Root Based Optimization Algorithm for Task-Oriented Structural Design of a Multi-Axial Road Test Rig

XIAO WANG, DACHENG CONG, ZHIDONG YANG, AND JUNWEI HAN
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China
Corresponding author: Zhidong Yang (yangzhidong@hit.edu.cn)
This work was supported in part by the National Natural Science Foundation of China under Grant 51205077.

ABSTRACT
A novel root based optimization algorithm (ROA), which mimics the proliferation of plant roots, is proposed to evaluate the performances of a multi-axial road test rig over a prescribed workspace. Plant roots have developed complex behavior patterns to search for nutrients in the soil and even exhibit swarm intelligence. The growing roots can adapt their actions such as elongation, branching, and tropistic movement according to the environment. During the process of foraging for water and nutrients, the adaptive growth behavior of roots can be simulated to tackle the optimization problem. The efficiency of the proposed ROA has been validated by comparing it with well-known intelligent algorithms using classical and IEEE CEC 2019 benchmark functions. Computational results indicate that ROA outperforms other algorithms to search a global optimum on several benchmarks. Furthermore, the results are tested with nonparametric statistical methods, e.g., Wilcoxon rank-sum test. Besides, ROA is applied to the optimal dimensioning synthesis of the multi-axial road test rig (MRTR) based on the orthogonal 6-RSS parallel manipulator to promote the device’s performance. The inverse kinematics of the multi-axial road test rig based on the asymmetrical orthogonal 6-RSS parallel mechanism is analyzed in detail. Applying the achieved dimensionless Jacobian, the performance criteria involving the global isotropy index, the force/velocity transmissibility index, the decoupling index are established. By simultaneously taking account the proposed criteria of performance, structural optimization is carried out resorting to ROA. Finally, a set of optimized dimensional parameters is obtained. The practical application further illustrates that ROA has the potential to deal with complex optimization problems.

INDEX TERMS
Root adaptive growth, optimization algorithm, benchmark test, orthogonal 6-RSS parallel manipulator, task-oriented performance criteria, optimal structural design.

I. INTRODUCTION
Many nature-inspired intelligence algorithms have been developed to address complex and nonlinear problems, which cannot be solved effectively by traditional methods. The existing meta-heuristic algorithms may be divided into two main categories: Evolutionary algorithms (EA) and Swarm intelligence algorithms (SI) [1].

Evolution is a stochastic process and emerges at the population level. A generation of individuals will bring forth offspring by recombination and mutation, and the fittest are selected to survive each generation [1]. Notable evolutionary algorithms include the Genetic Algorithm (GA) [2], Differential Evolution (DE) [3], Invasive Weed Optimization (IWO) [4] and Improved Genetic Immune Algorithm (IGIA) [5], etc.

Swarm intelligence is the collective behavior of the biological population and the local interactions between individuals lead to global behavior. Most species show swarm intelligence and the well-known examples consist of bird flocks, fish schools, herds of quadrupeds, the colony of social insects and bacteria molds [1]. Therefore, inspired by these diverse swarm systems, a plethora of algorithms, such as particle swarm optimization (PSO) [6], PSO with inertial weights (GPSO) [7], Cuckoo Optimization Algorithm (COA) [8], Harry’s Hawk Optimization Algorithm (HHO) [9], Grey Wolf Optimizer (GWO) [10], Whale Optimization...
Algorithm (WOA) [11], Squirrel Search Algorithm (SSA) [12], Bee Colony Optimization (BCO) [13], Fitness Dependent Optimizer (FDO) [14], Moth Flame Optimization Algorithm (MFO) [15], Slime Mould Algorithm (SMA) [16], etc, have been created.

An elaborate literature review on nature-inspired algorithms is presented in [17]. In contrast to animals, plants based algorithms are scarce and only a few studies inspired by plant reproduction and root growth have been carried out. Mehrabian and Lucas [4] introduced the evolution-based Invasive Weed Optimization (IWO) through imitating the behavior of weeds in occupying territory and colonizing. The IWO is further enhanced by combining chaos theory [18]. Plant propagation algorithm (PPA) is based on the way the strawberry plant propagates using runners, which can explore the search space with long runners while short runners enable it to exploit the local space [19]. Similarly, Runner Root Algorithm (RRA) mimics the behavior pattern of strawberry and spider plants which propagate through their runners for global search and produce roots to search for local resources simultaneously [20]. Flower Pollination Algorithm (FPA) is proposed to simulate the features of flower pollination, combining local and global searches through self-pollination and long-distance cross-pollination [21]. Inspired by local and global seed dispersal methods of trees, Forest Optimization Algorithms (FOA) has been proposed [22]. Similarly, Natural Forest Regeneration Algorithm (NFR) searches the global and local space through random seed dispersal and decreasing the radius of seed dispersal [23]. Artificial plant optimization algorithm (APOA) is a methodology mapping the growing process of a tree into optimization, including three phenomena: photosynthesis operator, phototropism operator and apical dominance operator [24]. Also, Cheraghali [25] developed the Trees Growth Algorithm (TGA) inspired by trees competition for acquiring light and foods. According to the self-defense mechanisms of plants, an optimization algorithm is presented based on the predator-prey model [26]. Apart from propagation, plants have evolved complex root systems to forage for water and nutrient. Root Mass Optimization (RMO) has been developed based on root re-growing and root branching [27]. Furthermore, Artificial root foraging optimization (ARFO) is proposed which uses root branching and elongation for local and global search, and random growth of lateral roots keeps the diversity [28].

When a root is located on a fertile circumstance, it will grow faster and produce more collateral roots, otherwise, it will generate fewer branches or even cease growing. According to the No-Free-Lunch (NFL) theorem [29], there is no algorithm for solving all optimization problems. Therefore, it is still a challenge to design better optimizers. Inspired by the root proliferation strategies, a new root optimization algorithm (ROA) is developed for global optimization. Unlike RMO and ARFO, the proposed ROA adopts three new strategies: (1) emission of primary roots, (2) branching, (3) re-growth. In the emission phase, let the main roots elongate to move toward a better food source. According to the fitness, the main roots produce new roots to exploit around the local area. In the re-growing phase, let some of the branch roots randomly move to reach new places. Depending on the fitness value, the new main roots are selected from all populations. The performance and efficiency of the ROA have been tested on some benchmarks. Besides, ROA is applied to the optimal design of road test rig based on asymmetrical orthogonal 6-RSS parallel manipulator with a specific task. The kinematic and static instantaneous performance indices based on the scaled Jacobian matrix have been specifically developed for the service load simulation in this article, containing force isotropy index, force/motion transmission index and decoupling index.

The major contributions of this article are as follows:
1. A novel plant roots-inspired intelligent algorithm (ROA) is proposed. The growth mechanisms of roots are studied thoroughly and modeled mathematically including emission of primary roots, branching and re-growth, which combines swarm intelligence and evolutionary strategies to trade-off in both exploration and exploitation abilities.
2. The proposed algorithm is validated on 29 benchmark functions including the classical as well as modern CEC 2019 benchmarks. The efficiency is also studied by rigorous comparison with existing well-known nature-inspired optimization algorithms through nonparametric statistical analysis.
3. The inverse kinematics of the multi-axial road test rig based on the asymmetrical orthogonal 6-RSS parallel mechanical system is analyzed in detail and the Newton-Raphson algorithm has been proven to be efficient for solving the problems of the inverse kinematics. Based on the four performance indices, the effectiveness of ROA is investigated for the optimal dimensioning synthesis of the multi-axial road test rig.

The remainder of the paper is organized as follows. Section II describes the principle of root optimization algorithm (ROA) and validation and analysis have been performed. Section III analyzes the inverse kinematics of a multi-axial road test rig. Section IV establishes the kinematic and static performance indices. Section V examines the effect of each design parameter on the performance indices and presents the dimensional optimization with the ROA algorithm. Finally, Section VI presents the conclusions of the paper.

