbf{H}_p = (I_{p_{xx}}\omega_{px})\mathbf{i}_p + (I_{p_{yy}}\omega_{py})\mathbf{j}_p + (I_{p_{zz}}\omega_{pz})\mathbf{k}_p, \quad (6.7)$$

where $I_{p_{xx}}$, $I_{p_{yy}}$, and $I_{p_{zz}}$ are, respectively, the mass moments of inertia of the moving platform about the X , Y , and Z axes. The rate change of angular momentum is

$$\dot{\mathbf{H}}_p = (\dot{\mathbf{H}}_p)_{xyzp} + \boldsymbol{\omega}_p \times \mathbf{H}_p. \quad (6.8)$$

The total torque acting on mass center P of the moving platform is given by

$$\sum \mathbf{M}_P = \sum_{i=1}^3 {}^p\mathbf{P}_i \times {}^p\mathbf{F}_{pi}, \quad (6.9)$$

where ${}^p\mathbf{P}_i$ is the vector from P to B_i , $i = 1, 2, 3$.

6.1. Numerical Simulations

In this section, numerical simulations are offered to demonstrate the applicability and effectiveness of the proposed algorithms of the 3RPS manipulator platform. The physical model is shown in Figure 6. The moving platform of the manipulator is a circle with radius $b = 22.08$ cm, and the distance between any two of the three spherical joints on the moving platform is $w = 36.16$ cm. The fixed base is an inscribed triangle within a circle with radius $a = 41.72$ cm. In the physical model, the constraints of the lengths d_1 , d_2 , and d_3 are ± 10 cm, and the revolution angles θ_1 , θ_2 , and θ_3 are $\pm 20^\circ$. A program has been written in order to calculate the singular point. It was found that when $d_1 = 0.0717$ cm, $d_2 = 9.7488$ cm, and $d_3 = 0.0771$ cm, and $\theta_1 = 19.9834^\circ$, $\theta_2 = 0.2438^\circ$, and $\theta_3 = 19.9074^\circ$, then one singular point exists in the workspace. That is, the center coordinate of the moving platform with respect to the fixed base is $(1.6144, -2.7933, 0.0308)$, and the unit is cm. In this study, point P is required to move at a constant speed from the starting point $(0, 0, 0)$, to then pass through the singular point $(1.6144, -2.7933, 0.0308)$, and then it arrives at its final point $(5, -5, 5)$.

In this study, the HDRAC scheme is used to solve the damping factors ρ by means of the MPSO optimization method. Damping factors ρ are not fixed values and are used to damp the accelerations of point P when the manipulator is in the neighborhood of singularities. The dependence of the damping factors on the accelerations can change. For on-line applications, the optimization method can immediately provide optimal damping factors.

Figures 7(a)–7(f) show numerical simulations of the trajectories and velocities along the X -axis, Y -axis, and Z -axis for the controller incorporating HDRAC with fixed damping values, and for the controller incorporating HDRAC with MPSO. For the controller incorporating HDRAC with MPSO, when the manipulator is near the singular point, the system functions are more smooth and there is no fluctuation.



Figure 6: Physical model of the 3RPS parallel manipulator.

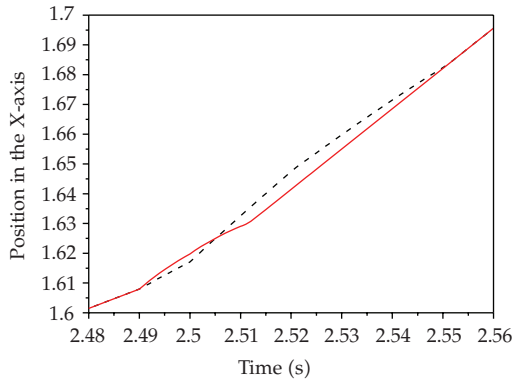
Table 1: The characteristic of position limiter sensor.

