machines

MDPI

Article

A Model-Based Digital Toolbox for Unified Kinematics and Dimensional Synthesis in Parallel Robot Design

Zhen He [1], Chengjin Hu [1], Tengfei Tang [2], Hanliang Fang [3], Yibo Jiang [1], Fufu Yang [1] and Jun Zhang [1,*]

[1] School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350116, China; 230210010@fzu.edu.cn (Z.H.); 210210007@fzu.edu.cn (C.H.); jiangyb@fzu.edu.cn (Y.J.); yangfufu@fzu.edu.cn (F.Y.)
[2] School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; tengfei@zstu.edu.cn
[3] School of Advanced Manufacturing, Fuzhou University, Jinjiang 362251, China; hanliang_fang@fzu.edu.cn
* Correspondence: zhang_jun@fzu.edu.cn; Tel.: +86-13489993804

Abstract

A unified digital toolbox is introduced for kinematics analysis and dimension synthesis of parallel robots, addressing challenges in configuration diversity and computational complexity. By integrating hierarchical kinematic modeling with screw theory, the toolbox establishes standardized analytical frameworks for mobility, inverse kinematics and dexterity evaluation. A modular toolbox architecture—comprising interactive, data, external module, database and functional layers—enables systematic design, workspace estimation and dexterity-driven optimization. A hybrid MATLAB-C++ interface ensures computational efficiency and scalability. The efficacy of the toolbox is demonstrated through a case study on a novel 2UPR-2RPS parallel mechanism, achieving optimized dimensional parameters (k_1 = 0.85, k_2 = 1.3, k_3 = 0.85, k_4 = 1.3) with a mean dexterity index of 0.637 and validated workspace symmetry. Results confirm that the toolbox streamlines the design process, ensures computational accuracy and enables rapid adaptation to new robotic configurations. This work provides a robust foundation for advanced parallel robot design, offering significant potential for industrial and research applications requiring high-precision motion control.

Keywords: parallel robots; kinematic; digital toolbox; dimension synthesis

1. Introduction

Parallel robots, renowned for their robust reconfigurability, compact structure, high stiffness and superior dynamic performance, have become indispensable in the aerospace, automotive and other industries for precision component manufacturing purposes [1]. They are celebrated as one of the most innovative engineering designs of the past 20 years, with significant impacts on various sectors, including high-precision manufacturing. For instance, the Tricept robot [2], initially employed in automotive manufacturing, has demonstrated its superior stiffness and repeatability in high-precision aerospace manufacturing, surpassing traditional serial robots. This innovation, along with other advancements in parallel robot design [3,4], has boosted the development of numerous high-performance parallel machining systems like Sprint Z3 [5], Exechon [6,7], Trimule [8] and DiaRom [9]. However, the diverse limb configurations of parallel robots, unlike conventional serial robots, introduce significant complexity to the design process, often leading to prolonged development cycles.

Check for updates

Academic Editor: Xinjun Liu

Received: 9 December 2025
Revised: 28 December 2025
Accepted: 30 December 2025
Published: 31 December 2025

Copyright: © 2025 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.

Kinematic analysis is the cornerstone of parallel robot design, involving a structured sequence: motion pattern analysis, forward and inverse kinematics analysis, singularity analysis, kinematic performance evaluation and kinematic performance-based dimensional synthesis. Current research trends show a growing trend towards standardization in the methods used for kinematic analysis of novel configurations. While the Denavit–Hartenberg (DH) method [10] remains a viable approach, the screw theory [11–13] has become the prevailing method for analyzing motion patterns in most newly developed mechanisms. The closed-chain characteristics of parallel robots facilitate universal application of closed-loop vector methods for inverse kinematics solutions [14,15]. Beyond workspace, kinematic performance is also evaluated by motion/force transmission characteristics and dexterity. These latter two metrics are incorporated into dimensional synthesis to optimize the robot's overall performance. The motion/force transmission index aims to enhance the transfer of motion and forces between actuators and the moving platform during the design phase of parallel robots [16,17]. Dexterity, on the other hand, encompasses posture-dependent indices [18], such as the manipulability index [19] and condition number [13,20], which indicate the robot's stability in a particular posture. Integrating and digitizing kinematic analysis could significantly reduce the design cycle for parallel robots.

Many research institutions have contributed to digitizing the design process of parallel robots. G. Bianchi et al. [21] pioneered an integrated rapid design software for parallel mechanisms that combines structural design, numerical computation, finite element analysis and remote monitoring software. The developed VPE-PKM platform integrates fundamental theoretical knowledge of robot design and allows for reachable workspace estimation, Jacobian matrix calculation, stiffness estimation and elastic mechanics analysis. The platform represents a significant step towards comprehensive design tools for parallel robots. However, its current focus on 3-DOF parallel robots limits its applicability to a narrow range of designs. Ding's team [22–24] developed automated configuration sketching techniques, culminating in a 5-DOF parallel mechanism synthesis toolbox that employs standardized limb decomposition into rotational/translational joint combinations, enabling intuitive configuration retrieval and visualization. Wang et al. [25] introduced an analysis feature model aimed at the integration of commercial CAD and CAE software. By employing feature mapping, this model automatically converts analytical features into script codes for finite element analysis (FEA), enabling structural optimization of machine tools. Additionally, Zhang et al. [26,27] utilized Java and Java3D to develop a suite of virtual design toolboxes tailored for tripod-like parallel mechanisms. These toolboxes include functionalities for analyzing the dynamic performance of 3-DOF kinematics, dynamics analysis, statics optimization and structural optimization. Rega et al. [28] proposed a method for modeling and virtually testing complex machine tools. They integrated the kinematic inverse solution of the P800 parallel machine tool into a secondary developed CAD toolbox to simulate the testing trajectory of the machine tool. Kocak et al. [29] focused on a user-friendly graphical user interface (GUI) for the structural synthesis of parallel manipulators. They utilized distinct mobility formulations, introduced variable constrained loops and showcased the design toolbox of single-platform manipulators with the capacity for scalability to multi-platform configurations.

Critical analysis of existing tools reveals a predominant focus on structural optimization and dynamic analysis of specific mechanisms, with kinematic analysis typically embedded as isolated modules. Moreover, research exploring the relationship between topological structures and performance remains insufficient [30]. These toolboxes are often unable to facilitate the selection of configurations based on kinematic performance requirements. Consequently, this study aims to develop an integrated toolbox for the kinematic analysis and dimensional synthesis of multiple parallel robot configurations. The growing

standardization of kinematic modeling methodologies and performance metrics provides a theoretical foundation for this development.

The subsequent sections present three key aspects: Section 2 establishes a unified model utilized in each functional module, outlining the expression methods for various kinematic models. Section 3 outlines the toolbox architecture, encompassing software framework, functional modules, HMI design, core technologies and implementation pathways. Section 4 demonstrates system extensibility through complete integration and analysis of a redundantly actuated parallel robot. The paper concludes with a summary of research contributions.

2. Basic Theory

A prerequisite for conducting the kinematic analysis of parallel robots is the establishment of the kinematic models corresponding to the configurations. However, there are various configurations of parallel robots, each of which is described by a unique kinematic model. In this section, each kinematic model is designated by a specific modeling approach, so that the standardized model representation for data exchange is obtained. Upon integration into the database, models established via this mode are availed to each layer for direct invocation through interfaces. Through this standardized model expression mode, it becomes feasible to use the expansion of the database directly in the kinematic analysis.

2.1. Mobility Analysis Model

A hierarchical modeling strategy is employed to establish the mobility analysis framework for serial-parallel hybrid mechanisms. As illustrated in Figure 1, the architectural decomposition of parallel robots comprises three fundamental components: parallel kinematic chains, serial kinematic limbs and elementary joints. The serial kinematic limbs integrate five fundamental joint types: revolute (R), prismatic (P), universal (U), cylindrical (C) and spherical (S) joints. These serial configurations, including but not limited to UPR, RPS and UPS arrangements, demonstrate dual functionality. They can either directly configure parallel robotic systems or serve as composite parallel joints when integrated into parallel kinematic limbs. Two representative examples are the 3R(2SS)-type and the 2RPS-2UPR-type parallel robots presented in Figure 1. The parametric formulation of mechanism motion characteristics is derived through screw theory principles.

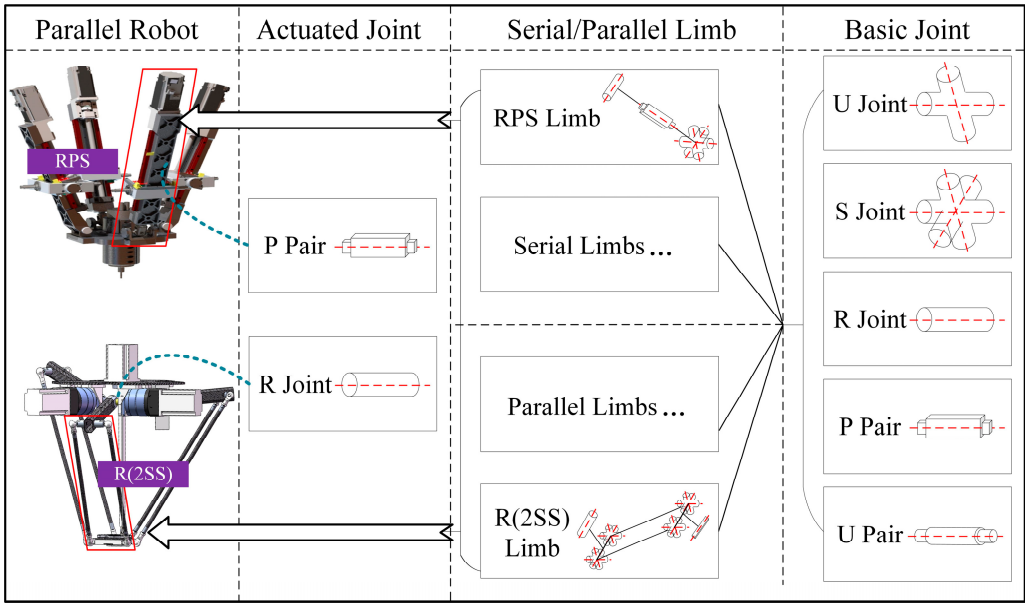


Figure 1. Hierarchical decomposition architecture of parallel robots.

The joint motion screw can be mathematically expressed as:

$$\$^r = (S; S_0) = (S; r \times S + pS) \tag{1}$$

where the unit vector S denotes the directional vector of the motion pair axis, and S_0 represents the moment vector comprising both rotational ($r \times S$) and translational (pS) components. The position vector r specifies an arbitrary point along the screw axis, while the scalar parameter p indicates the screw pitch.

The fundamental screw representations are categorized as:

- Translational screw $\$_t = (0; S)$;
- Rotational screw $\$_r = (S; r \times S)$.

To maintain mathematical generality and ensure computational compatibility with digital analysis toolboxes, the screw matrix formulation for an n-degree-of-freedom (DOF) kinematic limb follows systematic representation rules:

$$\{\$\} = \begin{Bmatrix} \$_1 \\ \$_2 \\ \cdots \\ \$_n \end{Bmatrix} = \begin{bmatrix} S_1; r_1 \times S_1 \\ 0; r_2 \\ \cdots \\ S_n; r_n \times S_n \end{bmatrix} \tag{2}$$

The constraint characteristics of parallel mechanisms adhere to dual principles:

(a) The constraint screw system of a parallel mechanism constitutes the union set of constraint screws from all constituent limbs:

$$\left\{\$_T^r\right\} = \left\{\$_1^r\right\} \cup \left\{\$_2^r\right\} \cup \left\{\$_3^r\right\} \cup \ldots \cup \left\{\$_n^r\right\} \tag{3}$$

(b) Conversely, the motion screw system of a parallel mechanism represents the intersection set of motion screws from all limbs:

$$\left\{\$_T^m\right\} = \left\{\$_1^m\right\} \cap \left\{\$_2^m\right\} \cap \left\{\$_3^m\right\} \cap \ldots \cap \left\{\$_n^m\right\} \tag{4}$$

Moreover, practical implementation challenges arise from the high dimensionality of screw systems in complex mechanisms. To address this computational complexity, an antiscrew formulation approach is recommended. The constraint screw system derivation initiates with constituent limb constraints, followed by subsequent antiscrew transformation to obtain the complete parallel constraint system. Through linear independence analysis of the derived constraint screws, the parallel motion screw system can be systematically obtained via inverse antiscrew calculation.

2.2. Kinematic Analysis Mode

Through integration of joint axis geometric relationships with the motion characteristics established in Section 2.1, the inverse kinematic solution formulation for parallel robotic systems is derived, explicitly relating actuator displacements to the moving platform's pose parameters.