II. ROOT OPTIMIZATION ALGORITHM
A. MATHEMATICAL PRINCIPLE OF ROA
Foraging for limited resources, plant roots develop complex behavior patterns and even exhibit swarm intelligence like animals through fast electrical and chemical signals interaction. The roots can sense the environment to adapt their actions, e.g., the growth direction adjustment is based on the mutual distance and roots can be all directed towards areas with optimal water and nutrient gradients in the soil [30]–[32].

X. Wang et al.: ROA for Task-Oriented Structural Design of a MRTR
Several researchers have investigated many models in accordance with experimental data to mimic growing-roots patterns. The roots can be divided into two subsystems: main roots and collateral roots. The key principles are composed of three phases: (1) emission of primary roots, (2) branching, (3) re-growth. The enormous random proliferation of the main roots in areas free of other roots will dominate the initial stage, providing a highly explorative capability. As iterations surge, main roots slow down their growth rate and more collateral roots manage to navigate toward the source of the resource by local fluctuations. Furthermore, current resources can be exhausted rapidly and new research must be undertaken.

Assume the plant produces $M$ main roots, i.e. $X = (X_1, X_2, \cdots, X_M)$, each of which searches in the D-dimensional growth space. The $i-th$ root is described by $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})^T (i = 1, 2, \cdots, M)$, where the $j-th$ decision variable $x_{ij}$ is subject to

$$x_{ij} \in [LB_j, UB_j] \quad j = 1, 2, \cdots, D \quad (1)$$

where the $D$ is the number of design variables, and $LB_j$ and $UB_j$ are the lower and upper bound of growth space for $j-th$ design variable $x_{ij}$, respectively.

The element of the vector is initialized as follows

$$x_{ij} = LB_j + (UB_j - LB_j) \times \text{rand}() \quad (2)$$

where $\text{rand}()$ is a random number distributed uniformly in [0,1].

The roots in each generation are sorted in descending order (or ascending order) according to the fitness as $\hat{X} = (\hat{X}_1, \hat{X}_2, \cdots , \hat{X}_M)$, of which $\hat{X}_1$ and $\hat{X}_M$ are the best position and the worst position, respectively.

1) EMISSION OF PRIMARY ROOTS

At first, the branch roots growth mechanism is deduced as follows:

The grow radius of the $i-th$ root is defined as

$$R_i = \begin{cases} c_1 \ast (UB - LB) & i = 1 \\ 2c_1 \ast |X_i - \bar{X}_M| & i = 2, \cdots, M \end{cases} \quad (3)$$

where $c_1$ is a random vector distributed evenly in $[0,1]$.

When the main roots reside in better conditions, it will produce more collateral roots. The number of branch roots as offspring generation is a linearly decreasing function concerning the fitness value of the main roots. Additionally, the number $b$ of branch roots will diminish dynamically with the shrink of the grow radius using a nonlinear function.

The $k-th$ branch root produced by $i-th$ main root is written as

$$\hat{X}_k = \cos(2\pi c_2), \ast R_i + \bar{X}_i \quad k = 1, 2, \cdots, b \quad (4)$$

where $c_2$ is a random vector distributed evenly in $[-1,1]$, and $b = b_{\text{max}} - i + 1$, where the maximum number $b_{\text{max}}$ of branch roots satisfies

$$b_{\text{max}} = \text{ceil}(b_{\text{max}} \ast m_u) \quad (5)$$

where $m_u$ is a random vector distributed evenly in $[0,1]$. $\text{Iter}$ and $\text{Iter}_{\text{max}}$ are the iteration number and its maximum value, respectively. Inspired by [33], the $m_u$ is defined as the probability of generating branch roots by main roots, as shown in Figure 2.

2) BRANCHING

The grow radius is narrowed over time and the algorithm will converge quickly to the local optimum solution. To explore the growth space sufficiently, roots grow freely similar to the method in [27] as follows

$$\hat{X}_k = \gamma \ast c_3 \bar{X}_i \quad k = 1, 2, \cdots, b \quad (8)$$

where $c_3$ is a random number distributed evenly in $[-1,1]$ and $\gamma$ is defined as

$$\gamma = c_4 / \sqrt{c_4^T c_4} \quad (9)$$

where $c_4$ is a random vector distributed evenly in $[-1,1]$.

3) REGROWTH

After the current resources have been exhausted, new research will start to search for a new area. To enhance
exploration capability, the branches are relocated through the following equation:

\[
\hat{X}_k = c_4 \cdot R_i + \text{Lévy}(\beta) \cdot * \hat{X}_i \quad k = 1, \ldots, b
\]  

(10)

where lévy distribution can provide more efficient exploration. The lévy flight is stated as [12]

\[
\text{Lévy}(\beta) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^\beta}
\]  

(11)

where \(r_a\) and \(r_b\) are random numbers normally distributed in \([0,1]\), \(\beta\) is a constant \((\beta = 0.5)\), and \(\sigma\) is calculated as

\[
\sigma = \left( \frac{\Gamma(1 + \beta) \times \sin(\frac{\pi \beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}} \right)^{1/\beta}
\]  

(12)

where \(\Gamma(x) = (x + 1)\).

After the update of the position of the roots, the new main roots are reselected from the merged set according to the survival of the fittest and the number of new main roots in the next generation remains unchanged. The pseudo-code of the ROA algorithm is illustrated in TABLE 1.

### TABLE 1. Pseudocode of ROA.

**ROA**

// Initialization

Set \(\text{Iter}_{\text{max}}, M, b_{\text{max}}\), and \(m_w\)

// Initialize the main root location according to Eq.(2)

Sort \(X\) according to fitness and obtain sorted \n
\(\bar{X} = (\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_M)\)

Repeat \(\text{Iter} = 1\)

Calculating \(b_{\text{max}}\) according to Eq.(5) and Eq.(6)

// Main roots growth and generating branches root phase

For \(i = 1, 2, \ldots, M\)

\begin{align*}
\hat{X}_i & = c_1 \cdot (U - L) + \bar{X}_i, \\
R & = \left\{ c_i \cdot (U - L) + \bar{X}_i \right\}_{i = 1, \ldots, b, \quad \text{Iter} = \text{Iter} + 1}
\end{align*}

If \(r_\text{rand}(\cdot) > m_r/2\)

\(\hat{X}_i = \cos(2\pi c_i) \cdot \hat{X}_i + \bar{X}_i\)

ElseIf \(\text{rand}(\cdot) > 0.5\)

\(\hat{X}_i = \gamma_i \cdot c_i \bar{X}_i\)

Else

\(\hat{X}_i = c_i \cdot \hat{R} + \text{Lévy}(\beta) \cdot * \hat{X}_i\)

Endif

Check if any root goes beyond the space and amend it

Endfor

Sort merged roots \(\bar{X} \cdot \hat{X}\) according to fitness

and obtain new main roots \(\bar{X} = (\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_M)\)

**TABLE 2. Parameter setting of ROA and comparison algorithm.**

| Algorithms | Parameters |
|------------|------------|
| ROA        | \(b_{\text{max}} = 10, m_r = 0.01\) |
| MFO        | \(b = 1\) |
| GWO        | N/A        |
| WOA        | \(b = 1\) |
| IWO        | \(\sigma_{\text{rand}} = 0.5, \sigma_{\text{rand}} = 0.001, N = 2, S_{\text{max}} = 5\) |
| GPSO       | \(V_{\text{max}} = 0.2 \cdot UB, w_{\text{max}} = 0.9, w_{\text{min}} = 0, c_1 = 2, c_2 = 2\) |
| GA         | \(P_i = 0.9, P_w = 0.1\) |

**B. BENCHMARK TEST**

The ROA algorithm has been implemented in MATLAB computer program and its efficiency was validated with some well-known benchmark functions [34] as shown in TABLE 10. TABLE 10 summarizes the three types of test functions, the number of variables, the range of variation, and the optimal values. ROA was compared with GPSO, GA and the other state-of-the-art swarm-based IWO, GWO, MFO, and WOA algorithms.