Static platform error due to arm position sensor errors (mm), this error represents the vector conversion error from joint coordinates to the platform reference coordinates	
Translation (mm)	1 (max)
Rotation (deg)	1.5 (max)
Linear position static accuracy (mm)	± 3
Repeatability	
Translation (mm)	0.15 (max)
Rotation (deg)	0.2 (max)
Resolution	
Translation (mm)	0.1
Rotation (deg)	0.15

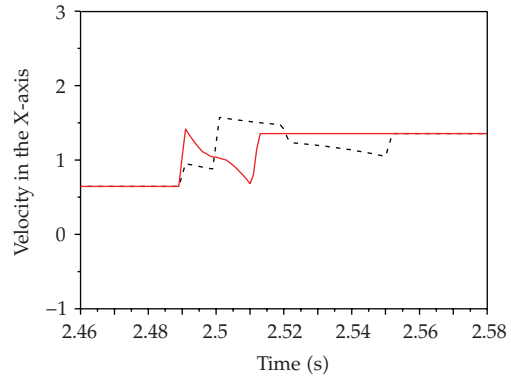
6.2. Experimental Setup

In order to demonstrate the proposed control rules, PC-based experimental equipment was devised. The parallel manipulator was coupled with a DC motor. The translation position was measured by a sensor. A photo of the experimental equipment is shown in Figure 6. In general, such parallel robotic is controlled by hydraulic servo, that is, to control the extension or retraction of the limb through driving with hydraulic system. Limiter sensor or measuring tools are generally installed on the link of the limb to enable accurate changes in its length when controlling positions; the characteristic of position limiter sensor is shown in the Table 1. The controller is based on a PC with Pentium-586 CPU. In order to measure the angle, angular speed, position, and velocity of the manipulator, the interface device was implemented by the motion control card PCI-7344. The graphical software of Simulink and the control software LABVIEW were used to implement the proposed control rule. At the same time, this software also determines the linear converts between the physical scales and measured voltages from sensors.

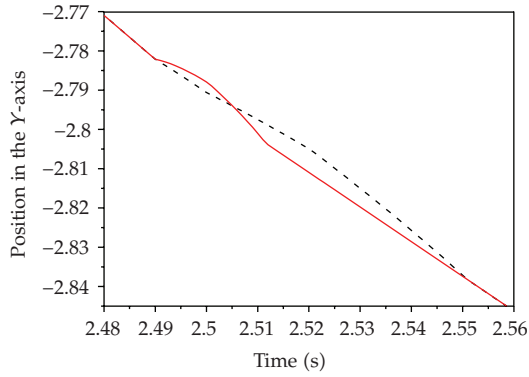
A block diagram of the overall system with the controller incorporating HDRAC with MPPO and acceleration feedback is illustrated in Figure 8. The experimental instrument of



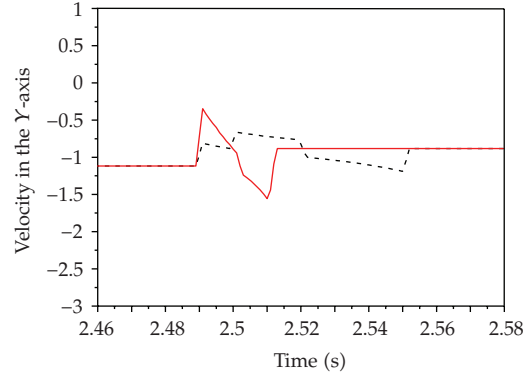
(a)



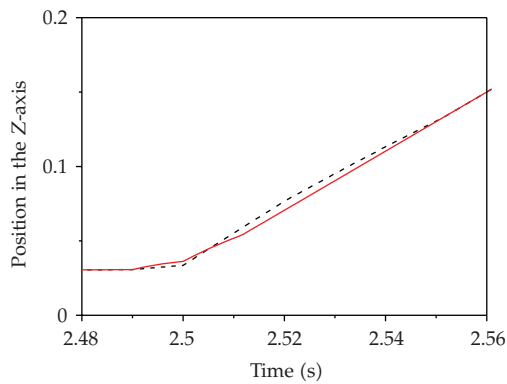
(b)



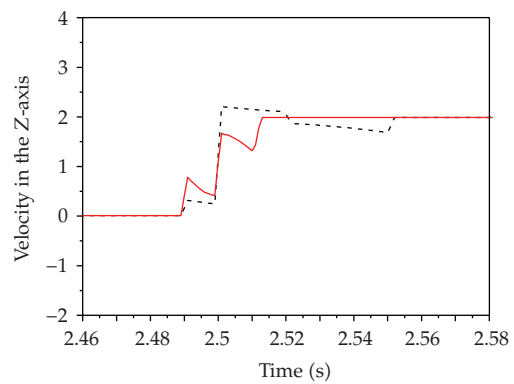
(c)



