Enhanced Singularity Avoidance Control for Hydraulic Manipulators Using Damped Least-Squares Optimization

Yang Li, YuKun Zheng*, Song Gao, Rui Song*, and Yibin Li

***Abstract*—This study introduces an enhanced Damped Least Squares (DLS) algorithm featuring adaptive damping coefficient selection to overcome motion control limitations in robotic manipulators operating near singular configurations. Conventional DLS methods face persistent difficulties in maintaining an optimal balance between trajectory tracking precision and velocity continuity at singular regions, often resulting in performance degradation during critical operational phases. Our innovative solution incorporates two key advancements: 1) real-time adjustment of damping parameters through joint velocity differential analysis, and 2) systematic integration of a nonlinear Particle Swarm Optimization (NPSO) technique for comprehensive performance enhancement. Comparative simulations and physical experiments validate the proposed method's superior capability in both accuracy and robustness. This developed control framework introduces a novel methodology for maintaining manipulator stability during operation within singular configuration zones.**

***Index Terms*—Singularity configuration, Extended damped least squares, Nonlinear particle swarm optimization, Hydraulic manipulator**

I. Introduction

As a fundamental component within the domain of intelligent manufacturing, manipulators have consistently intrigued researchers globally. Conventional articulated manipulators frequently encounter distinctive configuration issues arising from their mechanical design and kinematic structure. These challenges materialize as singular configurations, where specific joint configurations lead to restricted or unpredictable motion behaviors [1]. Singularities can arise due to various factors, such as joint limitations, restrictions in the workspace, or complex task demands [2]. Navigating effectively these unique configurations presents an obstacle for manipulator control systems. Conventional methods often struggle to manage motion near singularities, leading to degraded performance, tracking errors, or even potential system instability [3]. Consequently, scholars and practitioners are constantly exploring innovative strategies to address these obstacles and improve the resilience and adaptability of articulated manipulators.

The hydraulic manipulator employed in this study is a serial-type mechanism comprising hydraulic cylinder and hydraulic swing cylinder. Owing to its mechanical structure and operating conditions, this type of manipulator is particularly prone to encountering structural singularities Additionally, sudden changes in joint velocity can lead to fluctuations in hydraulic pressure. The high power and torque output of hydraulic manipulators can pose significant safety risks when velocities become uncontrollable. Therefore, addressing the singularity issues of hydraulic manipulators is crucial to ensuree and efficient operation. In Fig. 1, the singularity problem is evident in the fact that within the singular region adjacent to the manipulator, even a slight movement of the end effector can result in infinite joint angular velocity [5]. The singularity configuration occurs when the 4-th joint and the 6-th joint are collinear. A common strategy for mitigating such singularities involves avoiding these configurations altogether, typically through careful path planning and deliberate structural design.

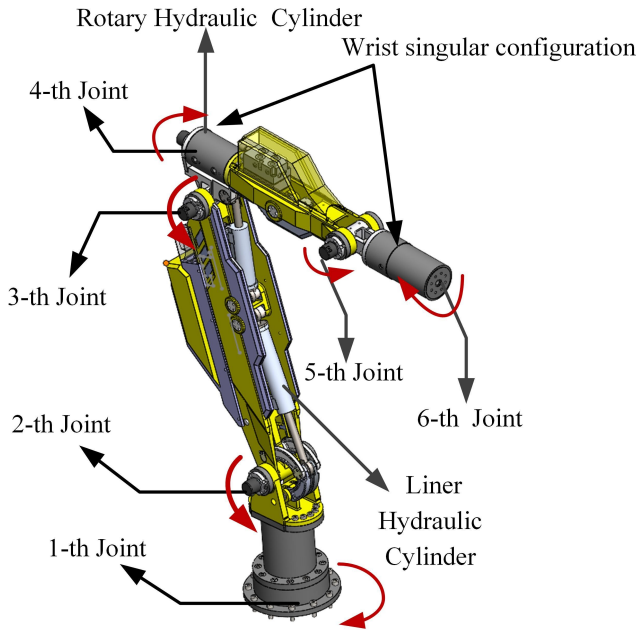


Fig. 1. Singular configuration of manipulator

The field of robotics is intricately linked to the concepts of

Manuscript received November 18, 2024; revised July 4, 2025.