The selection reasons for these algorithms are: 1) They are based on different search strategies involving EA and SI, and inspired by diverse species. 2) All of them are well cited and more mature. 3) They have been applied to tackle real-world problems. 4) These algorithm implementations can be obtained publicly. All algorithms parameter settings remain the same as ones in their original papers, as shown in TABLE 2.

For all the algorithms, population size and maximum iteration equal to 30 and 500 have been utilized. For each benchmark function, the ROA algorithm was run 30 times the same as the other algorithms.

1) **CLASSICAL BENCHMARK TEST FUNCTIONS VALIDATION**

Test functions involve three sets: unimodal, multimodal and composite functions, which are used for testing exploitation level, exploration level and practical problem-solving ability of the algorithm. Statistical results (i.e. average cost function and corresponding standard deviation) are presented in TABLE 3.

The unimodal functions TF1–TF7 have only one global optimum and can be used to evaluate the exploitation capability of the algorithm. It can be observed from TABLE 3 that ROA is very competitive with other algorithms. Obviously, ROA outperforms other all algorithms for all functions TF1–TF7 except TF5. Therefore, the proposed algorithm can provide very strong exploitation.

Unlike unimodal functions, multimodal functions TF8–TF13 are composed of an exponential number of local optima concerning the design variables size. Hence, this kind of test functions is more suitable to assess the exploration level. The results reported in TABLE 3 indicate that ROA shows better results for functions TF8–TF11 and provides suboptimal results for TF12-TF13. The results confirm that ROA can explore the growth space extensively and is capable of superior exploration capability.
TABLE 3. Classical benchmark results.

| Function | \( f(x) \) | Results | ROA | MFO | GWO | WOA | IWO | GPSO | GA |
|----------|-----------|---------|-----|-----|-----|-----|-----|-----|-----|
| TF1      | 0         | Mean    | 6.53E+13 | 2.11E+57 | 3.13E+78 | 130.7394 | 9.6E-20 | 5706.04 |
| Std      |           |         | 2.32E+12 | 4.35E+71 | 1.07E+77 | 206.8545 | 1.94E-19 | 2839.38 |
| TF2      | 0         | Mean    | 5.38E+09 | 3.05E+33 | 3.91E+33 | 0.004326 | 7.4E+12 | 48.16571 |
| Std      |           |         | 5.03E+09 | 3.49E+33 | 1.22E+32 | 0.006669 | 1.54E+11 | 6.716219 |
| TF3      | 0         | Mean    | 8.33E+08 | 2.19E+24 | 1.17E+16 | 0.000344 | 2.0E+07 | 14.60075 |
| Std      |           |         | 2.30E+05 | 1.5E+23 | 2.07E+17 | 1.237893 | 0.000025 | 5240.852 |
| TF4      | 0         | Mean    | 3.13E+03 | 4.93E+18 | 6.41E+47 | 18.0046 | 8.83E-05 | 0.83333 |
| Std      |           |         | 5.20E+01 | 1.6E+17 | 1.429519 | 8.297093 | 7.54E-05 | 0.379049 |
| TF5      | 0         | Mean    | 2.60E+01 | 2.238533 | 2.711086 | 27.89422 | 349.0726 | 6121.829 | 66969637 |
| Std      |           |         | 3.28E+01 | 3.825275 | 0.736295 | 0.451827 | 636.568 | 22809.21 | 29924451 |
| TF6      | 0         | Mean    | 3.00E+01 | 6.77E+02 | 7.619599 | 0.463927 | 19146.13 | 0.008864 | 10825.06 |
| Std      |           |         | 5.00E+02 | 2.561453 | 0.406474 | 0.272421 | 4258.699 | 0.009993 | 5080.49 |
| TF7      | 0         | Mean    | 7.05E+05 | 0.085865 | 0.000566 | 0.002129 | 0.002435 | 0.003878 | 3.47216 |
| Std      |           |         | 4.64E+05 | 0.004495 | 0.000422 | 0.000296 | 0.000112 | 0.000158 | 1.768185 |

Composite benchmark functions TF14-TF19 with abundant local optima can test the tradeoff ability to handle complications existing in real-world problems. As shown in TABLE 3, ROA provides better results than other algorithms except for TF18 and still had a relatively comparative result in TF18.

As a summary, the results demonstrate high exploration and exploitation characteristics of the proposed ROA algorithm.

2) CEC-C06 2019 BENCHMARK TEST FUNCTIONS VALIDATION

The CEC 2019 benchmark test functions [35] in TABLE 11 for a single objective optimization problem are also utilized to evaluate ROA. As shown in TABLE 4, ROA outperforms other algorithms except in CEC02, CEC03 and CEC06 benchmarks. Even though GPSO has better results for CEC02 and CEC03 benchmarks, ROA has only a small discrepancy.

3) STATISTICAL TESTS

Three nonparametric statistical tests involving Wilcoxon test, Kruskal Wallis test, and Friedman test [36] are selected to analyze the performance of these algorithms.

TABLE 5 shows the R+-, R−, and p-values of Wilcoxon signed-rank test for all the pairwise comparisons. As is clear, ROA shows a significant improvement versus MFO, GWO, WOA, IWO, GPSO and GA with a level of significance \( \alpha = 0.01 \).

Also, results of the Friedman ranks test and Kruskal Wallis test for 29 benchmarks are presented in TABLE 6. Based on comparison results, ROA ranks first, showing the best performance. Obviously, the ROA results are statistically significant compared with other algorithms.

III. INVERSE KINEMATICS MODELLING OF THE MRTR

The parallel manipulators possessing several advantages such as higher accuracy, larger load capacity over serial robots [37], [38] have been applied widely in various industrial fields. As a typical example in the automotive industry, the Multi-axial Road Test Rig (MRTR) is used to replicate the real road condition in the laboratory and has become an imperative instrument for vehicle durability testing [39].

The increasingly stringent requirements of service load simulation for vehicles put high demands on the performance of these testing devices. Since performances of the parallel mechanism are very sensitive to their dimensioning [40],...
the optimum structure parameter design plays a key role in the performance improvement of the MRTR.

A. STRUCTURE DESCRIPTION

The concrete structure and schematic diagram of the MRTR are shown in Figure 3 and Figure 4, respectively. In essence, the MRTR is a specific 6-DOF parallel manipulator as shown in Figure 3, composed of an end-effector, six coupling links, six hydraulic actuators, and seven bell cranks. Driven by six hydraulic actuators, the end-effector can exert vertical, lateral, and longitudinal forces as well as camber, steer, and brake moments at each spindle of the vehicle, as shown in Figure 4. The light-weighted coupling link and bell crank can lead to a reduction of the inertia of the moving links. Besides, the actuators are allowed to be located close to the base which also reduces the inertia of the moving rods. The MRTR with unique topology possesses superior properties, e.g. the orthogonal configuration of limbs reduces the cross-couplings between sub-chains and complexity of the control system [41].

As shown in Figure 4, all parts connect each other with S (spherical) joint, R (revolute) joint and actuated P (prismatic) joint by a hydraulic actuator. Thus, the end effector is attached to the fixed base through six limbs, where the limb1, limbs 4-6 have the identical structure. Limb 2 and limb 3 are merged and form a 2-DOF hybrid kinematic chain, which can execute vertical or brake input independently. The brake rotation is transmitted from the bell crank to the mobile platform by the parallelogram two-bar-linkage. The brake event occurs occasionally during the vehicle durability testing, so the 2-DOF serial sub-chain plays a significant role. And its principle is illustrated in Figure 7 in detail.