For generality analysis, consider the representative parallel mechanism configuration shown in Figure 2. Several coordinate systems are established. A fixed coordinate system $O - xyz$ is established at the center of the fixed platform, and a tool coordinate system $O' - x'y'z'$ is set up at the center of the moving platform. A_i and B_i represent the movement joint centers of the limb on the fixed and moving platforms, respectively, with corresponding unit position vectors a_i and b_i. The position vector of O' is $r_{o'}$. The vector $\overrightarrow{A_iB_i}$ is defined as q_i. $S_{i,j}$ represents the direction vector of the jth moving axis of the i limb,

where the moving axis fixed to the fixed platform is denoted as $S_{i,1}$. The moving axis fixed to the moving platform is denoted as $S_{i,n}$. The limb possesses a total of n moving joint axes. In the $O' - x'y'z'$ coordinate system, the axis line $S_{i,n}$ is represented as $S_{i,n}^{o'}$.

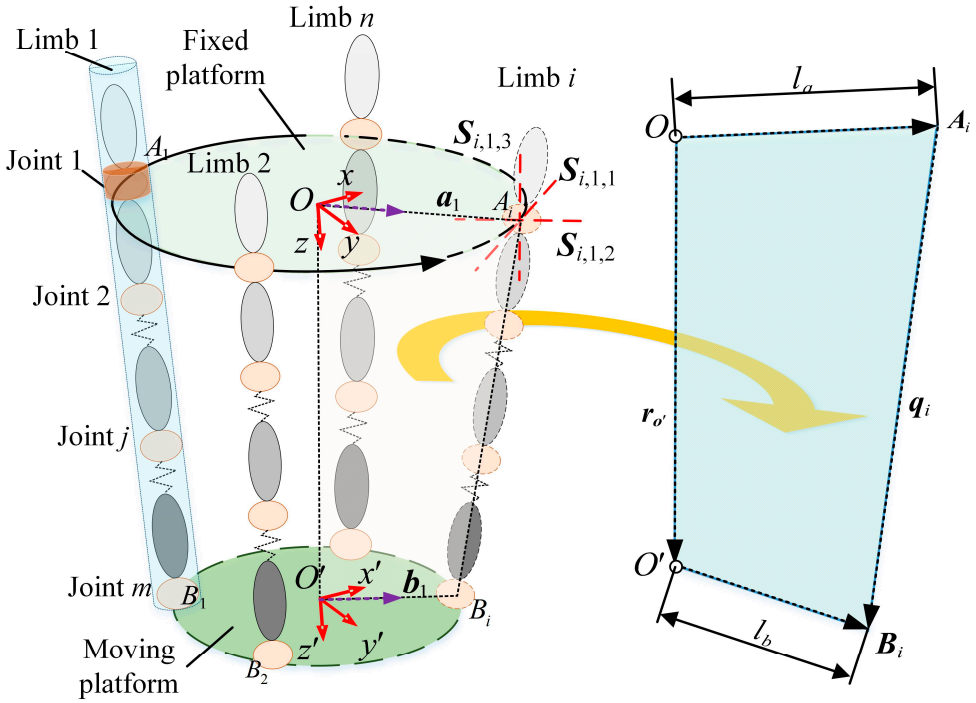


Figure 2. Generalized parallel mechanism configuration with kinematic parameterization.

The key axis direction vectors are represented as follows

$$\boldsymbol{r}_{o'} = (x_{o'} \quad y_{o'} \quad z_{o'}),\ \boldsymbol{S}_{i,1} = (x_{i,1} \quad y_{i,1} \quad z_{i,1}),\ \boldsymbol{S}_{i,n} = \boldsymbol{R} \cdot S_{i,n}^{o'} \tag{5}$$

where $\boldsymbol{R}$ represents the attitude transformation matrix, with ψ, θ and φ respectively indicating the precession angle, nutation angle and rotation angle within the attitude transformation matrix. The independent parameters ψ, θ, φ, $x_{o'}$, $y_{o'}$ and $z_{o'}$ collectively define the position and orientation of the moving platform.

As illustrated in the general schematic diagram of the structure, each endpoint $\boldsymbol{A}_i$, $\boldsymbol{B}_i$ (where $i = 1, 2, \ldots, n$) in the kinematic limb forms a set of closed-loop vector equations with points O and O'.

$$\boldsymbol{r}_{B_i} = \boldsymbol{q}_i + \boldsymbol{A}_i = \boldsymbol{r}_{o'} + R \cdot \boldsymbol{B}_i \tag{6}$$

In this equation, the position vectors $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ can be expressed as

$$\boldsymbol{A}_i = l_a \cdot \boldsymbol{a}_i,\ \boldsymbol{B}_i = l_b \cdot \boldsymbol{b}_i \tag{7}$$

By synthesizing Equations (5)–(7), the inverse solution equation of the kinematic can be derived. The actuator displacement parameter $\boldsymbol{e}_i$ can be expressed in terms of dimensional and kinematic parameters as

$$\boldsymbol{e}_i = E_i(\psi, \theta, \varphi, \boldsymbol{r}_{o'}, \boldsymbol{A}_i, \boldsymbol{B}_i) \tag{8}$$

where E_i represents the instantaneous kinematic limb length.

The inverse kinematic solution procedure follows a systematic, screw theory-based approach commonly employed for parallel mechanisms [11,13]. First, the motion screws of all joints in each limb are expressed in a common coordinate frame (usually the fixed

frame *O*-*xyz*). Using the screw system formulation (Equation (2)), the instantaneous twist of the moving platform is decomposed into contributions from each joint screw. Because the prismatic (P) or revolute (R) joints in typical limbs are either actuated or passive, the twist-screw equation can be rearranged to isolate the unknown actuator rates.

For the closed-loop vector method illustrated in Figure 2, the geometric closure condition (Equation (6)) is combined with the orientation relationships given by the screw axes (Equation (5)). Substituting the platform pose parameters (ψ, θ, φ, $x_{o'}$, $y_{o'}$ and $z_{o'}$) and the limb geometry ($\boldsymbol{A}_i$, $\boldsymbol{B}_i$ and $\boldsymbol{S}_{i,j}$) into Equation 6 yields a set of scalar equations that can be solved for the actuator displacements $\boldsymbol{e}_i$ (Equation (8)). This procedure ensures that the inverse kinematic model is both geometrically consistent and algebraically compatible with the screw-based mobility analysis described in Section 2.1. A detailed exposition of this methodology can be found in standard screw theory references [31,32].

2.3. Dexterity Analysis Model

In parallel robotics, a manipulator achieves full actuation when possessing n independent DOF coordinated through n distinct actuators. For such fully actuated systems, dimensional synthesis employs the spatial average of local dexterity across the entire workspace as the primary optimization metric. This approach quantifies the system's capacity to maintain consistent motion precision throughout its operational envelope.

The end-effector's twist vector combines angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ and linear velocity $\boldsymbol{v} \in \mathbb{R}^3$ at the moving platform's centroid. On the ith limb, $\hat{\mathbf{S}}_{ta,i,j}$ represents the movable screw of the jth joint, while $\hat{\mathbf{S}}_{tr,i,j}$ represents the immovable screw of the j_rth joint. The moving platform's resultant twist derives from limb contributions:

$$\hat{\boldsymbol{\$}} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = \sum_{j=1}^{d_i} q_j \cdot \hat{\mathbf{S}}_{ta,i,j} + \sum_{jr}^{6-di} 0 \cdot \hat{\mathbf{S}}_{tr,i,j_r} \tag{9}$$

where d_i indicates the ith limb's DOF count and $\boldsymbol{q}_j$ denotes the jth actuator's velocity.

Here, $\hat{\mathbf{S}}_{wa,i,k_n}$ signifies the driving screw transmitted by the ith limb to the moving platform, which exhibits duality with the kth movable screw while maintaining orthogonality to all other movable screws. Similarly, $\hat{\mathbf{S}}_{wr,i,k_r}$ represents the constraint screw provided by the ith limb, demonstrating orthogonality to all movable screws. These relationships satisfy the following conditions:

$$\hat{\mathbf{S}}_{wa,i,k_n} \cdot \hat{\mathbf{S}}_{ta,i,j} = 0,\ \hat{\mathbf{S}}_{wr,i,k_r} \cdot \hat{\mathbf{S}}_{ta,i,j} = 0 \tag{10}$$

The driving and constraint Jacobian matrices of this actuation system are, respectively, formulated as follows:

$$\begin{aligned} \boldsymbol{J}_a &= \begin{bmatrix} \hat{\mathbf{S}}_{wa,1,k_1}^T / \hat{\mathbf{S}}_{wa,1,k_1}^T \hat{\mathbf{S}}_{ta,1,k_1}^T \\ \hat{\mathbf{S}}_{wa,2,k_2}^T / \hat{\mathbf{S}}_{wa,2,k_2}^T \hat{\mathbf{S}}_{ta,2,k_2}^T \\ \cdots \\ \hat{\mathbf{S}}_{wa,n,k_n}^T / \hat{\mathbf{S}}_{wa,n,k_n}^T \hat{\mathbf{S}}_{ta,n,k_n}^T \end{bmatrix}, \boldsymbol{J}_r = \begin{bmatrix} \boldsymbol{J}_{r_1} \\ \boldsymbol{J}_{r_2} \\ \cdots \\ \boldsymbol{J}_{r_n} \end{bmatrix}, \\ \boldsymbol{J}_{r,i} &= \begin{bmatrix} \hat{\mathbf{S}}_{wa,i,1}^T / \hat{\mathbf{S}}_{wr,i,1}^T \hat{\mathbf{S}}_{tr,i,1}^T \\ \hat{\mathbf{S}}_{wa,i,2}^T / \hat{\mathbf{S}}_{wr,i,2}^T \hat{\mathbf{S}}_{tr,i,2}^T \\ \cdots \\ \hat{\mathbf{S}}_{wa,i,6-d_j}^T / \hat{\mathbf{S}}_{wr,i,6-d_j}^T \hat{\mathbf{S}}_{tr,i,6-d_j}^T \end{bmatrix} \end{aligned} \tag{11}$$

From Equation (11), the generalized Jacobian matrix $\boldsymbol{J}$ is derived as:

$$\begin{bmatrix} \boldsymbol{J}_a \\ \boldsymbol{J}_r \end{bmatrix} \cdot \hat{\boldsymbol{\$}} = \boldsymbol{J} \cdot \hat{\boldsymbol{\$}} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \cdots \\ q_{d_j} \\ \boldsymbol{0} \end{bmatrix} \tag{12}$$

As illustrated in Figure 2, three non-collinear geometric centers, $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$, corresponding to the limb-platform connection points, are selected as characteristic motion reference points. The position vectors $\boldsymbol{B}_1$, $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ in the moving platform coordinate system constitute the motion representation matrix.

$$\boldsymbol{J}_{pa0} = \begin{bmatrix} \boldsymbol{e} & \boldsymbol{e} & \boldsymbol{e} \\ \boldsymbol{B}_1 \times \boldsymbol{e} & \boldsymbol{B}_2 \times \boldsymbol{e} & \boldsymbol{B}_i \times \boldsymbol{e} \end{bmatrix}^T, \ \boldsymbol{e} = [0\ 0\ 1]^T \tag{13}$$

Through combination of Equations (12) and (13), the homogeneous Jacobian matrix $\boldsymbol{J}_{pa}$ for this fully actuated system is obtained:

$$\boldsymbol{J}_{pa} = \left[\boldsymbol{J}_{pa0} \cdot \boldsymbol{A}_{g0}^T \cdot \left((\boldsymbol{J})^T \cdot \boldsymbol{J} \right)^{-1} \cdot \boldsymbol{J}_a^T \right] \tag{14}$$

where

$$\boldsymbol{A}_{g0} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix} \tag{15}$$

For this fully actuated system, the local dexterity at the current pose is defined as the reciprocal of the condition number of the homogeneous Jacobian matrix. This relationship is expressed as follows:

$$d_p = 1 / \left(\left\| \boldsymbol{J}_{pa} \right\| \cdot \left\| \boldsymbol{J}_{pa} \right\| \right) = D_p(W, \{\boldsymbol{A}_i, \boldsymbol{B}_i\}) \tag{16}$$

In this equation, W denotes the set of workspace parameters defined in Equation (17), whereas $\{\boldsymbol{A}_i, \boldsymbol{B}_i\}$ encodes the dimensional data, specifically the relative coordinates of the joint endpoints, as detailed in Equation (8).

The dexterity analysis model presented in this section builds upon well-established screw theory-based Jacobian formulations and conditioning indices widely used in parallel robotics [33–35]. The dexterity analysis model is standardized into a reusable computational template that can be automatically instantiated for any parallel configuration stored in the toolbox database. This allows consistent, configuration-independent evaluation of kinematic dexterity, which is then directly used in the constrained dimensional synthesis described in Section 3.2. Key equations (Equations (9)–(16)) are expressed in a general form that accommodates hybrid limb compositions and redundant actuation, extending classical formulations to the heterogeneous architectures supported by the toolbox.