This work was funded by the National Key Research and Development Program of China (2022YFC2604004), Shandong Province Natural Science Foundation (ZR2024QE438), Shandong Province Science and Technology Small and Medium Enterprises Innovation Capacity Improvement Project (2022TSGC2175).

Y. Li is an automation engineer in the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: ly1032948066@163.com).

Y. Zheng is a postdoctoral researcher in the School of Control Science and Engineering, Shandong University, Jinan 250061, China (corresponding author to provide phone: 18563920008; e-mail:zhengyk@sdu.edu.cn).

S. Gao is an associate professor in the School of Information Science and Electrical Engineering, Shandong Jiaotong University, Jinan 250357, China (e-mail: gaosong@sdjtu.edu.cn).

R. Song is a professor in the School of Control Science and Engineering, Shandong University, Ji'nan 250061, China (corresponding author to provide phone: 13505319131; e-mail: rsong@sdu.edu.cn).

Y. Li is a professor in the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: liyb@sdu.edu.cn).

robot Inverse Kinematics (IK) and singularities. Singularities are specific points in a robot's IK where it undergoes a reduction in degrees of freedom. This reduction leads to situations where the kinematic solution becomes ambiguous or impractical, thereby complicating the robot's motion planning and control processes. In the pursuit of enhancing performance, researchers and scholars have extensively explored and suggested various optimization strategies to address the singularity issue in manipulators. Nakamura et al. [5] introduced a manipulability index based on Jacobian matrix to quantify the proximity to singular configurations of manipulators. However, this index has limitations in accuracy when dealing with complex systems. Wampler et al. [6] proposed damped least square (DLS) method by combining Jacobian matrix pseudo-inverse with the least square method, which compromises the accuracy of inverse solutions near singular configurations to enhance the stability of manipulator motion. Wampler method employs a constant damping coefficient, resulting in redundant damping in the nonsingular region. Maciejewski et al. [7] incorporated the minimum singular value into DLS method to ensure stable solutions. Le Minh Phuoc et al. [8] determined the damping coefficient based on a Gaussian DLS (G-DLS), and simulations illustrated the achievement of continuous joint velocity in the vicinity of the singular region. Samuel et al. [9] proposed a Selective DLS (SDLS) approach based on the relationship between the end-effector position and the target position, considering each joint's effect on the relative end position individually. The computation of singular vectors requires high controller computational performance. In [10], an extended DLS combined with compensation was used to prevent dynamic singularities in space robotics, which can gradually reduce or eliminate errors. The selection of parameters impacts the algorithm's effectiveness. Inappropriate parameter values, either too large or too small, can result in poor results, rendering parameter selection a challenging task.

Advanced algorithms for optimizing parameters have been extensively studied, with techniques such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) being prominent in the research. PSO, in particular, has experienced significant developments due to its efficient global search capabilities and parallel processing functionalities. Its applications span across various domains, including object optimization and parameter identification. To address challenges such as local optimization and convergence time, several enhanced PSO algorithms and GA have been introduced. In [10], a chaotic particle swarm was utilized to optimize the trajectory of a space robot, addressing the premature convergence issue of PSO. In [11], the combination of nonlinear PSO (NPSO) with kernel-based extreme learning machine (KELM) model enhanced location optimization tasks. Furthermore, Alhussein M et al. [12] incorporated particle velocity clamping and penalizing principles to boost the performance of the standard PSO. Similarly, Phuoc et al. [8] employed a GA to optimize Gaussian distribution parameters, validated through simulation experiments. Kalra et al. [13] proposed a single-level real-coded GA for multimodal manipulators, evaluating positional error and total joint displacement in the fitness function. Tabandeh et al. [14] introduced an adaptive niches and clustering-based GA to minimize joint angles and reduce end-effector positioning errors. Deb et al. [15] utilized a niche strategy and Taguchi-GA (TGA) [16] to optimize predefined functions within the artificial immune system and enhance joint angle computational results. Furthermore, improvements in precision and computational efficiency are attained by integrating techniques such as neural network ensemble models [17] extreme learning machines [18], and deep neural networks [19]. These methodologies detect residual errors in industrial robots [21], predict increment coefficients for DLS [20], and determine optimal neural network configurations for multi-joint robots [22]. In conclusion, advanced optimization algorithms, coupled with innovative approaches such as neural networks, play a vital role in enhancing the performance and stability of manipulators across diverse applications. However, one of the assumptions in these adaptive strategies is that the parameter damping coefficient remains constant with the joint velocity, potentially rendering the damping redundant and affecting tracking performance deviation.