B. COORDINATES DESCRIPTION

To facilitate kinematic analysis, a fixed Cartesian frame \( O_b \) with \( x_b, y_b, z_b \) and a moving frame \( O_p \) with \( x_p, y_p, z_p \) are assigned on the fixed base and at the centroid of the moving platform in Figure 5, respectively. The local reference frames \( a_i \) are attached to the limb \( i \). Define vector \( \mathbf{p}_i \) as the position of the point \( p_i \) in the moving frame \( \{ p \} \) and vectors \( a_i \) and \( c_i \) concerning the frame \( \{ b \} \). The pose of end-effector and \( t = [x, y, z]^T \) as the coordinate of point \( O_p \) relative to the frame \( \{ b \} \). Define \( \mathbf{q} = [x, y, z, \phi, \psi, \theta]^T \) as the pose of end-effector and \( \mathbf{t} = [x, y, z]^T \) as the coordinate of point \( O_p \) relative to the frame \( \{ b \} \). Define \( \mathbf{q} = [x, y, z, \phi, \psi, \theta]^T \) as the pose of end-effector and \( \mathbf{t} = [x, y, z]^T \) as the coordinate of point \( O_p \) relative to the frame \( \{ b \} \). The angular velocity mapping function \( \omega \)
\[ \omega = \begin{bmatrix} c\phi \cdot c\psi & -s\psi & 0 \\ c\phi \cdot s\psi & c\psi & 0 \\ -s\phi & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \phi \\ \psi \end{bmatrix} = U \begin{bmatrix} \psi \\ \phi \end{bmatrix} \] (13)

where \( s(\cdot) = \sin(\cdot) \), \( c(\cdot) = \cos(\cdot) \) and the same in the following.

As shown in Figure 5, in the reference frame \([b]\), the position of the point \(p_i\) on the platform can be described as

\[ p_i^b = t + R p_i^p \] (14)

FIGURE 3. Multi-axis road test rig.

FIGURE 4. Structural geometry of MRTR.

FIGURE 5. Coordinate systems for the MRTR.

FIGURE 6. Schematic representation of a kinematic chain (\(i = 1, 4, 5, 6\)).

where \( R \) is the transformation matrix from the moving frame \([p]\) to the fixed frame \([b]\), following \(Z-Y-X\) convention. The matrix \( R \) can be written as (15), as shown at the bottom of the next page.

C. INVERSE DISPLACEMENT SOLUTION

As shown in Figure 6 and Figure 7, the constrained equation for \(i\)th \((i = 1, \ldots, 6)\) limb can be written as

\[ l_i = p_i^b - a_i - b_{1i} \quad i = 1, 4, 5, 6 \] (16)
\[ l_2 = p_2^b - a_2 - b_{12} - b_{13} \quad i = 2 \] (17)
\[ l_3 = p_3^b - a_3 + b_{12} - b_{13} \quad i = 3 \] (18)

where \( l_i \) represents the vector of the \(i\)th coupling link in the fixed frame \([b]\), \( a_i \) is the position vector of the \(i\)th revolute joint \(a_i\) and \( b_{1i}, b_{2i} \) denote the position vector of the \(i\)th bell crank transmission arm and actuation arm, respectively.

Define unit position vector of the \(i\)th transmission arm and actuation arm as \(s_{1i}\) and \(s_{2i}\) respectively, so the \(b_{1i}\) and \(b_{2i}\) can be rewritten as

\[ b_{1i} = b_{1i} s_{1i} \] (19)
\[ b_{2i} = b_{2i} s_{2i} \] (20)
where \( b_{1i} \) and \( b_{2i} \) represent the length of transmission arm and actuation arm, respectively and the unit vector \( s_{1i} \) and \( s_{2i} \) can be derived as

\[
s_{1i} = \begin{cases} 
[0, c\theta_i, s\theta_i]^T, & (i = 1, 2, 3) \\
[c\beta_i, 0, -s\beta_i]^T, & (i = 4, 5, 6)
\end{cases}
\]

\[
s_{2i} = \begin{cases} 
[0, c(\theta_i + \beta_i), s(\theta_i + \beta_i)]^T, & (i = 1, 2, 3) \\
[c(\theta_i - \beta_i), 0, -s(\theta_i - \beta_i)]^T, & (i = 4, 5, 6)
\end{cases}
\]

where \( \theta_i \) and \( \beta_i \) are the rotation angle and constant angle of the bell crank, respectively.

The Eq. (16)-(18) depicts the holistic kinematics of MRTR and reformulate them in matrix form as follows:

\[
l = p^b - a + Ss
\]

where \( l = [l_1 \cdots l_{60}]^T, p^b = [p^b_1 \cdots p^b_{16}]^T, a = [a_1 \cdots a_{60}]^T \) and \( s = [s_{11} \cdots s_{16}]^T \) are the matrix of position vectors about coupling link, the joints on moving platform, joint \( a_i \) and bell crank transmission arm, respectively. And \( S \) represents the coupling coefficient matrix, which is

\[
S = \begin{bmatrix}
S_1 \\
S_2 \\
S_3 \\
S_4 \\
S_5 \\
S_6
\end{bmatrix} = \begin{bmatrix}
-b_{11} & 0 & 0 & 0 & 0 & 0 \\
0 & -b_{12} & -b_{13} & 0 & 0 & 0 \\
0 & b_{12} & -b_{13} & 0 & 0 & 0 \\
0 & 0 & 0 & -b_{14} & 0 & 0 \\
0 & 0 & 0 & 0 & -b_{15} & 0 \\
0 & 0 & 0 & 0 & 0 & -b_{16}
\end{bmatrix}
\]

(24)

According to the Eq. (23), the function concerning \( \theta_i \) can be derived as:

\[
f_i(\theta) = \|p_i^b + a_i + S_i s\|^2 - l_{i0}^2 = 0
\]

(25)

where \( \theta = [\theta_1 \cdots \theta_6]^T \) is the matrix of bell crank rotation angles.

Define \( s'_{1i} = \frac{ds_{1i}}{d\theta} \) as derivative concerning \( \theta_i \) and \( s' = [s'_{11} \cdots s'_{16}]^T \) is the matrix of the derivative function.

Based on the Eq. (25), the matrix form can be obtained

\[
f(\theta) = l \circ l - l_0 \circ l_0
\]

(26)

where \( \circ \) means the Hadamard product (i.e. the elements multiplication).

By resorting to the Newton-Raphson algorithm, \( \theta_i \) can be solved on-line with the Eq. (27)-(30) and Eq. (28) can be solved by QR decomposition to obtain the \( \Delta \theta_k \).

\[
J_{\theta} = \frac{df(\theta)}{d\theta} = 2l \circ Ss'
\]

(27)

\[
J_{\theta} \Delta \theta_k = l_0 \circ l_0 - l(\theta_k) \circ l(\theta_k) \quad k = 0, 1, \cdots
\]

(28)

\[
\theta_{k+1} = \Delta \theta_k + \theta_k
\]

(29)

\[
\max|\Delta \theta_k| < \varepsilon
\]

(30)

where \( l_0 = [l_{10} \cdots l_{60}]^T \) is the matrix about coupling link length, \( \theta_0 = [\theta_{10} \cdots \theta_{60}]^T \) is the matrix of the initial rotation angle, \( k \) is the iteration number, and \( \varepsilon \) is a small scalar solution tolerance.

Regarding Figure 6, the vector of the hydraulic actuator \( d_i \) \( (i = 1, 2, \cdots, 6) \) can be described as

\[
d_i = a_i + b_{2i} - c_i
\]

(31)

Which leads to

\[
d_i = \|d_i\| = \sqrt{(a_i + b_{2i} - c_i) \cdot (a_i + b_{2i} - c_i)}
\]

(32)

Additionally, the unit vector of the hydraulic actuator can be derived as

\[
d_{ni} = d_i / d_i
\]

(33)
The velocity of the joint point \( p_i \) \((i = 1, 2, \cdots, 6)\) by taking the derivative of Eq. (14) can be obtained as
\[
\dot{p}_i = i + \omega \times R p_i^p = \left[ I - R p_i^p \times I \right] \dot{i} = J_{piq} \dot{q} \quad (34)
\]
where \( \dot{p}_i \) represents the velocity of \( p_i \) and \( I \in \mathbb{R}^{3 \times 3} \) is a unit matrix, and \( J_{piq} \) represents a Jacobian matrix between the platform and joint point velocity, \( J_{piq} = \left[ I - R p_i^p \times I \right] \).