3. Digital Toolbox Design

3.1. Framework Description

To facilitate systematic configuration design, kinematic analysis and dimensional synthesis of various parallel robotic systems, a hierarchical digital toolbox architecture has been developed as illustrated in Figure 3. The framework comprises five functionally distinct strata: Interactive Layer, Data Layer, External Module Layer, Database Layer and Functional Layer, each characterized as follows.

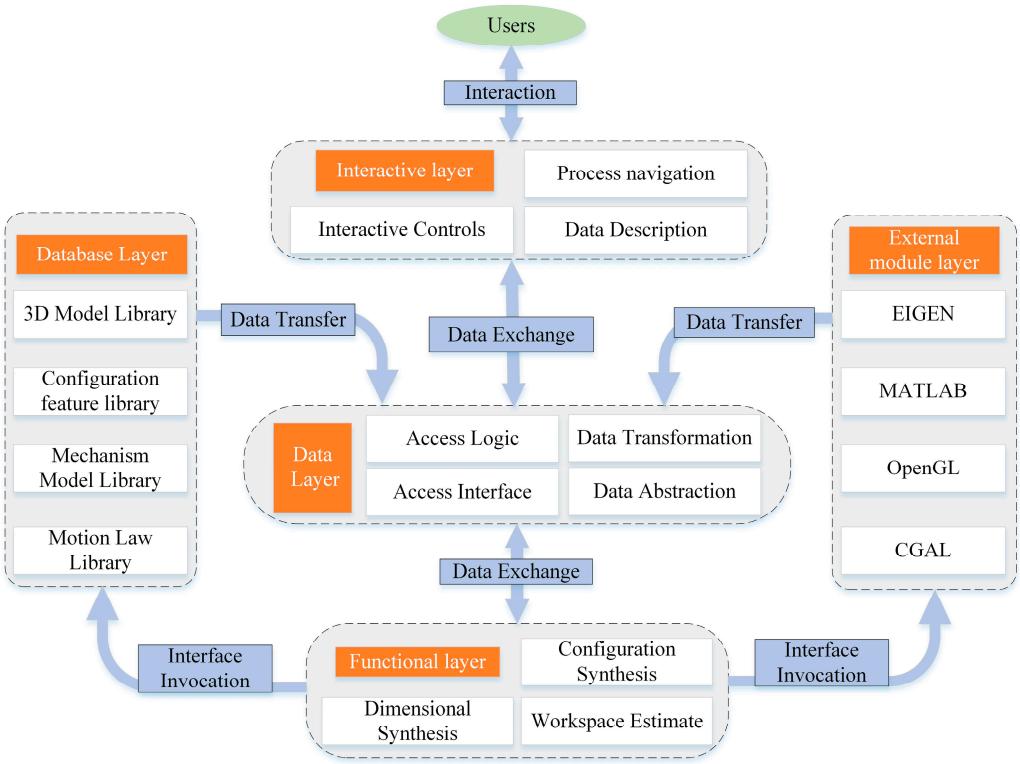


Figure 3. Multilayer architecture of the parallel robotics design toolbox.

(1) Interactive layer

The interactive layer incorporates an intuitive operating interface, acting as the bridge for user-to-function module interaction. It guides users throughout the design process, ensuring simplicity and efficiency. Through prompts and controls, this layer also upholds data accuracy.

(2) Data layer

Functioning as the centralized data repository, this layer governs:

- Parameter initialization (geometric constraints, material properties).
- Inter-module data exchange through standardized APIs.
- Persistent storage of intermediate states (XML-based serialization).

A hierarchical namespace architecture manages the computational workflow parameters, enabling efficient memory allocation and cross-layer data accessibility.

(3) External module layer

The external module layer bolsters crucial yet complex design and analysis functions. For complex data processing, it uses the EIGEN template library (offering efficient, flexible data structures and algorithms) and the CGAL library (providing computational geometry functionalities). For data visualization, it leverages the MATLAB algorithm library and EIGEN template library, which facilitate robust data visualization, analysis and management. Moreover, the OpenGL library is employed for 3D model visualization of parallel robots.

(4) Database layer

Structured as a relational database (SQLite implementation), this layer maintains:

- Kinematic chain templates (3-RRR, 6-SPS configurations).
- Performance metric archives (condition numbers, workspace volumes).
- Material property datasets (Young's modulus, inertia tensors).

Standardized XML schemas enable configuration portability across robotic platforms.

(5) Functional layer

The functional layer is the application layer for the three functional modules detailed in Section 3.2. It executes specific functions based on calls from other layers' interfaces.

Collectively, the five layers create a user-friendly, efficient and scalable design environment. The toolbox simplifies the complexities of parallel robot kinematic analysis and dimensional synthesis by accomplishing the following:

- Streamlining workflows through intuitive guidance and effective data management.
- Allowing the addition of new robot configurations and functionalities via its scalable architecture.
- Utilizing external tools for complex calculations and visualization, thereby advancing data analysis.

3.2. Functional Modules

As illustrated in Figure 4, three core computational modules—Configuration Synthesis Module (CSM), Workspace Estimation Module (WEM), and Dimension Synthesis Module (DSM)—are developed to establish bi-directional mappings between the topological parameters of parallel robots and their kinematic performance metrics. These modules enable systematic determination of dexterity indices and workspace characteristics through configuration-dependent analysis, as detailed below.

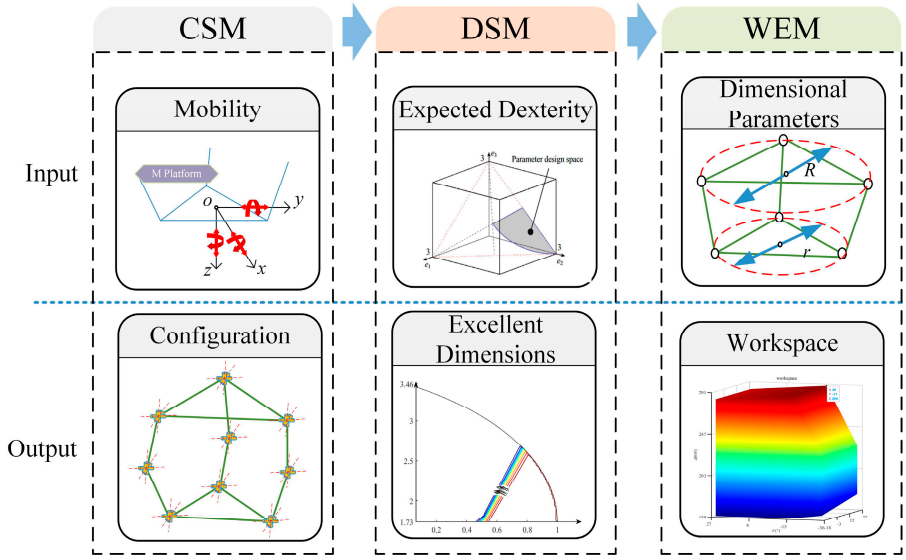


Figure 4. Functional modules linking design parameters to kinematic performance.

3.2.1. Configuration Synthesis Module (CSM)

The moving platform, serving as the end-effector mounting interface, exhibits motion characteristics constrained by the reciprocal screw systems of connected kinematic limbs. Direct determination of platform mobility through structural observation of individual limbs is precluded due to the following:

- Nonlinear kinematic coupling between limbs.
- Over-constrained motion transmission mechanisms.
- Configuration-dependent singularity conditions.

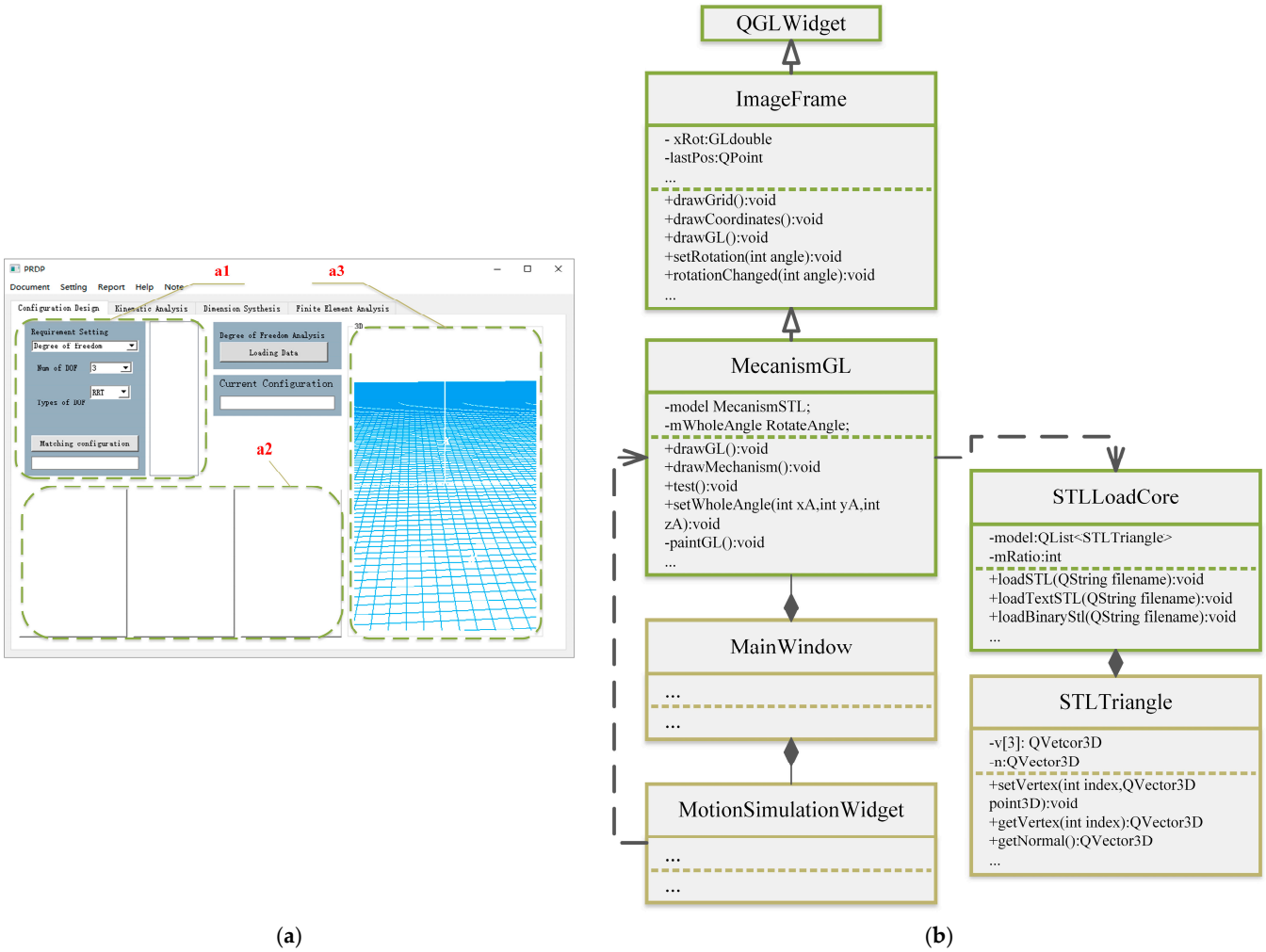
Direct determination of platform mobility through simple structural inspection of individual limbs is insufficient because of the following:

(1) Nonlinear kinematic coupling between limbs means that the motion of one limb affects the constraint system of others, making the overall mobility a function of the combined screw systems, not a sum of limb freedoms.

(2) Over-constrained motion transmission mechanisms (e.g., redundant constraints) can reduce the effective degrees of freedom in ways that are not visually apparent from the limb topology alone.

(3) Configuration-dependent singularity conditions can instantaneously change the mobility characteristics, meaning that a mobility analysis must be valid across the entire workspace, not just at a single pose.

Therefore, the toolbox performs mobility analysis via a systematic screw theory-based decomposition of limb interactions, as illustrated in Figure 5.



(**a**) (**b**)

Figure 5. Configuration synthesis module architecture: (**a**) user interface for CSM; (**b**) UML for 3D display functionality.

As depicted in Figure 5, mobility analysis is performed through constraint screw-based decomposition of limb interactions, enabling parametric extraction of platform motion characteristics via screw theory formalization.

As depicted in Figure 5a, the CSM comprises three integrated components:

(a) Feature Specification (a1):

Implements ontology-driven configuration tagging through the following:

- User-defined kinematic criteria (DOF types, limb topologies)
- Database query optimization via fuzzy-logic pattern matching.
- Open-source configuration labeling for community-driven expansion.
- Integrates modular components through:
- Parametric assembly of joint/limb/platform CAD models.