(d)



(e)



(f)

----- HDRAC with a fixed damping value
 ——— HDRAC with MPSO

----- HDRAC with a fixed damping value
 ——— HDRAC with MPSO

Figure 7: Numerical simulations for the controller incorporating HDRAC with fixed damping values and the controller incorporating HDRAC with MPSO. (a) Translation responses along the X-axis. (b) Velocities along the X-axis. (c) Translation responses along the Y-axis. (d) Velocities along the Y-axis. (e) Translation responses along the Z-axis. (f) Velocities along the Z-axis.

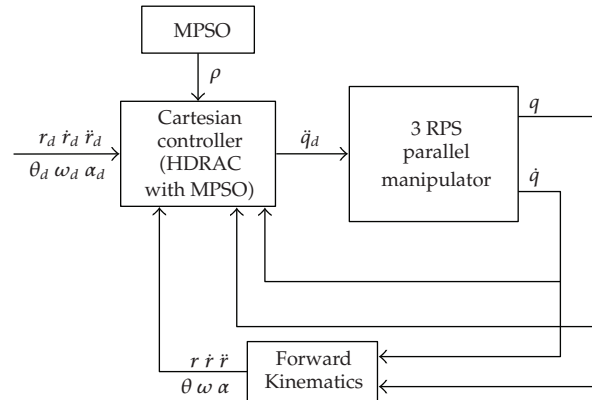


Figure 8: Block diagram of the controller incorporating HDRAC with MPSO and acceleration feedback.

the 3RPS parallel manipulator was divided into three parts: an actuator, parallel manipulator, and controller. The actuator part consisted of a DC motor and a driver. The parameters of the DC motor were $K_t = 0.21 \text{ Nm/A}$, $J_m = 0.000037 \text{ Nms}^2$, $B_m = 0.000013 \text{ Nms/rad}$, $l_a = 2.4 \text{ mH}$, and $r_a = 4.8 \Omega$.

6.3. Experimental Results

Figures 9(a)–9(f) show the experimental result of trajectory and velocity along the X-axis, Y-axis, and Z-axis with the DARAC controller that was obtained by using the 3RPS parallel manipulator, in which the end effector passed through a singular point. The DARAC controller functioned, such that joint acceleration along the degenerated direction was damped when the end effector was close to the singular point. It was found that when the manipulator approached the singular point, the system was unstable for the DARAC controller and displayed great fluctuations.

Figures 10(a)–10(f) show the experimental result of trajectory and velocity along the X-axis, Y-axis, and Z-axis for the controller incorporating HDRAC with fixed damping values and for the controller incorporating HDRAC with MPSO. It was found that, by using the controller incorporating HDRAC with MPSO, a very good performance could be achieved under the same conditions and that the system was generally stable and had very slight fluctuations. These experimental results demonstrate that the HDRAC overcomes the problem of fluctuation when the manipulator arrives at the singular point. It also proves that the proposed controller incorporating HDRAC with MPSO has great robustness.

7. Conclusion

This paper proposes a simple HDRAC scheme with an MPSO method, which has the merits of offering a high-quality solution and better computation efficiency in the neighborhood of a singular point. The proposed method directly damps velocities and accelerations, such that unnecessary joint velocities are removed at a singular point. The proposed HDRAC scheme with MPSO control is asymptotically stable and remedies the problem of fluctuation.