Differentiating Eq. (23) concerning time, the velocity function can be given by
\[
I \circ \dot{l} = I \circ (\dot{\rho}^b + S \dot{s}) = I \circ (\dot{\rho}^b + S s \dot{\theta}) = 0 \quad (35)
\]
where \( \dot{\theta} = \left[ \dot{\theta}_1 \cdots \dot{\theta}_6 \right] = \left[ \omega_{b1} \cdots \omega_{b6} \right] ^T \) represents the scalar angular velocity vector of the bell crank.

Let \( \dot{J}_{obiq} \) denote the Jacobian matrix between scalar angular velocity and platform velocity.

And then, as shown in Figure 8, the angular velocity \( \omega_{bi} \) of the \( i \)th bell crank can be calculated as
\[
\omega_{bi} = \omega_{bi} e J_{obiq} \dot{q} = J_{obiq} \dot{q} \quad (37)
\]
where \( J_{obiq} \) denotes the Jacobian matrix between the \( i \)th bell crank angular velocity and platform velocity, and \( e \) represents the unit vector, which is
\[
e = \begin{cases} e_x (i = 1, 2, 3) \\ e_y (i = 4, 5, 6) \end{cases} \quad (38)
\]
where \( e_x \) and \( e_y \) are the unit vector along with the \( x_b \) axe and the \( y_b \) axe of the fixed frame \( \{b\} \), respectively.

Assume \( v_{di} \) be the \( S \) joint point velocity of the \( i \)th bell crank actuation arm, which can be described as
\[
v_{di} = \omega_{bi} \times b_{2i} = -b_{2i} J_{obiq} \dot{q} = J_{vdiq} \dot{q} \quad (39)
\]
where \( \cdot \) denotes the skew symmetry matrix, \( J_{vdiq} \) denotes the Jacobian matrix between the \( S \) joint point velocity of the \( i \)th bell crank actuation arm and platform velocity.

Let \( v_{dni} \) be the \( i \)th actuator scalar sliding velocity, which can be derived as
\[
v_{dni} = d_{ai} \cdot v_{di} = d_{ai} \cdot J_{vdiq} \dot{q} = J_{vdniq} \dot{q} \quad (40)
\]
\[
v_{dni} = J \dot{q} \quad (41)
\]
where \( v_{dni} = [v_{dni1} \cdots v_{dini6}]^T \) denotes the sliding velocity of the actuator and \( J = [J_{vdiq1} \cdots J_{vdiq6}]^T \) is the kinematic Jacobian matrix mapping Cartesian velocity to joint rate.

**IV. KINOSTATIC PERFORMANCE INDICES FORMALIZATION**

Performance synthesis for a multi-DOF mechanism is an application-dependent task [43] and the desired performance of MRTR consists of good kinematic conditioning, high forces transmissibility, and low motion cross-coupling. Performance evaluation indices are a prerequisite for the dimensional synthesis. The Jacobian matrix contains significant information concerning the kinematic and static characteristics of the mechanism so a variety of indices are derived from it to evaluate the kinostatic performance of a parallel mechanism [40]. Isotropy indicates the directional dexterity degree of a mechanism and is one of the common performance measures [44]. Salisbury and Gosselin used the condition number of Jacobian to define the Local Conditioning Index (LCI) [45] and the global conditioning index (GCI) [46] for mechanism isotropy optimization.

The homogeneous Jacobian can also be accomplished by multiplying scaling matrices [47]. Applying the task space and joint space scaling matrices, Stocco et al. [48], [49] designed the global isotropy index (GII) to obtain a position independent worst-case optimum over a workspace. As an alternative to condition number, Olds [50] proposed p, q-norm indices to quantify local and global worst-case kinematic and force performance. Lou et al. [51] applied the minimum singular value of Jacobian in the prescribed workspace to evaluate different performances. Shao et al. [52] adopted the matrix orthogonal degree of Jacobian to evaluate the motion/force transmission performance. Apart from the Jacobian-based indices, motion/force transmissibility performance indices based on the transmission angle [53] or the screw theory [54] have also been developed and are especially prevalent in the analysis of lower-mobility parallel mechanism. When a parallel manipulator possesses kinematics decoupling characteristics, it can simplify the kinematics analysis, and enable calibration and control easier, especially for the high-frequency vibration application [38]. The parallel road test rig requires minimal joint cross-coupling with good force manipulability [55]. Jin and Yang [56] classified the kinematic decoupling into the strong coupling, complete decoupling characteristics, it can simplify the kinematics analysis, and enable calibration and control easier, especially for the high-frequency vibration application [38]. The parallel road test rig requires minimal joint cross-coupling with good force manipulability [55]. Jin and Yang [56] classified the kinematic decoupling into the strong coupling, complete decoupling and partial decoupling. Jin et al. [57] introduced the concept of group decoupling. Chu and Gao [58] proposed the distribution difference of nonzero elements in Jacobian matrix to measure the kinematic coupling degree of forging manipulator. Shen et al. [59] employed a systematic method to analyze the parallel kinematic structures with a lower coupling degree. Legnani et al. [41] developed some new 6-DoF parallel mechanisms decoupled in one specific configuration by modifying the leg structure. Lee et al. [43], [60] put forward the local kinematic cross-coupling index (LKCI) with the angle of interaction between column vectors of the Jacobian to signify joint dependency. In [42], kinematics and...
statics cross-coupling indices were also explored in detail with the same mathematical principle.

Establishing the kinetostatic indices allows us to quantify the performance of the test rig. The multi-criteria based design can provide sufficient control on the range of the design parameters so it is necessary to establish multiple performance indices [61]. For the multi-axial road test rig, the dexterity, velocity/force transmissibility and decoupling degree are more concerned. Therefore, four corresponding indices are given to measure these performances in the following part.

When a specific desired performance goal exists, the Jacobian matrix can be normalized with scaling factors to remove the inhomogeneity limit [49]. Because the multi-axial road test rig has the unambiguous task requirements as shown in TABLE 7, the task-specific design approach is more suitable and practical. As a consequence, four performance measures are all developed based on the scaled Jacobian.

**A. FORCE MANIPULABILITY MEASURE**

Assuming the gravitational forces, inertia forces, and frictional forces unconsidered, the mapping between the generalized force \( f \) exerted on the end-effector and the joint force \( Jf \) is

\[ F = JTf \] (42)

According to the maximum desired task-space forces and the maximum force capabilities of actuators, the scaling matrices are given by

\[ \Omega_F^{-1} = diag([F_{x_{\text{max}}}F_{y_{\text{max}}}F_{z_{\text{max}}}M_{x_{\text{max}}}M_{y_{\text{max}}}M_{z_{\text{max}}}] \] (43)
\[ \Gamma_f = diag([f_{\text{max},1} \cdots f_{\text{max},6}]) \] (44)

The normalization of force transformation Jacobian \( JF \) can be derived as follows

\[ \Omega_F^{-1} \Delta F = J^T \Gamma_f \Delta f \] (45)
\[ \Delta F = \Omega_F J^T \Gamma_f \Delta f \] (46)
\[ J = \Gamma_f J \Omega_F \] (47)

where \( \Delta F \) and \( \Delta f \) are the percentage vectors of maximum task-space forces and maximum actuation forces, respectively.

The workspace-inclusive global isotropy index is a more stringent measure than the condition number, which is defined as [62]

\[ \eta_h = GII(\xi) = \min_{q_0,q_1 \in W} \frac{\sigma_{\text{min}}(Jf(\xi,q_0))}{\sigma_{\text{max}}(Jf(\xi,q_1))} \] (48)

It can be seen that a larger GII value promises a better control dexterity in general.

For the multi-axial road test rig, exerting larger generalized forces on the spindle of the vehicle is expected for the given actuation forces. Here, the minimum singular value \( \sigma_{\text{min}}(Jf^T) \) is used to characterize the force manipulability of the mechanism, then the force transmission index \( \kappa_f \) can be expressed as

\[ \kappa_f = 1 - e^{-\tau_f} \] (49)
\[ \tau_f = \sigma_{\text{min}}(Jf^T) = \sigma_{\text{min}}(JF) \] (50)

In a real implementation, usually the maximum task-space force is expected to be produced in all points in the prescribed workspace for the test rig. The global force index (GFI) of a defined workspace \( W \) is defined as

\[ \eta_f = \frac{\int W (\kappa_f) dW}{\int W dW} \] (51)

A larger GFI value allows a better force transmission rate.