- Real-time OpenGL rendering pipeline (STL file compatibility).
- UML-based class relationship visualization (Figure 5b).

(b) Motion Demonstration (a3):
Animates kinematic behavior through the following:

- Screw-coordinate-based trajectory generation.
- Interactive DOF-specific motion decomposition.
- Grübler–Kutzbach criterion validation.

The configuration database undergoes continuous expansion through user-driven submissions, with novel configurations systematically categorized by the following:

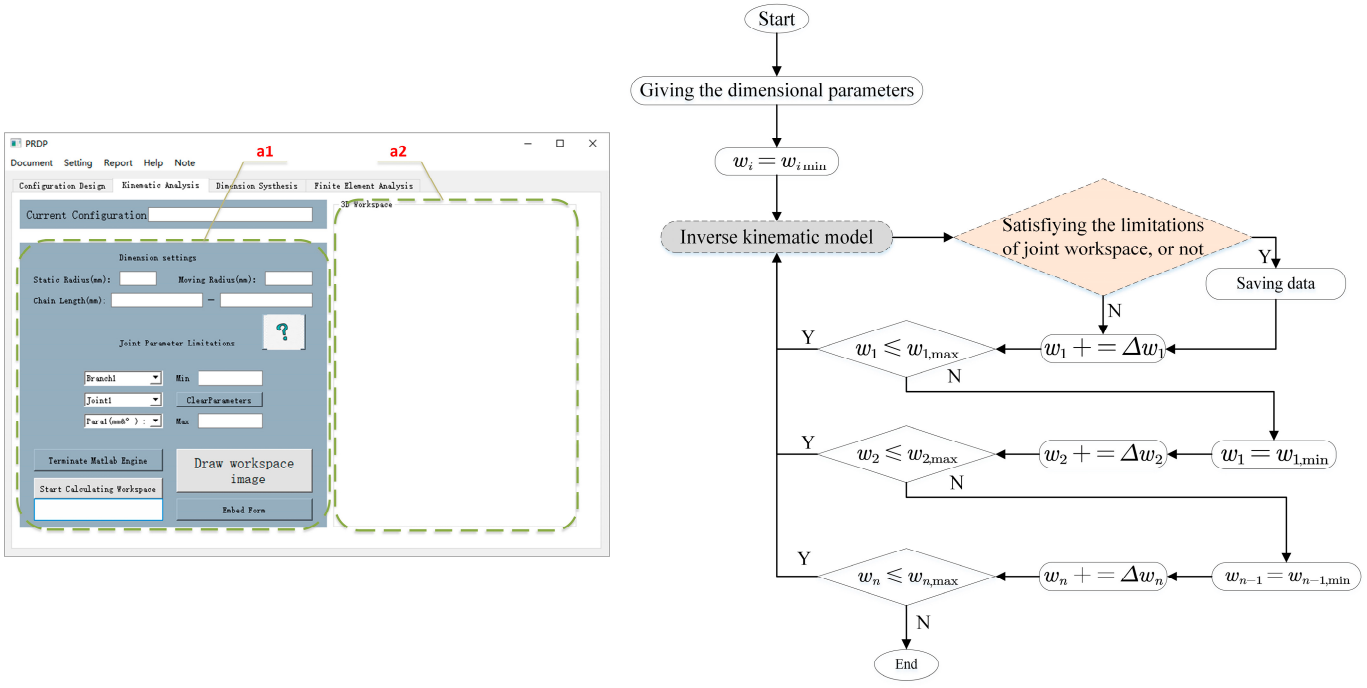
- DOF characteristics (translational/rotational constraints).
- Limb topology (serial/parallel chain configurations).
- Joint type (revolute/prismatic/spherical).

Validated configurations are propagated to downstream modules (WEM/DSM) through an XML-based data interchange, ensuring model consistency across the computational workflow.

3.2.2. Workspace Estimation Module (WEM)

As illustrated in Figure 6a, the WEM comprises two core computational components (a1 and a2). Component a1 enables manual parameter initialization, including the following:

- Dimensional constraints of kinematic chains.
- Joint motion boundary definitions.
- Component a2 subsequently performs automated workspace characterization through the following:
- Constraint-conditioned reachability analysis.
- Spatial occupancy visualization.



(**a**) (**b**)

Figure 6. Workspace computational workflow: (**a**) user interface for WEM; (**b**) workflow for workspace searching.

The workspace of a parallel robot is mathematically defined as follows:

$$W = \{(w_1, w_2, w_3 \cdots w_n) \mid P_t \in U_{pt}(t = 1, 2, 3 \ldots l), q_i \in U_{qi}(i = 1, 2, 3 \ldots m)\} \quad (17)$$

where

$w_i \in \mathbb{R}^3$: Platform pose parameters.

$\boldsymbol{P}_t$: Passive joint displacements (t = 1, 2, . . ., l).

$\boldsymbol{q}_i$: Active joint coordinates (i = 1, 2, . . ., m).

U_{pt}, $U_{qi} \in \mathbb{R}$: Joint motion constraints.

After the hierarchical search of w_1, w_2, . . ., w_n, the workspace can be estimated. The specific steps for this process are shown in Figure 6b and can be outlined as follows: First, the independent variables in the workspace are determined. Subsequently, the workspace is divided into n dimensions, and an n-layer ($1 < n < 6$) nested loop is established. Next, the coordinates of the current node are input into the inverse kinematic solution equation within a loop, resulting in the current dimension parameter. Finally, all workspace node coordinates satisfying the dimension range are output.

Singularity analysis is integrated into the workspace estimation process. For each candidate pose generated during workspace discretization, the homogeneous Jacobian matrix $\boldsymbol{J}_{pa}$ is computed. Its condition number $\kappa(\boldsymbol{J}_{pa}) > \kappa_{max}$ (where κ_{max} is a user-definable threshold, typically 10^3) is flagged as near-singular and excluded from the feasible workspace. This numerical check is supplemented by an analytical singularity-detection step based on the rank of the screw systems (Equations (3) and (4), ensuring that both local (pose-dependent) and architectural (configuration-dependent) singularities are identified.

As depicted in Figure 7, to overcome MATLAB's platform dependency and limited UI scalability, a hybrid programming framework is implemented via the following:

- Algorithm encapsulation: MATLAB core algorithms → DLLs via MATLAB Compiler SDK.
- Interface development: Qt-based GUI with COM interoperability.
- Data pipeline: XML-based parameter exchange (ISO/IEC 19775-1 compliant) [36].
- This architecture preserves MATLAB's computational advantages (e.g., matrix operations, visualization toolboxes) while enabling the following:
- Cross-platform deployment without MATLAB runtime.
- Memory-efficient C++ execution management.
- Seamless integration with robotic middleware (ROS-compatible).

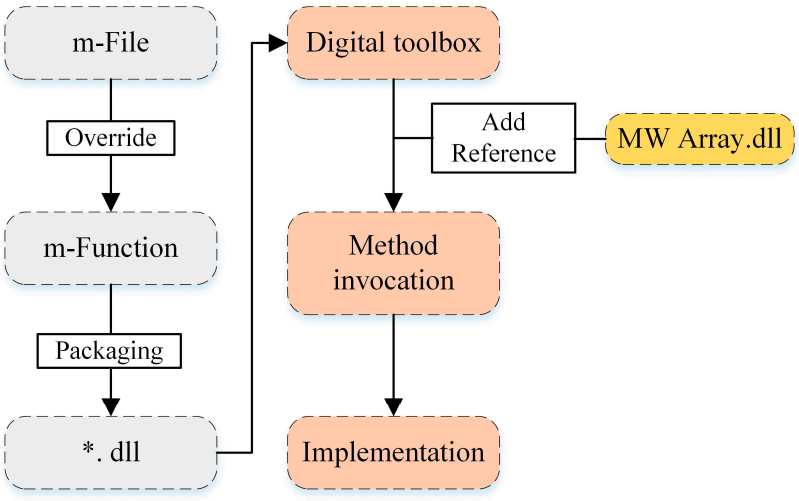
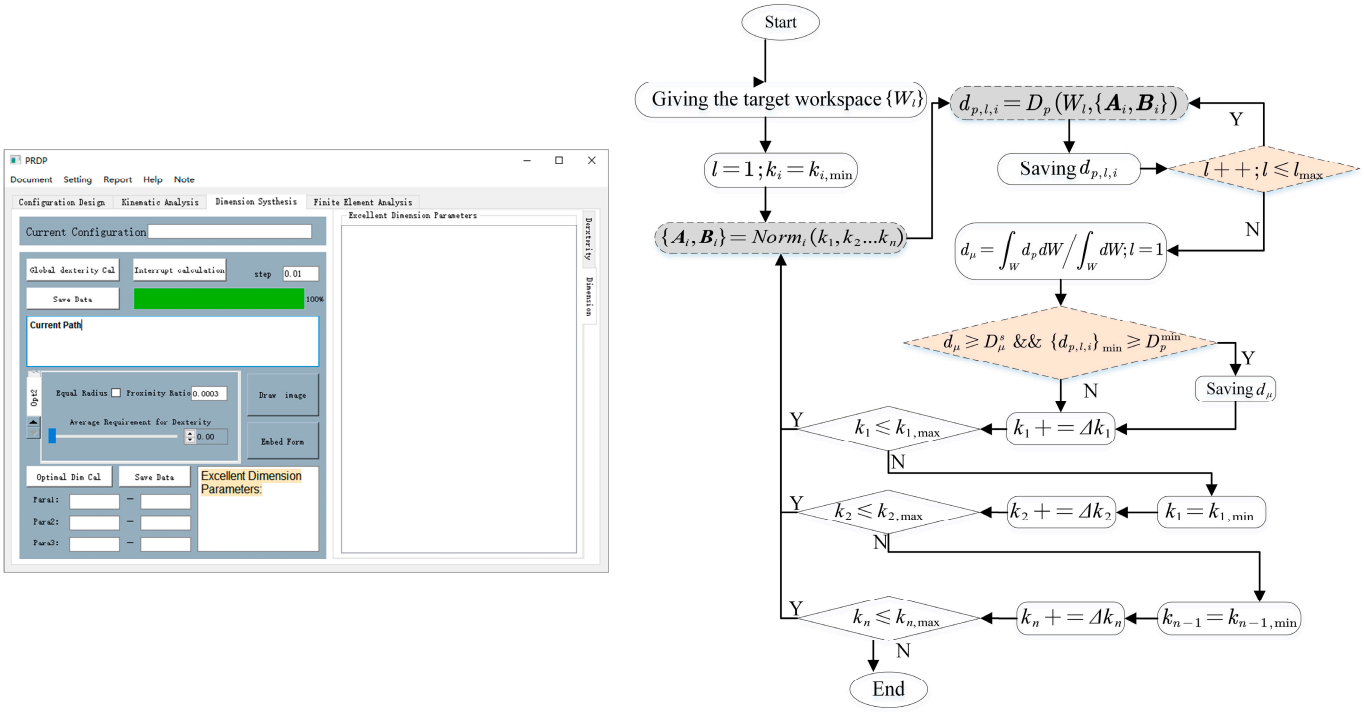


Figure 7. Heterogeneous Programming Interface Architecture.

3.2.3. Dimension Synthesis Module (DSM)

As depicted in Figure 8a, the DSM implements kinematic dexterity optimization through four integrated computational phases:

- Constraint Integration: Inherits joint motion limits from WEM outputs and computes global dexterity indices.
- Workspace Parameterization: Defines target workspace boundaries.
- Dexterity Mapping: Visualizes the spatial distribution of dexterity metrics.
- Dimensional Optimization: Selects Pareto-optimal solutions from candidate parameters.



(**a**) (**b**)

Figure 8. Dimension synthesis framework integrating workspace and dexterity constraints: (**a**) user interface for DSM; (**b**) workflow for screening the optimal dimension.

Given the configuration-dependent nature of dexterity metrics (d_p), a workspace-averaged index d_μ is formulated to quantify global performance:

$$d_\mu = \int_W d_p dW \Big/ \int_W dW \tag{18}$$

where $d_p dW$ signifies the global dexterity within the workspace; d_μ serves as the mean value of d_p. Moreover, the larger value of d_μ indicates better dexterity in the entire workspace.

Within a given workspace, the motion range of each passive joint affects d_μ. This value is also influenced by the motion range of active joints and the dimension parameters related to moving and fixed platforms. To enhance dexterity visualization within the workspace, various dimension parameters are normalized based on the architectural parameters of moving and fixed platforms, expressed as

$$\begin{cases} A_i = Norm_{ia}(k_1, k_2 \ldots k_n), (0 < n \le i) \\ B_i = Norm_{ib}(k_1, k_2 \ldots k_n), (0 < n \le i) \end{cases} \tag{19}$$

where k_m $(1 \le m \le n)$ denotes the normalization factor.