To address the aforementioned challenges, this work proposes an extended DLS method featuring adaptive damping coefficient selection for motion control of manipulators operating under singular configurations. The key contributions of this study are outlined as follows:

1) The Denavit-Hartenberg (D-H) model is employed to establish the serial hydraulic manipulator's kinematic model, incorporating systematic identification of configuration-dependent singularities through Jacobian matrix singularity analysis.
2) A novel damping coefficient adaptation mechanism is developed, dynamically integrating joint velocity differentials with nonlinear particle swarm optimization (NPSO) to mitigate discontinuous velocity transitions near singular configurations.
3) Comprehensive validation through numerical simulations and physical experiments confirms the proposed method's efficacy in suppressing velocity discontinuities while maintaining enhanced trajectory tracking accuracy compared to conventional DLS approaches.

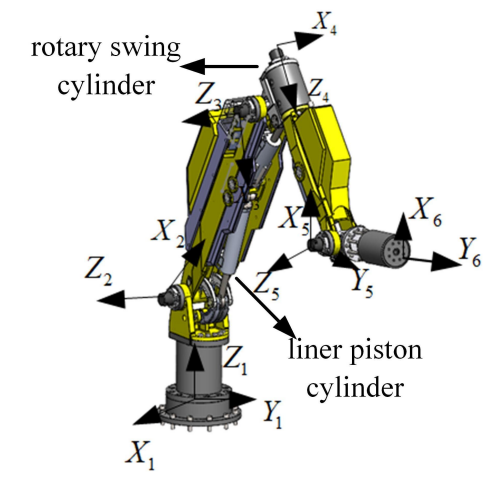
The paper is structured as follows: Section 2 introduces D-H model of the hydraulic manipulator and analyzes typical singular configurations. Section 3 outlines a method for calculating the damping coefficient using the velocity change rate (V-DLS). V-DLS algorithm parameters proposed in Section 4 are optimized using NPSO algorithm. Section 5 conducts simulations and experiments to validate the effectiveness of the proposed algorithm for velocity mutation suppression and reasonable damping allocation, Finally, Section 6 presents the conclusions.

II. Kinematic modeling and singularity analysis

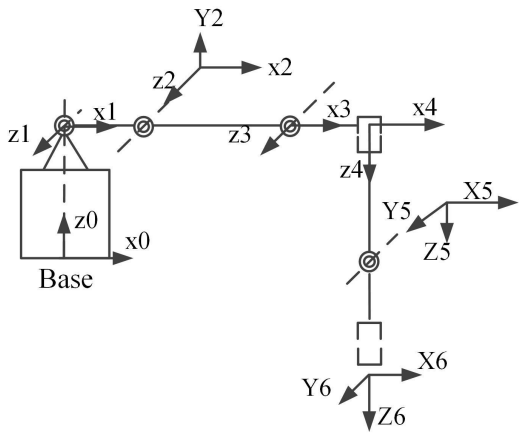
A. Forward kinematics analysis

The research focuses on a six-degree-of-freedom (DOF) hydraulic manipulator, excluding external environmental factors such as friction, gravity, and disturbances, as depicted in Fig. 2. The manipulator comprises a base platform, a linkage mechanism, and six hydraulic actuators. The hydraulic actuators consist of linear piston cylinders and rotary swing cylinders, as illustrated in Fig. 2 (a) .

This section focuses on establishing forward kinematics using the Standard D-H method. In accordance with the right-hand rule, a coordinate system is defined for the connecting



(a) 3D model of hydraulic manipulator



(b) Link coordinate system of hydraulic manipulator

Fig. 2. Mechanical structure and coordinate system of hydraulic manipulator

rod by taking the connection point of each joint as the circle's center (Fig. 2 (b)). Z-axis is assigned as the direction of the rotation axis of the connecting rod, while X-axis is oriented to the right (indicating the forward movement of the robotic arm), and Y-axis is determined following the right-hand rule. D-H parameters resulting from this model are summarized in Table I.