**B. MOTION MOBILITY MEASURE**

For succinctness, let \( v = v_{dn} \) and the mapping between task domain velocity and joint velocity vector can be reformulated as

\[ \dot{q} = J^{-1}v \] (52)

Similarly, the corresponding scaling matrices for velocities are

\[ \Omega_q^{-1} = diag([\dot{x}_{\text{max}}\dot{y}_{\text{max}}\dot{z}_{\text{max}}\dot{\omega}_x\dot{\omega}_y\dot{\omega}_z\dot{\text{max}}]) \] (53)
\[ \Gamma_v = diag([v_{\text{max},1} \cdots v_{\text{max},6}]) \] (54)

The scaled velocity Jacobian \( Jv \) can be derived as

\[ \Omega_q^{-1} \Delta \dot{q} = J^{-1} \Gamma_v \Delta v \] (55)
\[ \Delta \dot{q} = \Omega_q J^{-1} \Gamma_v \Delta v \] (56)
\[ J_v = \Omega_q J^{-1} \Gamma_v \] (57)

where \( \Delta \dot{q} \) and \( \Delta v \) are the percentage vectors of maximum task-space velocities and maximum actuation velocities, respectively.

Still, based on the minimal singular value, the velocity transmission index \( \kappa_v \) can be mathematically expressed as

\[ \kappa_v = 1 - e^{-\tau_v} \] (58)
\[ \tau_v = \sigma_{\text{min}}(J_v) \] (59)

In the automotive testing industry, the velocity of the multi-axial test rig should reach a desired value in the prescribed workspace and high-speed are always desirable. The global velocity index(GVI) over a defined workspace \( W \) is defined as

\[ \eta_v = \frac{\int W (\kappa_v) dW}{\int W dW} \] (60)

A larger GVI value indicates a better velocity transmission rate.
C. CROSS-COUPLING MEASURE

Kinematics decoupling means that one motion of the end-effector only corresponds to the input of one limb or one group of limbs [56]. It is difficult to design complete decoupling mechanisms for parallel manipulators’ overall workspace, but it is possible to reduce the coupling in one particular configuration or the limited workspace through optimization design.

The cross-coupling between joint motions can be interpreted as the degree of orthogonality between column vectors of the Jacobian matrix [43]. A quantitative measure of kinematic cross-coupling between two sub-chains is defined as [60]

\[ \kappa_{ij} = \sin \alpha_{ij} \]  
(61)

where \( \alpha_{ij} \) denotes the angle between \( i \)th and \( j \)th column vectors of the Jacobian, which is

\[ \alpha_{ij} = \cos^{-1} \left( \frac{\mathbf{J}_{ij}^T \mathbf{J}_{ij}}{\| \mathbf{J}_{ij} \| \| \mathbf{J}_{ij} \|} \right) \]  
(62)

Since the angle of intersection between two vectors is in the range of \((0, \pi)\), it can be deduced that \( 0 \leq \kappa_{ij} \leq 1 \).

At one specific configuration, the more general local kinematic cross-coupling index for analyzing all joints dependency is defined as [42]

\[ \kappa_{cp} = \prod_{i=1}^{m} \kappa_{ij} \]  
(63)

When \( \kappa_{cp} = 1 \), each two column vectors of the Jacobian matrix are mutually orthogonal and all joint motions are completely decoupled.

The global kinematic cross-coupling index (GCPI) in the prescribed workspace \( W \) is given by

\[ \eta_{cp} = \frac{\int_W (\kappa_{cp}) dW}{\int_W dW} \]  
(64)

Since there exists no analytical means of calculating the Jacobian, numerical integration is required to determine the index value [38]. Thus above integral of global performance indices may be approximated by a discrete sum

\[ \eta_v = \sum_{k=1}^{m} \kappa_{vk} / m \]  
(65)

\[ \eta_f = \sum_{k=1}^{m} \kappa_{fk} / m \]  
(66)

\[ \eta_{cp} = \sum_{k=1}^{m} \kappa_{cpk} / m \]  
(67)

where \( m \) is the integration points of workspace.

V. ASSESSMENT OF PERFORMANCE INDICES FOR PRACTICAL APPLICATION

A. TECHNICAL SPECIFICATIONS AND PHYSICAL CONSTRAINTS

System performance and actuator specifications of the MRTR are shown in TABLE 7. Performance and specification values are nominal at mid position.

To calculate the global performance index, the workspace volume should be determined at first. Alleviating the complexity by reducing the optimization space, a prescribed workspace is tailored to \( W = [-0.13m, 0.13m] \times [-0.2m, 0.2m] \times [-0.19m, 0.19m] \) and a linear interval 0.01 is added.

The joints position \( p_j^p \) of the movable platform is determined by the practical manipulating space and the physical size of joints, which is not considered as optimization variables.

The coupling link length \( l_i \), bell crank transmission arm length \( b_{ij} \) and the ratio \( \eta \) of the bell crank transmission arm to actuation arm are selected as the design parameters, of which initial values are shown in TABLE 8. Additional geometric parameters are fixed, where \([b_{12}, b_{13}, b_{22}, b_{23}] = [0.234m, 0.476m, 0.308m, 0.195m]\).

To reduce the fabrication cost and simplify the design procedure, some constraints are put forward as follows

\[
\begin{align*}
    l_2 &= l_3 \\
    l_4 &= l_5 = l_6 \\
    b_{11} &= b_{14} = b_{15} = b_{16} \\
    \eta &= b_{21}/b_{11} = b_{24}/b_{14} = b_{25}/b_{15} = b_{26}/b_{16} \\
    b_{11} \cdot l_i &= 0 \\
    b_{21} \cdot d_j &= 0
\end{align*}
\]  
(68)

B. APPLICATION TO KINEMATIC PERFORMANCE EVALUATION

To make the number of independent variables more manageable, the effect of individual candidate design parameters on the performance indices should be clarified.

An initial value of each design parameter is assigned as shown in TABLE 8 and a large candidate range of variation...
The change rule of global performance indices is shown in Figures 9-11.

In Figure 9, it is clear that increasing the coupling link length leads to an accompanying increase in all global performance indices. When scale out the initial size, change of indices inclines to be stable, indicating that the effect of coupling link length on the performance of mechanism can be ignored after the coupling link size reaching a certain length.
The variables $l_2$ and $l_4$ have more effects on GFI and GVI. And, GCPI changes to a similar extent with the variation of variables $l_1$, $l_2$ and $l_4$. In conclusion, the performance will improve with the increase of coupling links length.

In Figure 10, the larger the size of the bell crank transmission arm is, the better the isotropy, velocity transmission and decoupling become. While the GFI becomes smaller with the increase of variable $b_{11}$, i.e., force transmission performance becomes worse. It also can be seen that with the increase of variable $b_{11}$, four indices tend towards steady which is consistent with the change in length of the coupling link.

When the length of the bell crank transmission arm reaches a
certain extent, there will be no longer a significant influence on the performance of the test rig.

From Figure 11, the ratio $\eta$ of the bell crank actuation to transmission arm has a nonlinear effect on the GII, GVI. The GII increases rapidly to a maximum with the increase of the ratio. The GFI increases almost linearly while the changing trend of GVI is opposite to the ratio. And, the GCPI is insensitive to the ratio change.

Through the above analysis, it can be concluded that the selected geometric parameters have an overall effect on all the performance of the test rig, of which the length of the coupling link and bell crank transmission arm have a significant impact on the coupling of mechanism. As a result, by observing the variation of the established performance indices with an arbitrarily chosen design variable, it can be confirmed that all the four candidate design parameters are critical for the optimal design of the test rig.