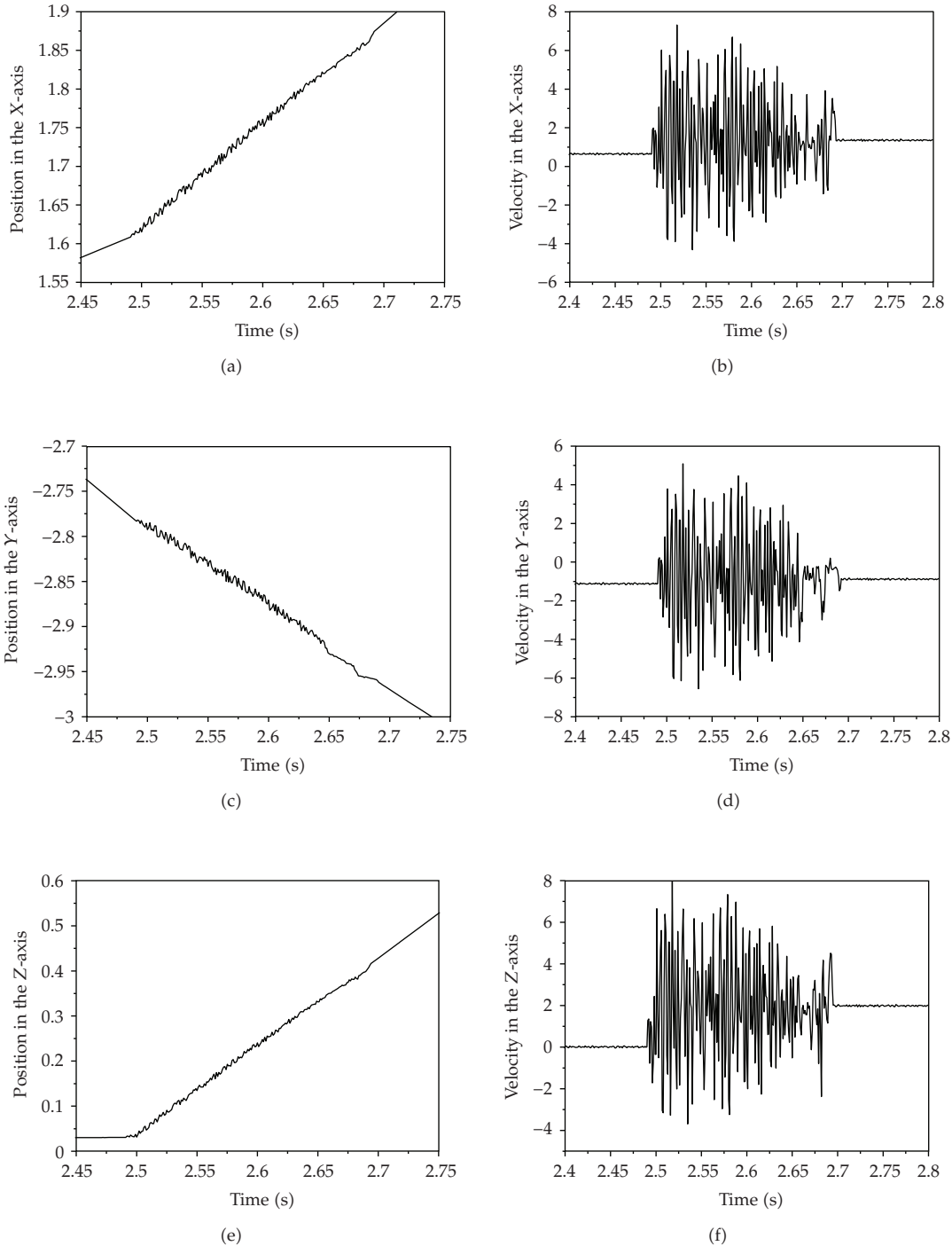


Figure 9: Experimental results for the controller incorporating DARAC with fixed damping values. (a) Translation responses along the X-axis. (b) Velocities along the X-axis. (c) Translation responses along the Y-axis. (d) Velocities along the Y-axis. (e) Translation responses along the Z-axis. (f) Velocities along the Z-axis.