The procedure in Figure 8b constructs a workflow to search for the optimal d_μ and identify optimal dimension factors k_m. Gray boxes represent Equations (16) and (19), decoupled from other workflow steps. Orange boxes denote user-configurable parameters. Dexterity visualization in the DSM is enabled through a hybrid of C++ and MATLAB.

As shown in Figure 8b, the Dimension Synthesis Module (DSM) solves the following constrained optimization problem:

Given:

- A kinematic configuration (limb arrangement, joint types).
- A target workspace $\boldsymbol{W}_t$ (defined by ranges in position and orientation).
- Joint motion limits.
- Normalization factor $k_1, k_2, \ldots, k_n$.
- A lower bound on acceptable local dexterity $D_p^{\min}$.

Determine:

The set of normalized dimension factors $k_1, k_2, \ldots, k_n$ that save the global mean dexterity d_μ while ensuring that the resulting mechanism (1) fully encloses $\boldsymbol{W}_t$, (2) satisfies all joint motion limits, (3) respects the geometric constraints and (4) avoids singularities ($\kappa(\boldsymbol{J}_{pa}) > \kappa_{\max}$ throughout $\boldsymbol{W}_t$).

WEM predicts the workspace and corresponding joint motion range and dimension parameters, while the DSM selects optimal dimensions based on the workspace and the dexterity. When inputting a desired workspace and the minimum mean dexterity into the DSM, several sets of preferred dimensions can be obtained. Selecting the optimal dimension from this set and feeding it into the WEM determines the practical workspace for that specific dimension. Conversely, when dimension is a priority, set the dimension range in the WEM and input the resulting workspace data into the DSM to select the dimension with the best dexterity within the range.

3.3. Module Integration and Implementation Flow

The operational workflow of the digital toolbox framework (Section 3.1) is systematically implemented through the process shown in Figure 9, with sequential phases defined as follows.

(a) User Authentication and Configuration Screening:
Phase S1–S3: Initiate system access and navigate to the Configuration Selection Module.

- Cross-reference user-defined requirements against the kinematic feature database.
- Match identified: Extract valid configuration parameters for module initialization.
- No match: Trigger dynamic library expansion or requirement reevaluation.

(b) Kinematic Validation:
Phase S4–S5.

- Adjust multi-view 3D schematic perspectives (S4).
- Verify motion characteristics via interactive simulation (S5).

(c) Joint Constraint Propagation:
Phase S6.

- Derive joint mobility limits from validated kinematics.
- Synchronize constraints across WEMs and DSMs.

(d) Workspace Synthesis:
Phase S7–S8.

- Parametrize platform geometries (moving/fixed radii) in WEM.
- Generate configuration-specific workspace envelopes (S8).

(e) Dexterity-Driven Optimization:
Phase S9–S12.

- Input: Target workspace boundaries and structural constraints (S9).
- Compute: Global dexterity indices (S10) with database archiving (S11).
- Output: Manual selection of Pareto-optimal solutions via embedded algorithms (S12).

(f) Design Iteration Closure:
Phase S13.

- Option 1: Feed optimized parameters to WEM for workspace validation.
- Option 2: Terminate analysis with export-ready configuration data.

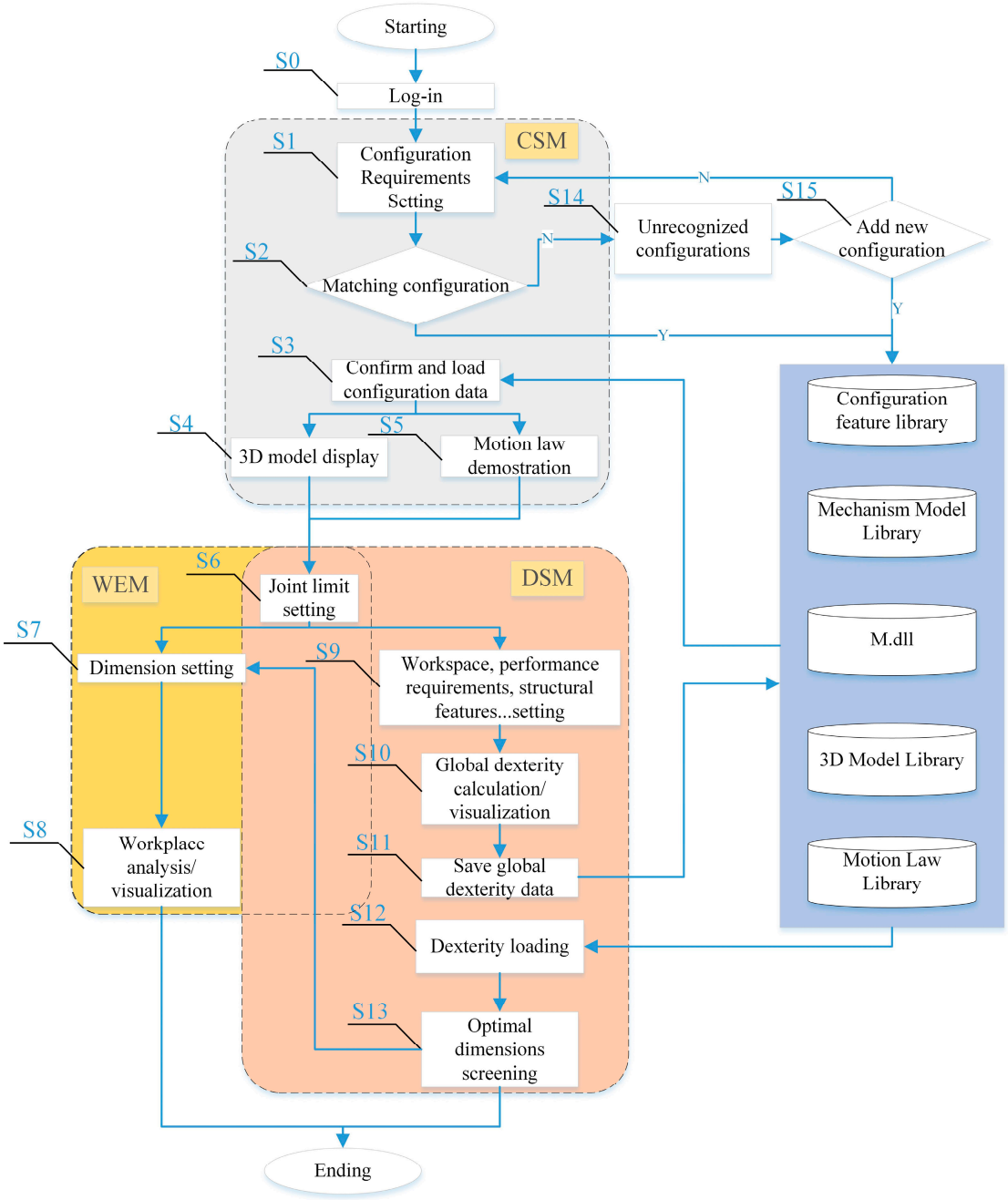


Figure 9. Toolbox implementation workflow architecture.

4. Case Study: 2UPR-2RPS Parallel Robot Implementation

4.1. Problem Description

To demonstrate the toolbox's capabilities, the novel 2UPR-2RPS parallel mechanism Figure 10 is implemented within the framework. Following the methodology in Section 3, we systematically develop its kinematic model for database integration.

Consistent with the structural parameterization methodology in Section 2.2, the geometric relationships governing joint axes are established through the following:

$$\begin{cases} S_{1,1} \parallel S_{1,5} \parallel S_{2,1} \parallel S_{2,5} \parallel S_{3,1} \parallel S_{4,1} \\ S_{1,4} \parallel S_{2,4} \parallel S_{3,4} \parallel S_{3,2} \parallel S_{4,4} \parallel S_{4,2} \\ Si,1 \perp Si,2 (i = 1,2) \\ S_{j,1} \perp S_{j,2}, S_{j,2} \perp S_{j,3}, S_{j,3} \perp S_{j,4}, S_{j,1} \perp S_{j,4} (j = 3,4) \end{cases} \tag{20}$$

In accordance with Equation (2), the motion screw system of each limb of the parallel robot is represented as

$$\hat{\$}_1 = \begin{cases} [\boldsymbol{S}_{1,1}; \boldsymbol{r}_{1,1} \times \boldsymbol{S}_{1,1}] \\ [0; \boldsymbol{r}_{1,2}] \\ [\boldsymbol{S}_{1,3}; \boldsymbol{r}_{1,3} \times \boldsymbol{S}_{1,3}] \\ [\boldsymbol{S}_{1,4}; \boldsymbol{r}_{1,4} \times \boldsymbol{S}_{1,4}] \\ [\boldsymbol{S}_{1,5}; \boldsymbol{r}_{1,5} \times \boldsymbol{S}_{1,5}] \end{cases}, \hat{\$}_2 = \begin{cases} [\boldsymbol{S}_{2,2}; \boldsymbol{r}_{2,1} \times \boldsymbol{S}_{2,1}] \\ [0; \boldsymbol{r}_{2,2}] \\ [\boldsymbol{S}_{2,3}; \boldsymbol{r}_{2,3} \times \boldsymbol{S}_{2,3}] \\ [\boldsymbol{S}_{2,4}; \boldsymbol{r}_{2,4} \times \boldsymbol{S}_{2,4}] \\ [\boldsymbol{S}_{2,5}; \boldsymbol{r}_{2,5} \times \boldsymbol{S}_{2,5}] \end{cases},$$
$$\hat{\$}_3 = \begin{cases} [\boldsymbol{S}_{3,1}; \boldsymbol{r}_{3,1} \times \boldsymbol{S}_{3,1}] \\ [\boldsymbol{S}_{3,2}; \boldsymbol{r}_{3,2} \times \boldsymbol{S}_{3,2}] \\ [0; \boldsymbol{r}_{3,3}] \\ [\boldsymbol{S}_{3,4}; \boldsymbol{r}_{3,4} \times \boldsymbol{S}_{3,4}] \end{cases}, \hat{\$}_4 = \begin{cases} [\boldsymbol{S}_{4,1}; \boldsymbol{r}_{4,1} \times \boldsymbol{S}_{4,1}] \\ [\boldsymbol{S}_{4,2}; \boldsymbol{r}_{4,2} \times \boldsymbol{S}_{4,2}] \\ [0; \boldsymbol{r}_{4,3}] \\ [\boldsymbol{S}_{4,4}; \boldsymbol{r}_{4,4} \times \boldsymbol{S}_{4,4}] \end{cases} \tag{21}$$

Through reciprocal screw analysis (Equation (3)), the constrained screw system satisfies the following:

$$\hat{\$}_T^r = \begin{cases} [\boldsymbol{S}_{1,3}; \boldsymbol{r}_{1,3} \times \boldsymbol{S}_{1,3}]\} = \hat{\$}_1^r \\ [\boldsymbol{S}_{2,3}; \boldsymbol{r}_{2,3} \times \boldsymbol{S}_{2,3}]\} = \hat{\$}_2^r \\ \left.\begin{matrix} [0; \boldsymbol{S}_{3,1}] \\ [\boldsymbol{S}_{3,2}; 0] \end{matrix}\right\} = \hat{\$}_3^r \\ \left.\begin{matrix} [0; \boldsymbol{S}_{4,1}] \\ [\boldsymbol{S}_{4,2}; 0] \end{matrix}\right\} = \hat{\$}_4^r \end{cases} \tag{22}$$

The mechanism's motion characteristics are determined via linearly independent reciprocal screw systems (Equation (4))

$$\hat{\$}_T^m = \begin{cases} [0; \boldsymbol{S}_{3,1}] \\ [\boldsymbol{S}_{3,2}; \boldsymbol{r}_{1,3} \times \boldsymbol{S}_{3,2}] \\ [\boldsymbol{S}_{3,3}; \boldsymbol{r}_{1,1} \times \boldsymbol{S}_{3,3}] \end{cases} \tag{23}$$

This result indicates that the mechanism possesses two independent rotational degrees of freedom (about the x- and y-axes) and one translational degree of freedom (along the z-axis), which aligns with the intended 2R1T motion pattern.

This expression reveals that the motion behavior of this mechanism corresponds to a 2R1T pattern that corresponds to the translational motion along the z-axis and rotational motion around the x- and y-axes.