TABLE I
D-H MODEL PARAMETERS

Link	Joint	Link off-set	Link length	Torsion angle	Range
1	φ_1	291.7	0	90°	$[-90°, 90°]$
2	φ_2	0	780	0°	$[-165°, 0°]$
3	φ_3	0	120	90°	$[-40°, 70°]$
4	φ_4	505.7	0	−90°	$[-90°, 90°]$
5	φ_5	0	0	90°	$[-70°, 70°]$
6	φ_6	308.7	0	0°	$[-180°, 180°]$

B. Singular configuration analysis

When addressing the operation of robotic manipulators, particularly in the proximity of a singular configuration, inherent challenges arise due to the mathematical characteristics of the inverse solution. Specifically, in situations where the manipulator is in close proximity to a singularity, even a minor displacement of the end-effector can lead to exceedingly high joint angular velocities. Assuming the reversibility of J, the expression for the joint velocity is derived using (1):

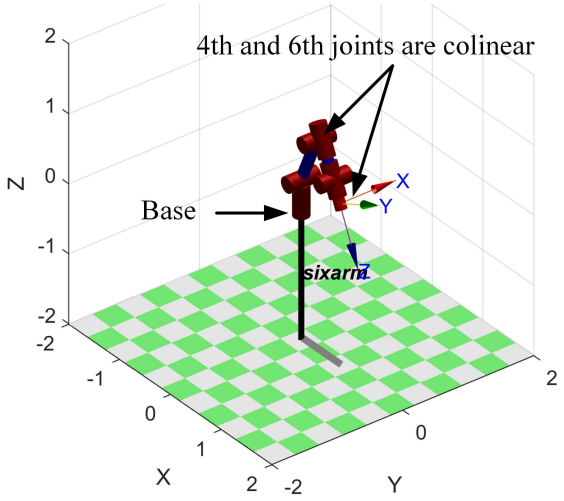
$$\dot{q} = J^{-1}V = J^{-1}\dot{x} \tag{1}$$

This occurrence is predominantly attributed to the attributes of J and its inverse. The difficulties presented by singular configurations in robotic manipulators can be effectively examined and comprehended through the application of singular value decomposition (SVD) to J. This mathematical technique offers a comprehensive insight into the behavior of J in the vicinity of singularities. J can be decomposed utilizing SVD as follows [28]:

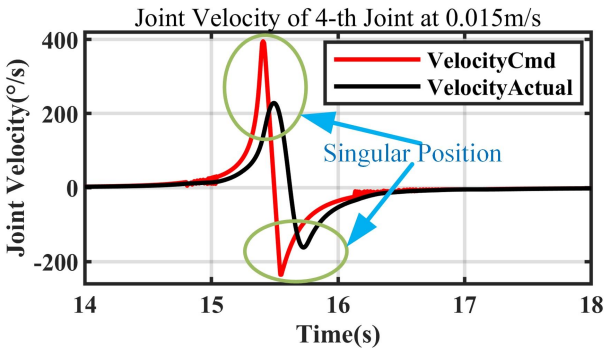
$$\dot{q} = \sum_{i-1}^{m} \sigma_i^{-1} \nu_i u_i^T \tag{2}$$

$$J^{-1} = US^{-1}V^T = \sum_{i-1}^{m} \sigma_i^{-1} \nu_i u_i^T \tag{3}$$

When the singular values of J tend towards zero, the joint velocities tend towards infinity. This scenario signifies that the manipulator is in proximity to a singular configuration. In such instances, J becomes nearly non-invertible, suggesting potential control and stability challenges for the robotic manipulator. A wrist singular configuration arises when specific joints of the manipulator align in a manner that results in J losing rank. In particular, when the 4-th joint aligns collinearly with the 6-th joint, the respective columns in J become linearly dependent (Fig. 3(a)).