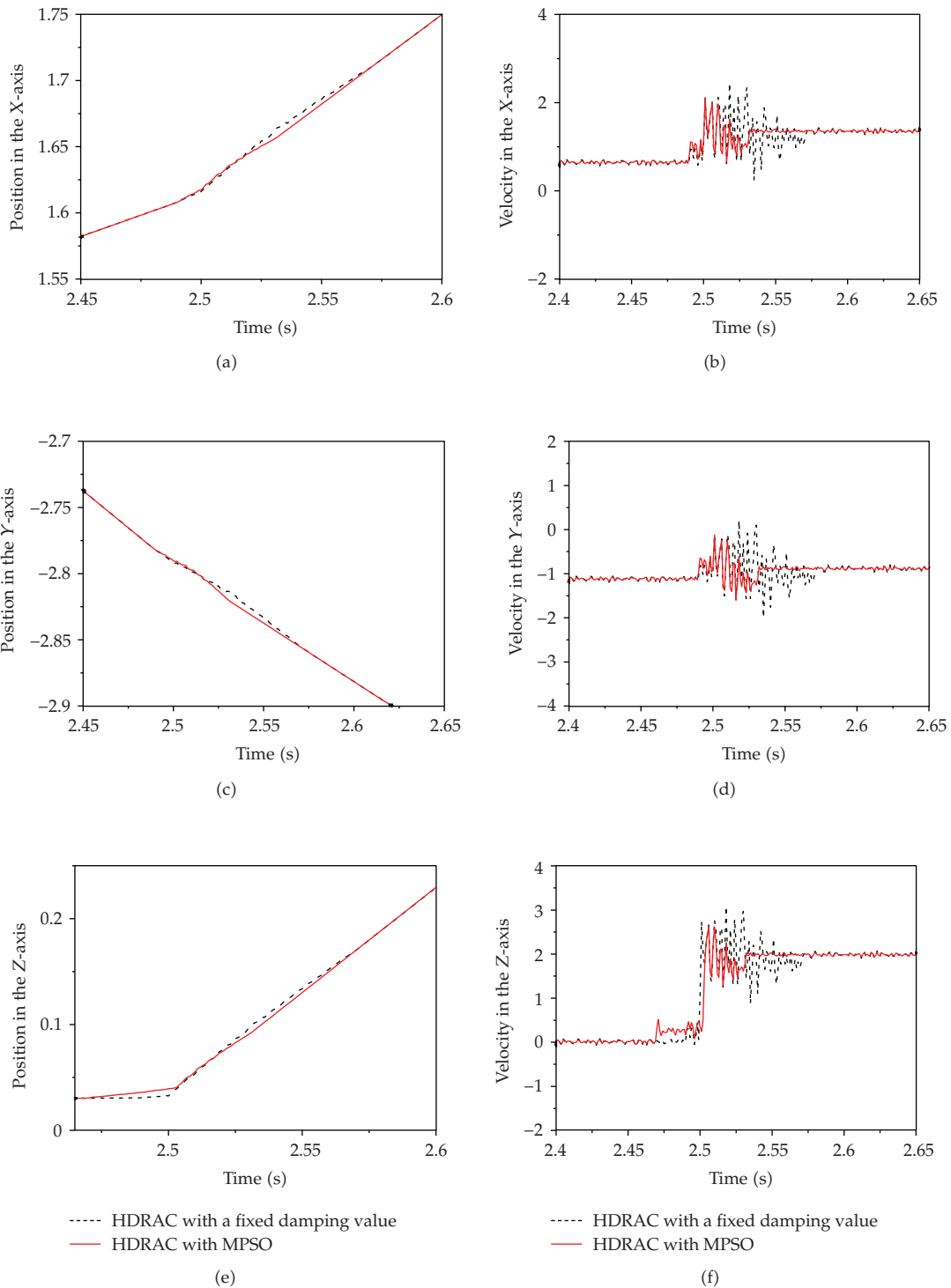


Figure 10: Experimental results for the controller incorporating HDRAC with fixed damping values and the controller incorporating HDRAC with MPSO. (a) Translation responses along the X-axis. (b) Velocities along the X-axis. (c) Translation responses along the Y-axis. (d) Velocities along the Y-axis. (e) Translation responses along the Z-axis. (f) Velocities along the Z-axis.

In this study, HDRAC with MPSO is used to solve the damping factors ρ by the optimization method. The damping factors ρ are not fixed values for on-line applications, and the optimization method can immediately provide optimal damping factors. Because it shortens the detection processes, the MPSO method can satisfactorily predict optimal damping values.

Comparisons between our numerical simulations and experimental results showed that the proposed method offers greater stability and efficiency and solves the problems of instability and fluctuation more easily and quickly. The main advantage of using a controller incorporating HDRAC with MPSO is that the 3RPS parallel manipulator control system does not need to plan its path to avoid any singular point, because the controller will command the manipulator to move near the singular point with minimum trajectory errors. Moreover, in comparison with a controller incorporating HDRAC with fixed damping values, the proposed approach makes it easier and simpler to improve the performance of the motion in the neighborhood of singular points.