The kinematic relationship (Equation (5)) is expressed as follows:

$$\boldsymbol{S}_{i,1} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \boldsymbol{S}_{i,n} = R \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{24}$$

In accordance with Equation (20), the essential axial vector relationships are derived as

$$\boldsymbol{S}_{i,1}^{T} \cdot \boldsymbol{r}_{o'} = 0, \boldsymbol{S}_{j,4}^{T} \cdot \boldsymbol{r}_{o'} = 0, \boldsymbol{S}_{j,1}^{T} \cdot \boldsymbol{S}_{j,4} = 0 (i = 1,2;\ j = 3,4) \tag{25}$$

Substitute Equations (24) and (25) into Equations (6) and (8) with $\varphi = 0$, $x_{o'} = z_{o'} \cdot \tan\theta$, $y_{o'} = 0$, and the inverse kinematic solution equation for the 2UPR-2RPS parallel robot can be derived as

$$\begin{cases} e_1 = \sqrt{(z_{o'} \cdot \tan\theta + R_1 \cdot \cos\theta - R_3)^2 + (z_{o'} - R_1 \cdot \sin\theta)^2} \\ e_2 = \sqrt{(z_{o'} \cdot \tan\theta - R_1 \cdot \cos\theta - R_3/2)^2 + (z_{o'} + R_1 \cdot \sin\theta)^2} \\ e_3 = \sqrt{(z_{o'} \cdot \tan\theta - R_2 \cdot \cos\theta \cdot \sin\varphi)^2 + (-R_2 \cdot \cos\theta + R_4)^2 + (z_{o'} + R_2 \cdot \cos\theta \cdot \sin\theta)^2} \\ e_4 = \sqrt{(z_{o'} \cdot \tan\theta + R_2 \cdot \cos\theta \cdot \sin\varphi)^2 + (R_2 \cdot \cos\theta - R_4)^2 + (z_{o'} - R_2 \cdot \cos\theta \cdot \sin\theta)^2} \end{cases} \tag{26}$$

As a redundantly actuated mechanism, the 2UPR-2RPS parallel robot can be divided into four subsystems. The last three limbs form a driving subsystem when the first one is deactivated. The dexterity of this mechanism at any possible posture is as follows:

$$d_L^R = max\{d_{p1}, d_{p2}, d_{p3}, d_{p4}\} \tag{27}$$

According to the diamond-shaped moving platform and fixed base of the 2UPR-2RPS parallel robot, the fixed platform has radii R_1 and R_2, while the moving platform has radii R_3 and R_4 satisfying $R_1/R_2 = R_3/R_4$.

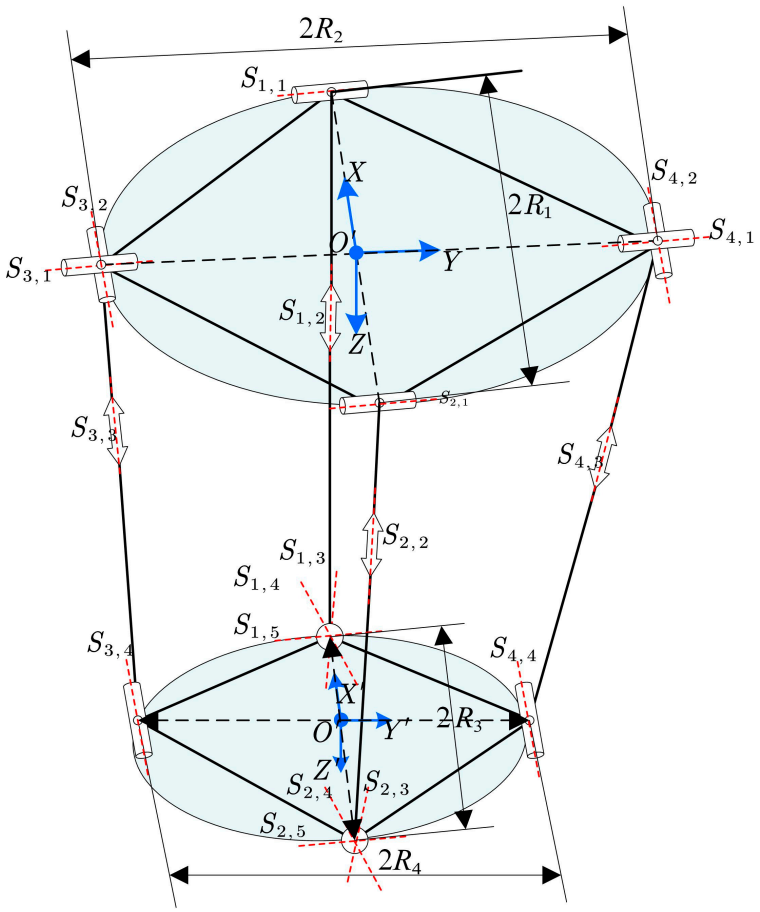


Figure 10. Topological configuration of the 2UPR-2RPS parallel mechanism.

For visualization, the parameter space is mapped onto a two-dimensional plane to outline the area. Considering the correlation between parameters in three-dimensional and two-dimensional spaces, the normalization factors $k_1, k_2 \ldots k_n (n = 4)$ in Equation (19) can now be represented as

$$\begin{cases} k_i = \frac{3R_i}{R_1+R_2+R_3} \\ l = \frac{\sqrt{3}(R_1+2R_2)}{R_1+R_2+R_3} \end{cases} \tag{28}$$

Joint coordinates of $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are parameterized via the following:

$$\left\{ \begin{array}{ll} \boldsymbol{A}_1 = \begin{bmatrix} R_1 & 0 & 0 \end{bmatrix} & \boldsymbol{B}_1 = \begin{bmatrix} R_3 & 0 & 0 \end{bmatrix} \\ \boldsymbol{A}_2 = \begin{bmatrix} -R_1 & 0 & 0 \end{bmatrix} & \boldsymbol{B}_2 = \begin{bmatrix} -R_3 & 0 & 0 \end{bmatrix} \\ \boldsymbol{A}_3 = \begin{bmatrix} 0 & R_2 & 0 \end{bmatrix} & \boldsymbol{B}_3 = \begin{bmatrix} 0 & R_4 & 0 \end{bmatrix} \\ \boldsymbol{A}_4 = \begin{bmatrix} 0 & -R_2 & 0 \end{bmatrix} & \boldsymbol{B}_4 = \begin{bmatrix} 0 & -R_4 & 0 \end{bmatrix} \end{array} \right\} \tag{29}$$

All equations are implemented as MATLAB functions following the standardized nomenclature in Table 1.

Table 1. MATLAB File Naming Convention.

Configuration (Function)	Equation
2UPR-2RPS (Mobility)	(23)
2UPR-2RPS (Inverse)	(26)
2UPR-2RPS (Norm)	(28), (29)
2UPR-2RPS ($\boldsymbol{J}_{pa}$)	(16), (27)

4.2. Kinematic Analysis and Dimensional Synthesis Procedure

The digital toolbox is validated on a workstation with an Intel Core i7-12700H processor, 16 GB RAM and an NVIDIA GeForce RTX 3060 GPU. Under typical workloads—rendering 3D models with 5k–15k polygons, manipulating point clouds of 50k–200k samples and updating dexterity maps interactively—the interface sustained frame rates $\geq$ 30 FPS during camera manipulation and $\geq$20 FPS during parameter-driven updates, with latency below 200 ms.

The software environment comprises a C++/Qt 5.15.2 GUI with an OpenGL 4.6 backend for rendering, a visualization pipeline that loads 3D models in STL format and renders them via Qt's OpenGL widget with shader-based lighting and depth testing, and a computation backend where MATLAB R2022b algorithms are compiled into DLLs using MATLAB Compiler SDK and invoked via a COM interface from the C++ application, all running on the NVIDIA Studio Driver 516.94.

The kinematic configuration parameters listed in Table 2 define the structural attributes required by the toolbox to identify or register a new mechanism. These parameters are specified by the user during the initial configuration screening phase (S1–S3).

Table 2. Kinematic configuration parameters.

DOF Setting		Limbs Setting		Joint Setting	
num	3	num	3	Including S	√
type	RRT	Redundantly	√	Active Joint	P

Upon accessing the digital toolbox and inputting the parameters from Table 2 into Phase S1, users encounter a configuration mismatch alert due to the absence of the 2UPR-2RPS model in the system database. To resolve this, the workflow necessitates the addition of the novel configuration through a dedicated interface. Within this window, users formally

designate the mechanism as "2UPR-2RPS" and upload its kinematic model file, enabling successful kinematic chain alignment. Post-integration, the 3D schematic and motion characteristics become accessible for visualization in Phases S4 and S5, as demonstrated in Figure 11.

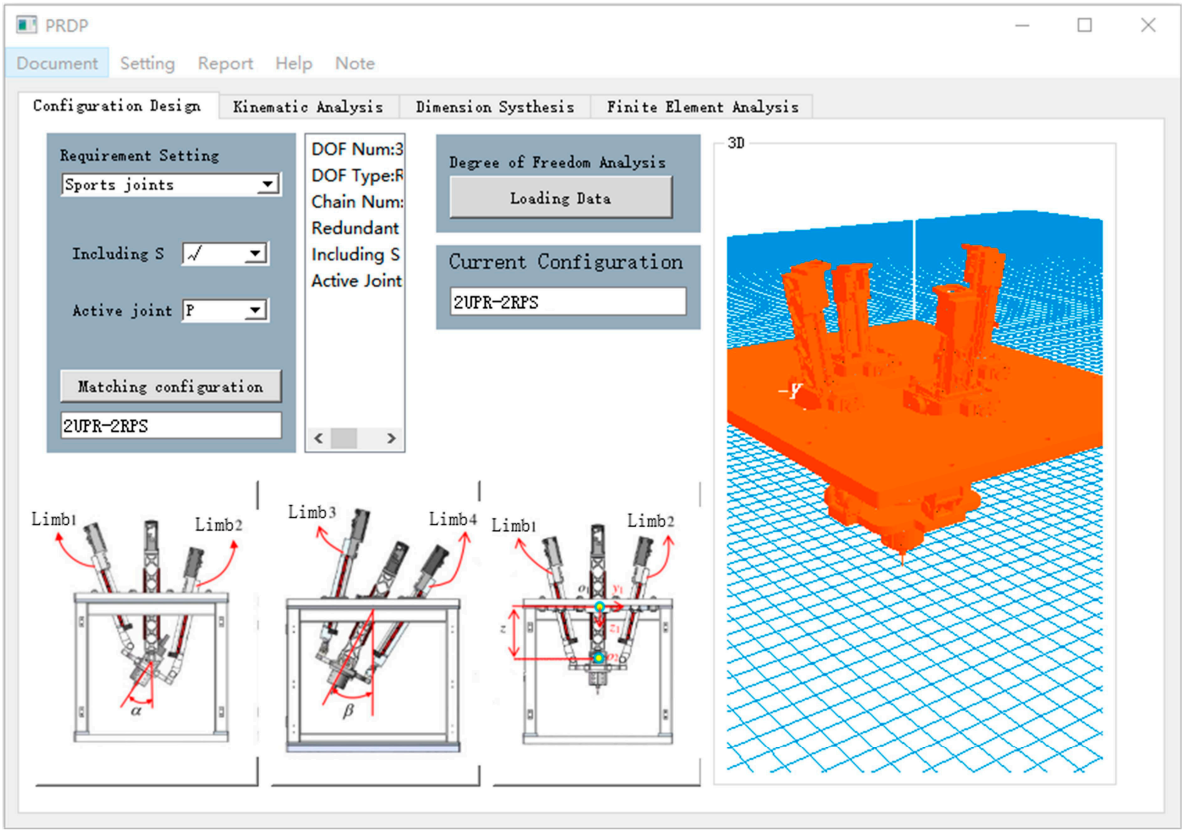


Figure 11. Dimension synthesis module interface with 2UPR-2RPS configuration.

Subsequently, the user can confirm the configuration, synchronize the WEMs and DSMs with the 2UPR-2RPS model, and, provided the fixed platform installation dimensions allow, set R_2 to 180 mm, input the required workspace and joint constraint range from Table 3 into Phases S6 and S9, and save the global dexterity data as a txt file. $P_{i,j,k}$ in Table 3 follow the same naming rule as $S_{i,j,k}$.

Following configuration validation, the system synchronizes the Workspace Estimation Module (WEM) and DSM with the newly added model. With fixed platform installation dimensions constrained, users configure R_2 = 180 mm while importing prescribed workspace boundaries and joint mobility ranges from Table 3 into Phases S6 and S9. The global dexterity dataset is subsequently archived in .txt format, adhering to the parameter indexing convention $P_{i,j,k}$, which mirrors the structural nomenclature $S_{i,j,k}$.

The joint-mobility constraints in Table 3 ensure that the optimized design respects physical limits of the actuators and passive joints. These values are derived from datasheets of commercial components or from design specifications, and they directly affect the feasible workspace boundaries.

Furthermore, the global dexterity graph for the current stage is generated, as depicted in Figure 12. Referring to the declining trend of mean dexterity in Figure 12, the global dexterity lower limit is set to 0.4 with the structural feature of equal radii for the moving and fixed platforms. The digital toolbox then filters dimensions meeting this criterion and displays the results in Figure 13.

Table 3. Joint mobility constraints.