(a) Wrist singular configuration



(b) 4-th Joint velocity mutation at 0.015m/s

Fig. 3. Singular configuration of joint

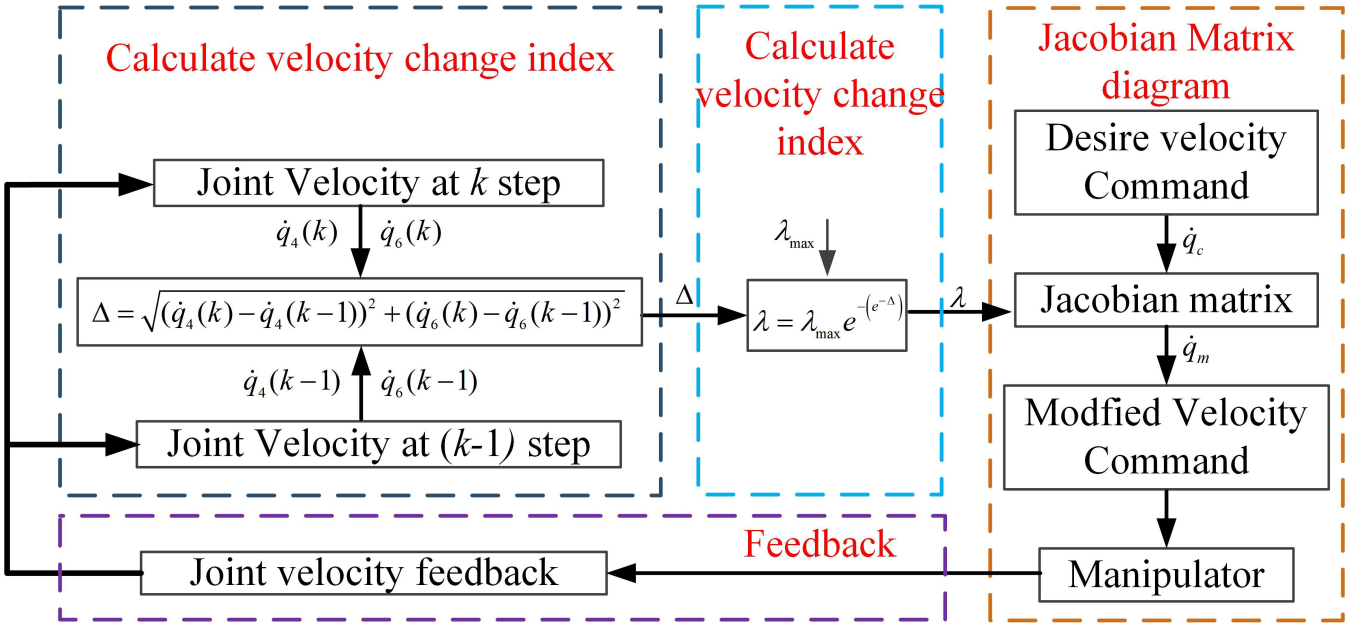


Fig. 4. Flow diagram of V-DLS

The joint velocity curves without processing the singular configuration are depicted in Fig. 3(b) reveals that the 4-th joint velocity undergoes a sudden change, or mutation, in the proximity of the singular configuration, reaching approximately $200°/s$. This abrupt alteration has the potential to significantly disrupt the hydraulic system's performance, leading to significant mechanical stress, instability, and possible damage to system components. The spike in velocity highlights the critical need for the implementation of strategies to handle and prevent singular configurations to uphold the optimal performance and reliability of the hydraulic manipulator.

III. DLS Based on Joint Velocity Change Rate

To mitigate premature convergence, various studies have been undertaken to address this issue. DLS is commonly utilized to handle singular configuration problems in industrial robotics. The typical approach includes incorporating a damping parameter into SVD of J to create a pseudo-inverse, which helps in limiting sudden fluctuations in joint angular velocity while compromising tracking accuracy [29].

The pseudo-inverse of a matrix J^P is a generalization of the matrix inverse for non-square matrices or matrices that do not have full rank. For J, its pseudo-inverse can be mathematically defined as follows [30]:

$$J^p = (JJ^T)^{-1}J^T \tag{4}$$

Remark 1. *The form specified in Eq. (4) is recognized as Moore-Penrose pseudo-inverse of J. Moore-Penrose pseudo-inverse is applicable when J possesses full column rank. In cases where J does not have full column rank (e.g., due to redundant or linearly dependent columns), it is necessary to preprocess J appropriately or resort to alternative generalized inverse techniques.*

To address the aforementioned issues, an extended pseudo-inverse is suggested, commonly referred to as the pseudo-inverse with a damping factor or Tikhonov regularized pseudo-inverse. This approach aims to mitigate numerical instability and enhance the system's robustness. The pseudo-inverse with a damping factor can be represented as follows [31],

$$J^+ = (JJ^T + \lambda^2 I)^{-1}J^T \tag{5}$$

where J represents the original Jacobian matrix; λ represents the damping factor or regularization parameter; I represents an identity matrix of the same size as J^TJ; and J^+ represents the pseudo-inverse in form of DLS.