Appendices

A.

The items contained in matrix **D** are as follows:

$$\begin{aligned} D_{1i} &= {}^pP_{ix} R_{pi21} - {}^pP_{iy} R_{pi11}, & D_{2i} &= -{}^pP_{ix} R_{pi31}, & D_{3i} &= {}^pP_{iy} R_{pi31}, & D_{4i} &= R_{i11}, \\ D_{5i} &= R_{i21}, & D_{6i} &= R_{i31}. \end{aligned} \quad (\text{A.1})$$

The items contained in matrix **Q** are as follows:

$$\begin{aligned} Q_1 &= I_{pzz}\dot{\omega}_{pz} + (I_{pyy} - I_{pxx})\omega_{py}\omega_{px} \\ &\quad - \sum_{i=1}^3 \left\{ {}^pP_{ix} \left(R_{pi22} {}^iF_{piy} + R_{pi23} {}^iF_{piz} \right) - {}^pP_{iy} \left(R_{pi12} {}^iF_{piy} + R_{pi13} {}^iF_{piy} + R_{pi13} {}^iF_{piz} \right) \right\} \\ &\quad + J_{m1}\dot{\omega}_{pz} + B_{m1}\omega_{pz} - K_{T1}i_{a1} - K_{T2}i_{a2}, \\ Q_2 &= I_{pyy}\dot{\omega}_{py} + (I_{pxx} - I_{pzz})\omega_{px}\omega_{pz} + \sum_{i=1}^3 {}^pP_{ix} \left(R_{pi32} {}^iF_{piy} + R_{pi33} {}^iF_{piz} \right) \\ &\quad \times J_{m1}\dot{\omega}_{py} + B_{m1}\omega_{py} - K_{T1}i_{a1} - K_{T2}i_{a2}, \\ Q_3 &= I_{pxx}\dot{\omega}_{px} + (I_{pzz} - I_{pyy})\omega_{py}\omega_{pz} - \sum_{i=1}^3 {}^pP_{iy} \left(R_{pi32} {}^iF_{piy} + R_{pi33} {}^iF_{piz} \right) \\ &\quad \times J_{m1}\dot{\omega}_{px} + B_{m1}\omega_{px} - K_{T1}i_{a1} - K_{T2}i_{a2}, \\ Q_4 &= m_p a_{px} - \sum_{i=1}^3 \left(R_{i12} {}^iF_{piy} + R_{i13} {}^iF_{piz} \right), \end{aligned}$$

$$\begin{aligned}
Q_5 &= m_p a_{py} - \sum_{i=1}^3 (R_{i22} {}^i F_{piy} + R_{i23} {}^i F_{piz}), \\
Q_6 &= m_p a_{pz} - \sum_{i=1}^3 (R_{i32} {}^i F_{piy} + R_{i33} {}^i F_{piz}) + W_p.
\end{aligned} \tag{A.2}$$

B.

The items contained in matrix **A** are as follows:

$$\begin{aligned}
a_{11} &= 2[\cos(\alpha_2 - \alpha_1)(d_1 \sin \theta_1 + d_1 d_2 \sin \theta_1 \cos \theta_2) - (d_1 \sin \theta_1 + d_1 d_2 \sin \theta_2 \cos \theta_1)], \\
a_{12} &= 2[\cos(\alpha_2 - \alpha_1)(d_2 \sin \theta_2 + d_1 d_2 \sin \theta_2 \cos \theta_1) - (d_2 \sin \theta_2 + d_1 d_2 \sin \theta_1 \cos \theta_2)], \\
a_{22} &= 2[\cos(\alpha_3 - \alpha_2)(d_2 \sin \theta_2 + d_2 d_3 \sin \theta_2 \cos \theta_3) - (d_2 \sin \theta_2 + d_2 d_3 \sin \theta_3 \cos \theta_2)], \\
a_{23} &= 2[\cos(\alpha_3 - \alpha_2)(d_3 \sin \theta_3 + d_2 d_3 \sin \theta_3 \cos \theta_2) - (d_3 \sin \theta_3 + d_2 d_3 \sin \theta_2 \cos \theta_3)], \\
a_{31} &= 2[\cos(\alpha_1 - \alpha_3)(d_1 \sin \theta_1 + d_1 d_3 \sin \theta_1 \cos \theta_3) - (d_1 \sin \theta_1 + d_1 d_3 \sin \theta_3 \cos \theta_1)], \\
a_{33} &= 2[\cos(\alpha_1 - \alpha_3)(d_3 \sin \theta_3 + d_1 d_3 \sin \theta_3 \cos \theta_1) - (d_3 \sin \theta_3 + d_1 d_3 \sin \theta_1 \cos \theta_3)].
\end{aligned} \tag{B.1}$$