Joint Constraint/ Workspace Range	Min (mm/Degree)	Max (mm/Degree)
$P_{1,1,1}$	20	160
$P_{1,2,1}$	155	340
$P_{1,3,1}$	40	140
$P_{1,3,2}$	40	140
$P_{1,3,3}$	40	140
$P_{2,1,1}$	20	160
$P_{2,2,1}$	155	340
$P_{2,3,1}$	40	140
$P_{2,3,2}$	40	140
$P_{2,3,3}$	40	140
$P_{3,1,1}$	40	140
$P_{3,1,2}$	40	140
$P_{3,2,1}$	155	340
$P_{3,3,1}$	20	160
$P_{4,1,1}$	40	140
$P_{4,1,2}$	40	140
$P_{4,2,1}$	155	340
$P_{4,3,1}$	20	160
$z_{o'}$	180	260
θ	−30	30
φ	−30	30

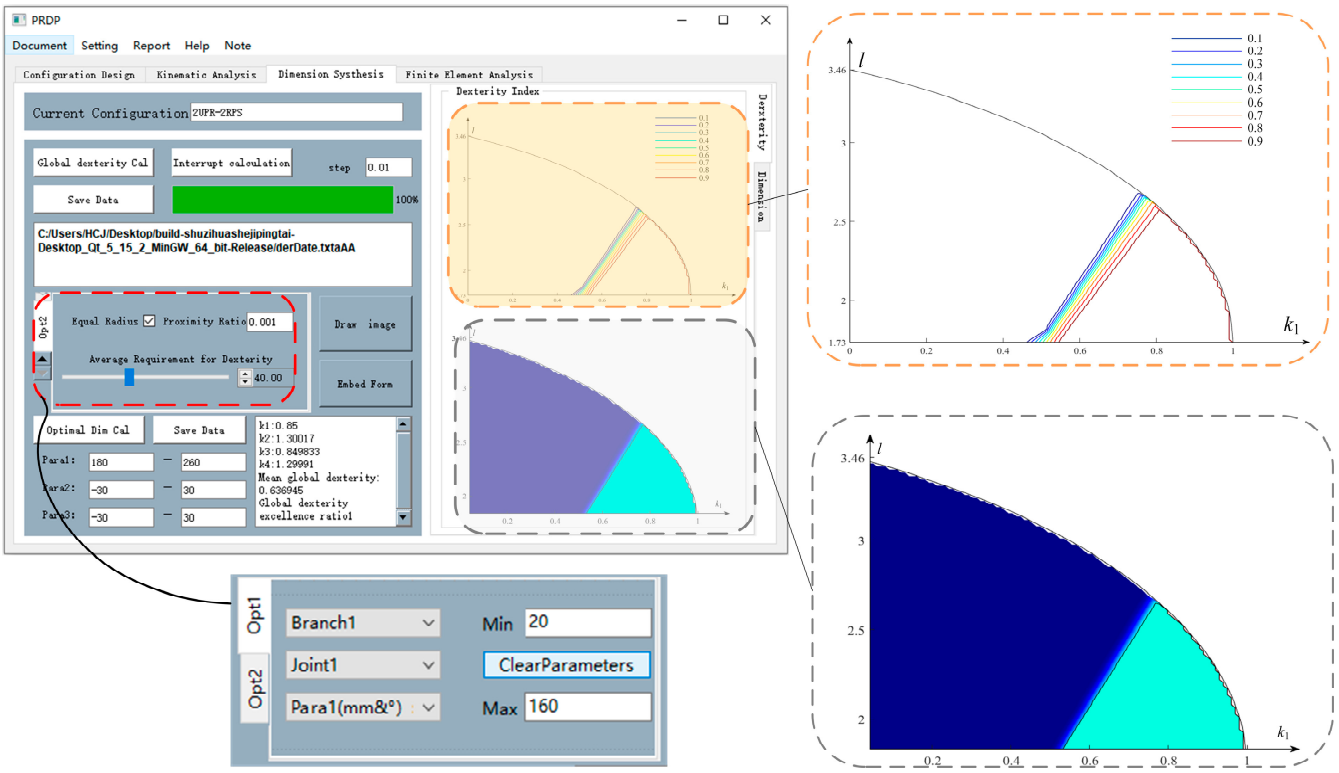


Figure 12. Temporal evolution of configuration-dependent dexterity indices.

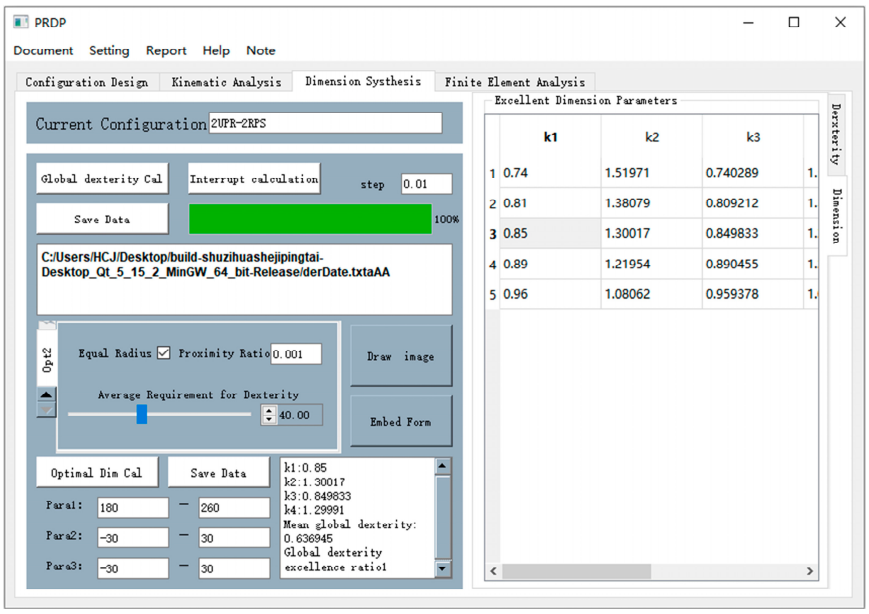


Figure 13. Multi-objective dimension synthesis results.

Figure 12 illustrates how the mean dexterity d_μ varies as the dimension factors k_m are iteratively adjusted during optimization. The declining trend beyond certain values helps the designer set a realistic lower bound for acceptable performance, which in this case is chosen as $d_\mu \geq 0.4$.

The normalized dimension factors k_1–k_4 were obtained through a constrained optimization process within the DSM. The optimization was subject to: (1) geometric proportionality $R_1/R_2 = R_3/R_4$; (2) preset installation limit $R_2 = 180$ mm; (3) joint mobility ranges listed in Table 3; (4) a predefined target workspace $z \in [180, 260]$ mm, $\theta \in [-30, 30]^\circ$ and $\phi \in [-30, 30]^\circ$, and (5) a lower dexterity threshold $d_\mu \geq 0.4$. The objective was to maximize the global mean dexterity d_μ while ensuring the target workspace remained fully reachable. The obtained symmetric solution $k_1 = k_3 = 0.85$, $k_2 = k_4 = 1.3$ satisfied all constraints and yielded $d_\mu = 0.637$, representing a Pareto-optimal trade-off between dexterity and geometric feasibility.

A set of dimension factors, $k_1 = 0.85$, $k_2 = 1.3$, $k_3 = 0.85$, and $k_4 = 1.3$, was identified as the optimal design result, yielding a d_μ value of 0.637. Based on the previously defined installation dimension R_1, the basic dimensional parameters of the prototype are summarized in Table 4.

Table 4. Optimized geometric parameters of the 2UPR-2RPS prototype.

R_1 (mm)	R_2 (mm)	R_3 (mm)	R_4 (mm)
118	180	118	180

Input the dimension synthesis data obtained from the DSM into the WEM to generate the actual three-dimensional workspace image, as depicted in Figure 14.

As evidenced by the workspace analysis in Figure 14, the 2UPR-2RPS parallel robot demonstrates bilateral symmetry about the $\theta = 0^\circ$ and $\varphi = 0^\circ$ planes, a geometric property inherently aligned with the topological symmetry of its kinematic chain configuration. Significantly, the synthesized workspace not only fully contains the predefined operational boundaries from dimensional synthesis but also strictly confines z-axis mobility to the stroke limits of the actuated prismatic (P) joints. These findings quantitatively validate that the analytical results derived via this digital toolbox exhibit full consistency with both

the prescribed design specifications and the mechanism's structural constraints, thereby confirming the toolbox's efficacy for parallel robotic system design and optimization.

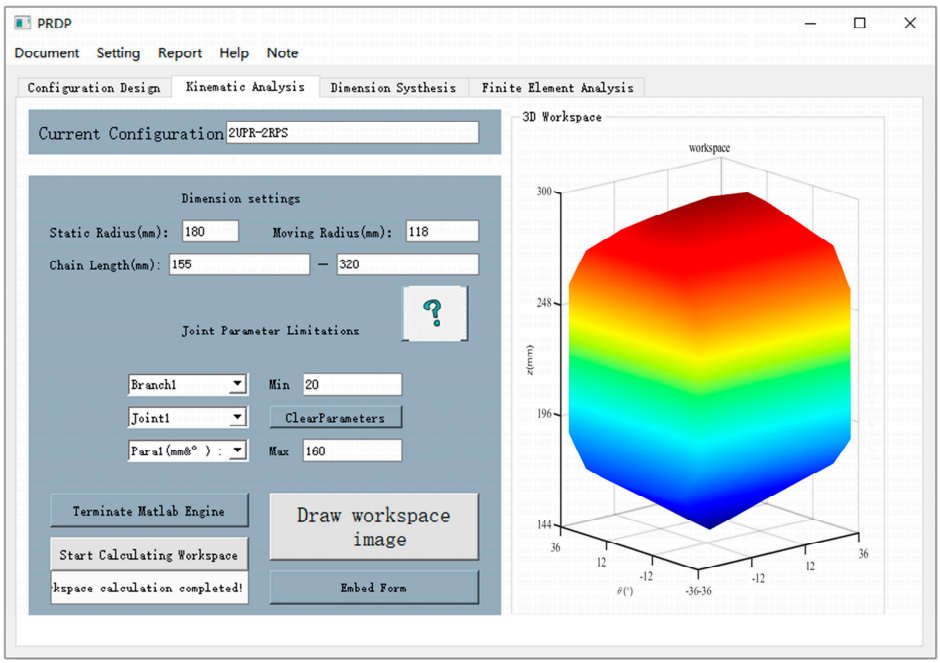


Figure 14. Workspace of 2UPR-2RPS.

As evidenced by the workspace analysis in Figure 14, the 2UPR-2RPS parallel robot demonstrates bilateral symmetry about the $\theta = 0^\circ$ and $\varphi = 0^\circ$ planes, a geometric property inherently aligned with the topological symmetry of its kinematic chain configuration. Significantly, the synthesized workspace not only fully contains the predefined operational boundaries from dimensional synthesis but also strictly confines z-axis mobility to the stroke limits of the actuated prismatic (P) joints. These findings quantitatively validate that the analytical results derived via this digital toolbox exhibit full consistency with both the prescribed design specifications and the mechanism's structural constraints, thereby confirming the toolbox's efficacy for parallel robotic system design and optimization.

4.3. Discussion and Future Extensions

To contextualize the performance of the toolbox-generated design, a comparative analysis was conducted against (1) a manually optimized version of the 2UPR-2RPS mechanism and (2) published 2R1T parallel mechanisms from the literature. The manual design process, involving iterative adjustment in standalone software, required ≈3 weeks and achieved a mean dexterity of 0.52 with 87% workspace coverage of the target volume. In contrast, the toolbox produced an optimized design in ≈2 h with a mean dexterity of 0.637 and 100% workspace coverage, illustrating substantial gains in both efficiency and kinematic quality. Compared to literature-reported 2R1T mechanisms—such as the 3-RPS [11] (mean dexterity ≈ 0.48–0.55) and the 3-UPS [14] (mean dexterity ≈ 0.50–0.60)—the toolbox-optimized 2UPR-2RPS design exhibits competitive or superior dexterity while offering the additional benefits of redundant actuation and inherent workspace symmetry. Architecturally, the toolbox differs from existing digital tools (e.g., VPE-PKM [21], Ding et al.'s synthesis toolbox [22–24]) by providing a unified kinematic-modeling environment that supports hybrid limb compositions, redundant actuation, and extensibility to higher DOF, thereby addressing a broader range of design scenarios.