C. ROA FOR THE OPTIMIZATION DESIGN OF MRTR

1) FORMULATION OF THE OPTIMAL DESIGN PROBLEM

To avoid the challenge of a multi-objective problem, the cost function combining all performances in a weighted sum is therefore defined as

$$\min \Psi (X) = \lambda^T \eta$$

subject to $W_{\text{min}} \leq W \leq W_{\text{max}}$

$$LB \leq X \leq UB$$

(70)

where $X = [l_1, l_2, l_4, b_{11}, \eta]^T$ is the set of design parameters, $\lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T$ is the weighting vector and $\eta = [\eta_h, \eta_f, \eta_v, \eta_{cpl}]^T$ is the performance indices vector.
### TABLE 10. Benchmark functions.

| Function | Dim | Range          | $f_{\text{min}}$ | Type         |
|----------|-----|----------------|------------------|--------------|
| $T_1(x) = \sum_{i=1}^{n} x_i^2$ | 10  | $[-100,100]$   | 0                |              |
| $T_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$ | 10  | $[-10,10]$     | 0              |              |
| $T_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{n} x_{ij})^2$ | 10  | $[-100,100]$   | 0                |              |
| $T_4(x) = \max_i |x_i|, 1 \leq i \leq n$ | 10  | $[-100,100]$   | 0                | Uni-modal    |
| $T_5(x) = \sum_{i=1}^{n} [100(x_{i1} - x_i^2)^2 + (x_i - 1)^2]$ | 30  | $[-30,30]$     | 0                |              |
| $T_6(x) = \sum_{i=1}^{n} (|x_i + 0.5|)^2$ | 30  | $[-100,100]$   | 0                |              |
| $T_7(x) = \sum_{i=1}^{n} x_i^2 + \text{random}(0,1)$ | 10  | $[-1.28,1.28]$ | 0                |              |
| $T_{18}(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$ | 10  | $[-500,500]$   | $\approx 418.9829$ |              |
| $T_{31}(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | 10  | $[-5.12,5.12]$ | 0                |              |
| $T_{35}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} - \exp(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) + 20 + e)$ | 10  | $[-32,32]$     | 0                |              |
| $T_{39}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 10  | $[-600,600]$   | 0                |              |
| $T_{43}(x) = \frac{1}{\sqrt{n}} \left[10 \sin(\pi x_1) + \sum_{i=2}^{n} (x_i - 1)^2 [1 + 10 \sin^2(\pi x_i)] + (y_n - 1)^2 \right]$ | 10  | $[-50,50]$     | 0                | Multi-modal  |
| $\quad + \sum_{i=1}^{n} a(x_i, 10, 100, 4)$ | 10  | $[-50,50]$     | 0                |              |
| $T_{47}(x) = 0.1 \sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$ | 10  | $[-50,50]$     | 0                |              |
| $\quad + \sum_{i=1}^{n} a(x_i, 5, 100, 4)$ | 10  | $[-50,50]$     | 0                |              |
| $T_{51}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{52}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{53}(x)$ | 10  | $[-5,5]$       | 0                | Composite    |
| $T_{54}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{55}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{56}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{57}(x)$ | 10  | $[-5,5]$       | 0                |              |
| $T_{58}(x)$ | 10  | $[-5,5]$       | 0                |              |

**Notes:**
- $y_i = 1 + \frac{x_i + 1}{4}$ for the piecewise function $a(x_i, b, c, d)$.
- $k(\cdot)$ is a predefined function.
- The range for $T_{51}(x), T_{52}(x), T_{53}(x), T_{54}(x), T_{55}(x), T_{56}(x), T_{57}(x), T_{58}(x)$ is $[-5,5]$. 
- The range for $T_{59}(x)$ is $[0,1]$. 
- The type for $T_{51}(x), T_{52}(x), T_{53}(x), T_{54}(x), T_{55}(x), T_{56}(x), T_{57}(x), T_{58}(x)$ is Composite.
TABLE 10. *(Continued)* Benchmark functions.

| No. | Functions                      | Dim | Range            | $f(x^*)$ |
|-----|--------------------------------|-----|------------------|---------|
| 1   | STORNS CHEBYSHEV POLYNOMIAL FITTING PROBLEM | 9   | [-8192, 8192]    | 1       |
| 2   | INVERSE HILBERT MATRIX PROBLEM         | 16  | [-16384, 16384]  | 1       |
| 3   | LENNARD-JONES MINIMUM ENERGY CLUSTER   | 18  | [-4, 4]          | 1       |
| 4   | RASTRIGIN’S FUNCTION                   | 10  | [-100, 100]      | 1       |
| 5   | GRIEWANGK’S FUNCTION                   | 10  | [-100, 100]      | 1       |
| 6   | WEIERSTRASS FUNCTION                   | 10  | [-100, 100]      | 1       |
| 7   | MODIFIED SCHWEFEL’S FUNCTION           | 10  | [-100, 100]      | 1       |
| 8   | EXPANDED SCHAEFFER’S F6 FUNCTION       | 10  | [-100, 100]      | 1       |
| 9   | HAPPY CAT FUNCTION                     | 10  | [-100, 100]      | 1       |
| 10  | ACKLEY FUNCTION                        | 10  | [-100, 100]      | 1       |

The longer the length of the coupling links, the smaller the coupling between the sub-chains, but the stiffness of the material needs to be considered and a compact structure also should be guaranteed. The limits of workspace are selected as $W_{\text{min}} = [-0.13m, -0.2m, -0.19m]$ and $W_{\text{max}} = [0.13m, 0.2m, 0.19m]$. The same consideration should be put on the choice of the ratio $\eta$ along with the size of the transmission arm. As a result, the limits chosen for the design variables are $LB = [1.5m, 1.0m, 1.3m, 0.5m, 0.5]^{T}$ and $UB = [1.7m, 1.2m, 1.5m, 0.6m, 0.75]^{T}$. The four performance indices mentioned above conflict with each other, as shown in Figures 10-11. Some trade-offs have to be made to select the appropriate structural parameters. In this article, more attention is placed on the decoupling performance index, so the weight vector is selected as $\lambda = [0.1, 0.2, 0.2, 0.2, 0.5]^{T}$.

2) DESIGN RESULTS AND DISCUSSION

The optimization procedure with the ROA algorithm has been carried out in the MATLAB environment, the results of which are presented in the following part.

As can be seen from Figure 12 (a)-(b), the optimal fitness converges quickly and is reached by all particles using about 10 iterations. In Figure 12 (b), the design variables eventually converge to $[100mm, 100mm, 100mm, 50mm, 0.5]^{T}$.

The structural parameters before and after optimization by the ROA algorithm are shown in TABLE 8.

In TABLE 8, it can be seen that the length of the coupling links and the bell crank transmission arm all reach the maximum value, while the ratio $\eta$ reaches the minimum value. To clearly illustrate the change of performance indices before and after optimization, the distribution of the later three indices in the work plane with $Z = 0$ is shown in Figures 13-15.

In Figures 13-15, it can be observed that the variation of the performance indices is almost symmetrical. The comparison of performance indices before and after optimization is shown in TABLE 9.

After optimization, the speed transmission index $\kappa_v$ increases while the force transmission index $\kappa_f$ shows the opposite trend. The improvement of the speed performance can allow the designer to select the servo valve with the lower-rated flow as well as the oil source with smaller supply flow, which can slash the cost enormously. Although the force transmission performance worsens, the potential of high load capacity of hydraulic systems can be fully utilized. The coupling degree of sub-chains is significantly reduced by optimization, which makes the device more conducive to control.

It can be seen from TABLE 9 that the standard deviations all decrease so the performance becomes more uniform after optimization.

VI. CONCLUSION

A novel root optimization algorithm (ROA) is proposed in this article. Special growing characteristics of roots such as branching, tropism motivates the development of the new algorithm. The branching number is determined dynamically and levy flight enhances the exploration and exploitation capability. The effectiveness of ROA has been proven through...
comparison with the MFO, GWO, WOA, IWO, PSO and GA algorithms on classical and CEC 2019 benchmarks. ROA outperforms the mentioned algorithms in several benchmark functions and the statistical results further highlight that ROA is a promising and competitive algorithm. Resorting to optimized test rig. The physical prototype of the multi-axial test rig for fur-
simulation results show that the test rig achieves an ideal
parameters of a multi-axial test rig is obtained successfully. The

APPENDIX
See Table 10 and 11 here.