The items contained in matrix **B** are as follows:

$$\begin{aligned}
b_{11} &= [\cos \theta_1 (d_2 \cos \theta_2 + 1) + 2(d_1 + \cos \theta_1 - d_2 \sin \theta_1 \sin \theta_2)], \\
b_{12} &= [\cos \theta_2 (d_1 \cos \theta_1 + 1) + 2(d_2 + \cos \theta_2 - d_1 \sin \theta_1 \sin \theta_2)], \\
b_{22} &= [\cos \theta_2 (d_3 \cos \theta_3 + 1) + 2(d_2 + \cos \theta_2 - d_3 \sin \theta_2 \sin \theta_3)], \\
b_{23} &= [\cos \theta_3 (d_2 \cos \theta_2 + 1) + 2(d_3 + \cos \theta_3 - d_2 \sin \theta_2 \sin \theta_3)], \\
b_{31} &= [\cos \theta_1 (d_3 \cos \theta_3 + 1) + 2(d_1 + \cos \theta_1 - d_3 \sin \theta_1 \sin \theta_3)], \\
b_{33} &= [\cos \theta_3 (d_1 \cos \theta_1 + 1) + 2(d_3 + \cos \theta_3 - d_1 \sin \theta_1 \sin \theta_3)].
\end{aligned} \tag{B.2}$$

C.

The coefficients are as follows:

$$\begin{aligned}
\Omega_1 &= -\frac{3}{4} [3(e_1 e_2 e_3) + e_1 (e_2^2 + e_3^2) + e_2 (e_1^2 + e_3^2) + e_3 (e_1^2 + e_2^2)], \\
\Omega_2 &= \frac{3}{4} [e_1 h_1 (e_3^2 + e_3 - e_2^2 - e_2) + e_2 h_2 (e_1^2 + e_1 - e_3^2 - e_3) + e_3 h_3 (e_2^2 + e_2 - e_1^2 - e_1)],
\end{aligned}$$

$$\begin{aligned}
\Omega_3 = \frac{1}{4} \{ & - (h_1 h_2 + h_2 h_3 + h_3 h_1) [e_1 (e_2 e_3 - e_2 - e_3 + 4) \\
& + e_2 (e_1 e_3 - e_1 - e_3 + 4) + e_3 (e_1 e_2 - e_1 - e_2 + 4)] \\
& + h_1 (h_3 - h_2) [e_3 (e_2^2 + e_2 - 2) - e_2 (e_3^2 + e_3 - 2)] \\
& + h_2 (h_1 - h_3) [e_1 (e_3^2 + e_3 - 2) - e_3 (e_1^2 + e_1 - 2)] \\
& + h_3 (h_2 - h_1) [e_2 (e_1^2 + e_1 - 2) - e_1 (e_2^2 + e_2 - 2)] \}, \\
\Omega_4 = -\frac{16}{9} \{ & [h_1 h_2 (h_2 - h_1) + h_2 h_3 (h_3 - h_2) + h_3 h_1 (h_1 - h_3)] \\
& \times [e_1 (e_2 e_3 - e_2 - e_3 + 4) + e_2 (e_1 e_3 - e_1 - e_3 + 4) + e_3 (e_1 e_2 - e_1 - e_2 + 4)] \\
& + 3e_2 e_3 (h_2 - h_3)^2 (h_3 - h_1) (e_3 - 1) + 3e_2 e_3 (h_2 - h_3)^2 (h_1 - h_2) (e_2 - 1) \\
& + 3e_1 e_3 (h_3 - h_1)^2 (h_2 - h_3) (e_3 - 1) + 3e_1 e_3 (h_3 - h_1)^2 (h_1 - h_2) (e_1 - 1) \\
& + 3e_1 e_2 (h_1 - h_2)^2 (h_2 - h_3) (e_2 - 1) + 3e_1 e_2 (h_1 - h_2)^2 (h_3 - h_1) (e_1 - 1) \}.
\end{aligned} \tag{C.1}$$