While the presented toolbox and case study validate kinematic performance through simulation and numerical analysis, industrial applications would necessitate several ad-

ditional validation steps. These include: (1) experimental prototyping and physical accuracy/repeatability testing; (2) dynamic load-capacity and trajectory-tracking validation under operational conditions; (3) environmental robustness testing (thermal, vibrational, longevity); (4) integration with real-time control systems and closed-loop performance evaluation; (5) compliance with industrial safety standards and functional-safety certification; (6) manufacturability and cost-effectiveness analysis; and (7) interoperability testing with broader automation ecosystems. Future work will focus on extending the toolbox to support such validations, for example, by coupling with multi-body dynamics simulators, control-hardware-in-the-loop platforms, and digital-twin frameworks that bridge the gap between kinematic design and industrial deployment.

The case study of the 2UPR-2RPS parallel mechanism demonstrates the toolbox's ability to handle hybrid limb compositions, redundant actuation, and constrained dexterity optimization. However, the methodology is not limited to this configuration. Future work will focus on extending the toolbox in several directions: (1) expanding the kinematic database to include spherical, planar, and cable-driven parallel architectures; (2) integrating dynamic and stiffness performance metrics to support multi-domain design; (3) implementing a multi-objective optimization interface that explicitly trades off dexterity, workspace, singularity margin, and stiffness; (4) developing cloud-based collaborative features and digital-twin integration for real-time validation; and (5) releasing an open-source version to engage the research community in further development and application.

5. Conclusions

This study presents a digital toolbox that systematically addresses the design and optimization challenges of parallel robots through integrated kinematic modeling and dimension synthesis. The developed toolbox bridges theoretical screw-based analysis with practical implementation, offering a scalable solution for diverse robotic configurations. Contributions are summarized as follows:

- A screw theory-driven decomposition method standardizes mobility and constraint analysis, enabling rapid kinematic characterization of complex parallel mechanisms.
- The five-layer digital toolbox (interactive, data, external module, database, functional) ensures seamless integration of design, analysis, and optimization modules, supported by hybrid MATLAB-C++ programming for real-time visualization.
- Pareto-optimal dimension synthesis and XML-driven data interoperability enhance computational efficiency and design flexibility.
- Application to a 2UPR-2RPS robot demonstrates the toolbox's capability to optimize dimensional parameters ($k_1 = k_3 = 0.85$, $k_2 = k_4 = 1.3$) and validate workspace symmetry, achieving a mean dexterity index of 0.637.

Future work will expand the kinematic chain database, incorporate dynamic performance metrics, and integrate collaborative design features. This toolbox provides a pivotal tool for advancing parallel robotics in precision automation, reconfigurable systems, and industrial applications.

Author Contributions: Conceptualization, Z.H. and C.H.; methodology, Z.H., C.H. and T.T.; software, H.F. and Y.J.; validation, Z.H., T.T. and F.Y.; formal analysis, F.Y.; investigation, C.H.; writing—original draft preparation, Z.H.; writing—review and editing, F.Y. and J.Z.; visualization, Z.H. and C.H.; supervision, F.Y. and J.Z.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52375009, and the Natural Science Foundation of Fujian Province, grant number 2024J02007 and 2024J08147.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

DH	Denavit–Hartenberg
CAD	Computer-aided design
CAE	Computer-aided engineering
FEA	Finite element analysis
DOF	Degree of freedom
GUI	Graphical user interface
HMI	Human-machine interface
R	Revolute joint
P	Prismatic joint
U	Universal joint
C	Cylindrical joint
S	Spherical joint
CSM	Configuration synthesis module
WEM	Workspace estimation module
DSM	Dimension synthesis module
UML	Unified modeling language
XML	Extensible markup language
CGAL	Computational geometry algorithms library
SDK	Software development kit
COM	Component object model
ROS	Robot operating system

References

1. Wang, M.; Song, Y.; Lian, B.; Wang, P.; Chen, K.; Sun, T. Dimensional parameters and structural topology integrated design method of a planar 5R parallel machining robot. *Mech. Mach. Theory* **2022**, *175*, 104964. [CrossRef]
2. Dong, C.; Liu, H.; Huang, T.; Chetwynd, D.G. A screw theory-based semi-analytical approach for elastodynamics of the tricept robot. *J. Mech. Robot.* **2019**, *11*, 031005. [CrossRef]
3. Fang, H.; Tang, T.; He, Z.; Liu, Y.; Zhang, J. A novel hybrid machine tool integrating a symmetrical redundantly actuated parallel mechanism: Design, kinematics, prototype and experiment. *Mech. Mach. Theory* **2022**, *176*. [CrossRef]
4. Liang, D.; Liu, T.; Song, Y.; Chang, B.; Jin, G. Design, mathematical modeling and dynamic dimension synthesis for a modular parallel robot with end-articulated structure. *Appl. Math. Model.* **2025**, *144*, 116058. [CrossRef]
5. Chen, X.; Liu, X.-J.; Xie, F.; Sun, T. A Comparison Study on Motion/Force Transmissibility of Two Typical 3-DOF Parallel Manipulators: The Sprint Z3 and A3 Tool Heads. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 5. [CrossRef]
6. López-Custodio, P.; Fu, R.; Dai, J.; Jin, Y. Compliance model of Exechon manipulators with an offset wrist. *Mech. Mach. Theory* **2022**, *167*, 104558. [CrossRef]
7. Dong, C.L.; Li, J.T.; Liu, H.T.; Huang, T. Isotropy of Tangential Motion Transmissibility and Kinematic Performance Analysis of TriMule and Exechon Robots. *J. Mech. Eng.* **2021**, *57*, 23–32. [CrossRef]
8. Qin, X.; Li, Y.; Feng, G.; Bao, Z.; Li, S.; Liu, H.; Li, H. A novel surface topography prediction method for hybrid robot milling considering the dynamic displacement of end effector. *Int. J. Adv. Manuf. Technol.* **2024**, *130*, 3495–3508. [CrossRef]
9. Xie, Z.; Xie, F.; Liu, X.-J.; Wang, J.; Mei, B. Tracking error prediction informed motion control of a parallel machine tool for high-performance machining. *Int. J. Mach. Tools Manuf.* **2021**, *164*, 103714. [CrossRef]
10. Wang, J.; Liang, F.; Zhou, H.; Yang, M.; Wang, Q. Analysis of position, pose and force decoupling characteristics of a 4-UPS/1-RPS parallel grinding robot. *Symmetry* **2022**, *14*, 825. [CrossRef]
11. Tian, H.-B.; Wang, C.-Y.; Ma, H.-W.; Xia, J. Kinematic analysis and workspace investigation of novel 3-RPS/(H) metamorphic parallel mechanism. *J. Mech. Robot.* **2023**, *15*, 041008. [CrossRef]
12. Zhang, Y.; Wu, H.; Zhao, J.; Yan, S. Design and performance analysis of morphing nose cone driven by a novel bionic parallel mechanism for aerospace vehicle. *Aerosp. Sci. Technol.* **2023**, *139*, 108365. [CrossRef]

13. Liu, J.; Yang, Z.; Ding, H.; Li, M. Design and kinematical performance analysis of a novel reconfigurable parallel mechanism with three remote center-of-motion modes. *Mech. Mach. Theory* **2024**, *191*, 105513. [CrossRef]
14. Meng, Q.; Liu, X.-J.; Xie, F. Design and development of a Schönflies-motion parallel robot with articulated platforms and closed-loop passive limbs. *Robot. Comput. Integr. Manuf.* **2022**, *77*, 102352. [CrossRef]
15. Vargas-Riaño, J.; Agudelo-Varela, Ó.; Valera, Á. Applying Screw Theory to Design the Turmell-Bot: A Cable-Driven, Reconfigurable Ankle Rehabilitation Parallel Robot. *Robotics* **2023**, *12*, 154. [CrossRef]
16. Shi, J.; Li, R.; Guo, W. Configuration Synthesis and Performance Analysis of 1T2R Decoupled Wheel-Legged Reconfigurable Mechanism. *Micromachines* **2025**, *16*, 903. [CrossRef] [PubMed]
17. Shen, X.; Xu, L.; Li, Q. Motion/Force constraint indices of redundantly actuated parallel manipulators with over constraints. *Mech. Mach. Theory* **2021**, *165*, 104427. [CrossRef]
18. Ben, H.I.; Laribi, M.A.; Mlika, A.; Romdhane, L.; Zeghloul, S. Dimensional Synthesis and Performance Evaluation of Four Translational Parallel Manipulators. *Robotica* **2021**, *39*, 233–249. [CrossRef]
19. Stoughton, R.; Arai, T. A modified Stewart platform manipulator with improved dexterity. *IEEE Trans. Robot. Autom.* **1993**, *9*, 166–173. [CrossRef]
20. Angeles, J.; López-Cajún, C.S. Kinematic isotropy and the conditioning index of serial robotic manipulators. *Int. J. Robot. Res.* **1992**, *11*, 560–571. [CrossRef]
21. Bianchi, G.; Fassi, I.; Tosatti, L.M. PKM Analysis and design in a virtual prototyping environment. *J. Adv. Manuf. Syst.* **2002**, *1*, 211–226. [CrossRef]
22. Ding, H.; Huang, P.; Liu, J.; Kecskeméthy, A. Automatic structural synthesis of the whole family of planar 3-degrees of freedom closed loop mechanisms. *J. Mech. Robot.* **2013**, *5*, 041006. [CrossRef]
23. Ding, H.; Huang, P.; Zi, B.; Kecskeméthy, A. Automatic synthesis of kinematic structures of mechanisms and robots especially for those with complex structures. *Appl. Math. Model.* **2012**, *36*, 6122–6131. [CrossRef]
24. Ding, H.; Cao, W.; Cai, C.; Kecskeméthy, A. Computer-aided structural synthesis of 5-DOF parallel mechanisms and the establishment of kinematic structure databases. *Mech. Mach. Theory* **2015**, *83*, 14–30. [CrossRef]
25. Wang, J.; Niu, W.; Ma, Y.; Xue, L.; Cun, H.; Nie, Y.; Zhang, D. A CAD/CAE-integrated structural design framework for machine tools. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 545–568. [CrossRef]
26. Zhang, D.; Wang, L.; Lang, S.Y.T. Parallel kinematic machines: Design, analysis and simulation in an integrated virtual environment. *J. Mech. Des.* **2005**, *127*, 580–588. [CrossRef]
27. Zhang, D.; Wang, L.; Esmailzadeh, E. PKM capabilities and applications exploration in a collaborative virtual environment. *Robot. Comput. Integr. Manuf.* **2006**, *22*, 384–395. [CrossRef]
28. Rega, A.; Genua, A.; Vitolo, F.; Patalano, S.; Sanseverino, G.; Penter, L.; Arnold, F.; Ihlenfeldt, S.; Lanzotti, A. Toward a framework for virtual testing of complex machine tools. In Proceedings of the 3rd International Conference on Design Tools and Methods in Industrial Engineering, ADM 2023, Florence, Italy, 6–8 September 2023; Springer: Cham, Switzerland, 2024; pp. 530–536. [CrossRef]
29. Koçak, M.; Can, F.C.; Gezgin, E. Design of a graphical user interface for the structural synthesis of parallel manipulators with single platform. In Proceedings of the 4th International Conference on Interactive Collaborative Robotics (ICR), Istanbul, Turkey, 20–25 August 2019; pp. 182–192. [CrossRef]
30. Hess-Coelho, T.; de Oliveira, É.; Orsino, R.; Malvezzi, F. Modular modeling methodology applied to kinematically redundant parallel mechanisms. *Mech. Mach. Theory* **2024**, *194*, 105567. [CrossRef]
31. Chen, Z.; Kong, X.; Zhao, C.; Huang, Z. Type synthesis of 3-RSR equivalent 2R1T parallel mechanisms based on screw theory. *Mech. Mach. Theory* **2025**, *211*, 106032. [CrossRef]
32. Li, L.; Zhang, H.; Jin, X.; Chen, Q.; Ye, W. Motion/force transmissibility analysis and inverse kinematics optimization of kinematically redundant parallel mechanisms. *Robotica* **2025**, *43*, 3058–3079. [CrossRef]
33. Zarkandi, S. Kinematic analysis and optimal design of a novel 3-PRR spherical parallel manipulator. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *235*, 693–712. [CrossRef]
34. Xu, J.; Gao, M.; Feng, X.; Tu, Z.; Zhang, S.; Tan, J.; Tu, L.; Yao, R. Dexterity distribution design for attitude adjustment of multi-joint robotics based on singularity-free workspace decomposition. *Mech. Based Des. Struct. Mach.* **2024**, *52*, 1764–1784. [CrossRef]
35. Ding, J.; Han, B.; Tang, C.; Li, J. Research on dexterity of a redundant parallel machine applied in multi-facet drill grinding. *Int. J. Adv. Manuf. Technol.* **2025**, *140*, 3809–3831. [CrossRef]
36. *AS ISO/IEC 19775.1-2025*; Information Technology—Computer Graphics, Image Processing and Environmental Data Representation—Extensible 3D (X3D) Architecture and Base. Standards Australia: Sydney, Australia, 2025.

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.