ACKNOWLEDGMENT
The authors would like to thank the associate editor and
the reviewers for their careful reading and constructive

REFERENCES

[1] K. L. Du and M. N. S. Swamy, “Search and Optimization by Metaheuristics,” Basel, Switzerland: Birkhäuser, 2016.
[2] K. F. Man, K. S. Tang, and S. Kwong, “Genetic algorithms: Concepts and applications [in engineering design],” IEEE Trans. Ind. Electron., vol. 43, no. 5, pp. 519–534, Oct. 1996, doi: 10.1109/41.538609.
[3] R. Storn and K. Price, “Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces,” J. Global Optim., vol. 11, no. 4, pp. 341–359, 1997, doi:10.1023/A:10082821328.
[4] A. R. Mehrabian and C. Lucas, “A novel numerical optimization algo-

[5] F. Benhouzid-Si Tayeb, M. Besedek, M. Benhouzid, H. Cheurfi, and A. Blizak, “Research on permutation flow-shop scheduling problem based on improved genetic immune algorithm with vaccinated offSpring,” Perv-
cedia Comput. Sci., vol. 112, pp. 427–436, Jan. 2017, doi: 10.1016/j.
perv.2017.08.055.
[6] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm

[7] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in

[8] Rajabioun, “Cuckoo optimization algorithm,” Appl. Soft Comput.,
vol. 11, no. 8, pp. 5508–5518, Dec. 2011, doi: 10.1016/j.asoc.2011.
05.008.
[9] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljaraah, M. Mafjarah, and H. Chen, “Harris hawks optimization: Algorithm and applications,” Future Gener. Comput. Syst., vol. 97, pp. 849–872, Aug. 2019, doi:10.1016/j.future.2019.02.028.
[10] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Adv. Eng. Softw., vol. 95, pp. 69–46, Mar. 2014.
[11] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Adv. Eng.

[12] I. Jain, V. Singh, and A. Rani, “A novel nature-inspired algorithm for

[13] D. Teodorović, “Bee colony optimization (BCO),” in Innovations in Swarm Intelligence, C. P. Lim, L. C. Jain, and S. Dehuri, Eds. Berlin, Germany: Springer, 2009, pp. 59–60.
[14] J. M. Abdullah and T. Ahmed, “Fitness dependent optimizer: Inspired by the bee swarming reproductive process,” IEEE Access, vol. 7, pp. 43473–43486, 2019, doi: 10.1109/ACCESS.2019.2907012.
[15] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” Knowl.-Based Syst., vol. 89, pp. 228–249, Nov. 2015.
[16] D. R. Monismith and B. E. Mayfield, “Slimew mold as a model for numeri-
cal optimization,” in Proc. IEEE Swarm Intell. Symp., Sep. 2008, pp. 1–8, doi:10.1109/SIS.2008.4668295.
[17] D. Molina, J. Poyatos, J. D. Ser, S. García, A. Hussain, and F. Herrera, “Comprehensive taxonomies of nature- and bio-inspired optimization: Inspiration versus algorithmic behavior, critical analysis and recommendations.,” 2020, arXiv:2002.08136. [Online]. Available: http://arxiv.org/abs/2002.08136
[18] M. Misagh and M. Yaghobi, “Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID con-
troller,” I. Comput. Des. Eng., vol. 6, no. 3, pp. 284–295, Jul. 2019.
[19] A. Salhi and E. S. Fraga, “Nature-inspired optimization approaches and the new plant propagation algorithm,” in Proc. Int. Conf. Numer. Anal. Optim. (ICEMATH), Yogyakarta, Indonesia, vol. 1, 2011, pp. 1–8.
[20] F. Merrikh-Bayat, “The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature,” Appl. Soft Comput., vol. 33, pp. 292–303, Aug. 2015.
[21] X.-S. Yang, “Flower pollination algorithm for global optimization,” in

[22] M. Ghaemi and M.-R. Feizzi-Derakhshi, “Forest optimization algorithm,” Expert Syst. Appl., vol. 41, no. 15, pp. 6676–6687, Nov. 2014, doi: 10.1016/j.eswa.2014.05.009.
[23] H. Moez, A. Kaveh, and N. Taghizadeh, “Natural forest regeneration algorithm: A new meta-heuristic,” Iranian J. Sci. Technol., Trans. Civil Eng., vol. 40, no. 4, pp. 311–326, Dec. 2016.
[24] X. Cai, S. Fan, and Y. Tan, “Light responsive curve selection for photo-
synthesis operation of APOA,” Int. J. Bio-Inspired Comput., vol. 4, no. 6, pp. 373–379, 2012.
[25] A. Cheraghali, M. Hajiaghaei-Keshcoli, and M. M. Paydar, “Tree growth algorithm (TGA): A novel approach for solving optimization prob-
lems,” Eng. Appl. Artif. Intell., vol. 72, pp. 393–414, Jun. 2018, doi: 10.1016/j.engappai.2018.04.021.
[26] C. Caraveo, F. Valdez, and O. Castillo, “A new bio-inspired optimization algorithm based on the self-defense mechanisms of plants,” in Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization. Cham, Switzerland: Springer, 2015, pp. 211–218.
[27] X. Qi, Y. Zhu, H. Chen, D. Zhang, and B. Niu, “An idea based on plant root growth for numerical optimization,” in Proc. Int. Conf. Intell. Comput., 2013, pp. 571–578.
[28] L. Ma, Y. Zhu, Y. Liu, L. Tian, and H. Chen, “A novel bionic algorithm inspired by plant root foraging behaviors,” Appl. Soft Comput., vol. 37, pp. 95–113, Dec. 2015.
[29] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimiza-
tion,” IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, Apr. 1997.
[30] A. Trewavas, “Green plants as intelligent organisms,” Trends Plant Sci., vol. 10, no. 9, pp. 413–419, Sep. 2005.
[31] R. Newson, “A canonical model for production and distribution of root mass in space and time,” J. Math. Biol., vol. 33, no. 5, pp. 477–488, Apr. 1995.
[32] D. Leitner, S. Klepsch, G. Bodner, and A. Schnepp, “A dynamic root system growth model based on L-systems: Tropisms and coupling to nutrient uptake from soil,” Plant Soil, vol. 332, nos. 1–2, pp. 177–192, Jul. 2010.
[33] Q.-K. Pan, H.-Y. Sang, J.-H. Duan, and L. Gao, “An improved fruit fly optimization algorithm for continuous function optimization problems,” Knowl.-Based Syst., vol. 62, pp. 69–83, May 2014.
[34] A. Cheraghali, M. Hajiaghaei-Keshcoli, and M. M. Paydar, “Tree growth algorithm (TGA): A novel meta-heuristic algorithm inspired by runners and roots of plants in nature,” Plant Soil, vol. 332, nos. 1–2, pp. 177–192, Jul. 2010.
[35] X.-S. Yang, “Test problems in optimization,” 2010, arXiv:1008.0549. [Online]. Available: http://arxiv.org/abs/1008.0549
[36] K. V. Price, N. H. Awad, M. Z. Ali, and P. N. Saganathan, “The 100-digit challenge: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization,” Nanyang Technol. Univ., Singapore, Tech. Rep., Nov. 2018.
[37] J. C. Chen, S. Garcia, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” Swarm Evol. Comput., vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: 10.1016/j.swevo.2011.02.002.
[38] B. Dasgupta and T. S. Mruthyunjaya, “The Stewart platform manipulator: A review,” Mechanism Mach. Theory, vol. 35, no. 1, pp. 15–40, Jan. 2000.
[39] J.-P. Merlet, Parallel Robots, vol. 128. Dordrecht, The Netherlands: Springer, 2006.