References

- [1] K. M. Lee and D. Shah, "Kinematic analysis of three degree of freedom in-parallel actuated manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 354–360, Raleigh, NC, USA, 1987.
- [2] K. M. Lee and D. Shah, "Dynamic analysis of three degree of freedom in-parallel actuated manipulator," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 361–367, Raleigh, NC, USA, 1988.
- [3] Q. Xu and Y. Li, "An investigation on mobility and stiffness of a 3-DOF translational parallel manipulator via screw theory," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 402–414, 2008.
- [4] Y. Fang and Z. Huang, "Kinematic analysis of 3RPS parallel robotic mechanisms," *Mechanical Science and Technology*, vol. 16, no. 1, pp. 82–87, 1997.
- [5] Y. Li and Q. Xu, "Kinematics and inverse dynamics analysis for a general 3-PRS spatial parallel mechanism," *Robotica*, vol. 23, no. 2, pp. 219–229, 2005.
- [6] Y. Li and Q. Xu, "Kinematic analysis of a 3-PRS parallel manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 395–408, 2007.
- [7] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.
- [8] S.-L. Wu and S.-K. Lin, "Implementation of damped-rate resolved-acceleration robot control," *Control Engineering Practice*, vol. 5, no. 6, pp. 791–800, 1997.
- [9] R. Campa, R. Kelly, and E. García, "On stability of the resolved acceleration control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 3523–3528, May 2001.
- [10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE, December 1995.
- [11] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.
- [12] J. Kennedy, "Particle swarm: social adaptation of knowledge," in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 303–308, April 1997.
- [13] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 60–68, 2003.

- [14] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1945–1950, IEEE, 1999.
- [15] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 5, pp. 1271–1289, 2007.
- [16] C. H. Liu and S. Cheng, "Direct singular positions of 3RPS parallel manipulators," *Journal of Mechanical Design*, vol. 126, no. 6, pp. 1006–1016, 2004.
- [17] T.-S. Hwang and M.-Y. Liao, "Optimal mechanism design and dynamic analysis of a 3-leg 6-DOF linear motor based parallel manipulator," *Asian Journal of Control*, vol. 6, no. 1, pp. 136–144, 2004.
- [18] S. A. Joshi and L.-W. Tsai, "Jacobian analysis of limited-DOF parallel manipulators," *Journal of Mechanical Design*, vol. 124, no. 2, pp. 254–258, 2002.
- [19] C. C. Kao, S. L. Wu, and R. F. Fung, "The 3RPS parallel manipulator motion control in the neighborhood of singularities," in *Proceedings of the International Symposium on Industrial Electronics, Mechatronics and Applications*, vol. 1, pp. 165–179, 2007.
- [20] W.-M. Lin, F.-S. Cheng, and M.-T. Tsay, "Nonconvex economic dispatch by integrated artificial intelligence," *IEEE Transactions on Power Systems*, vol. 16, no. 2, pp. 307–311, 2001.
- [21] W.-M. Lin, F.-S. Cheng, and M.-T. Tsay, "An improved tabu search for economic dispatch with multiple minima," *IEEE Transactions on Power Systems*, vol. 17, no. 1, pp. 108–112, 2002.
- [22] A. A. Maciejewski and C. A. Klein, "Singular value decomposition. Computation and applications to robotics," *International Journal of Robotics Research*, vol. 8, no. 6, pp. 63–79, 1989.
- [23] J. T. Wen and J. F. O'Brien, "Singularities in three-legged platform-type parallel mechanisms," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 720–726, 2003.
- [24] K. W. Spring, "Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review," *Mechanism and Machine Theory*, vol. 21, no. 5, pp. 365–373, 1986.