Remark 2. *When the orientation of the joint angular velocity z aligns with Z-axis, employing the pseudo-inverse J^+ with a damping factor can address issues arising from singular configurations. This approach enhances numerical stability, ensuring effective control over joint angular velocity and force transmission even in singular configurations. By utilizing J and the pseudo-inverse with a damping factor, the robot's motion and mechanical capabilities can be more effectively regulated and controlled in singular configurations.*

By substituting Eq. (5) into Eq. (1), the expression for the joint velocity is obtained as follows:

$$\dot{q} = J^T(JJ^T + \lambda^2 I)^{-1}J^T\dot{x} \tag{6}$$

The conventional DLS technique utilizes a fixed damping coefficient, which may lead to superfluous errors in the trajectory tracking of the manipulator. Recognizing that a different feature of a singular configuration is the sudden alteration in velocity, this study introduces an enhanced DLS approach that determines a suitable damping coefficient λ by considering the rate of change in joint velocity. Based on the aforementioned analysis, it is identified that the 4-th and 6-th joints predominantly contribute to the velocity variation at the wrist singularity. Therefore, the coefficient λ can be calculated by examining the velocities of these two joints. As the manipulator nears singularity, an increased damping coefficient is essential due to the velocity variation, which is represented as an exponential function. Consequently, λ is

adjusted to adhere to this exponential progression.

$$\Delta = \sqrt{(\dot{q}_4(k) - \dot{q}_4(k-1))^2 + ((\dot{q}_6(k) - \dot{q}_6(k-1))^2} \quad (7)$$

$$\lambda = \lambda_{max} e^{-(e^{-\Delta})} \quad (8)$$

where Δ represents the index of joint velocity change rate; $\dot{q}_4$ and $\dot{q}_6$ represent the joint velocities; k represents the k step of the motion process. λ_{max} represents the upper limitation of damping coefficient which determines the range of λ.

Remark 3. *If λ_{max} is excessively large, it will lead to a disproportionately high average value of λ, resulting in significant tracking errors. Conversely, if λ is too small, it will diminish the damping effect of the algorithm. Therefore, λ_{max} is pivotal in determining the control performance of the manipulator.*

Fig. 4 illustrates the flow diagram of V-DLS, delineated into four main components. Initially, the sensor records the joint velocities at steps k and $k-1$. Subsequently, the algorithm computes the velocity change index Δ and adjusts λ. λ_{max} is established through iterative testing. Following this, the revised λ is integrated into J. Ultimately, the adjusted velocity command is obtained.

Remark 4. *The λ_{max} is determined through a trial-and-error process, which may lead to challenges such as prolonged tuning time and potential instability. Therefore, the efficient and prompt selection of parameters poses a significant challenge.*

IV. Nonlinear Particle Swarm Optimization

As previously mentioned, the issue of parameter identification can be viewed as an optimization problem. The forthcoming discussion will employ an optimized approach to expedite the selection of parameters.

A. Particle Swarm Optimization (PSO)

PSO is used to explore the optimal solution by mimicking the movement and flocking behavior observed in birds [32]. The algorithm initiates by randomly distributing a flock of particles across the search space [33], where each particle represents a potential solution. These particles traverse the space with a defined velocity, aiming to identify the global optimum position through iterative processes. During each iteration, a particle adjusts its velocity vector by considering its momentum, the effect of its personal best position, and the overall best position found by any particle within the flock [34]. Subsequently, the particle relocates to a newly computed position based on these adjustments.

Assuming the search space is n-dimensional. The position and velocity of particle i are represented by l_i and v_i, respectively. Starting from a random solution, the fitness function is utilized to update both the best individual solution ($pbest$) and the best group solution ($gbest$), along with the velocities and positions of the particles in each iteration. At each step, the velocity and subsequent position of a particle are determined as follows:

$$v_i^{t-1} = w v_i^t + c_1 r_1 (pbest_i^t - l_i^t) + c_2 r_2 (gbest_i^t - l_i^t) \quad (9)$$

$$l_i^{t+1} = l_i^t + v_i^t \quad (10)$$

where $i \in N$ represents the number of particles; $\omega \geq 0$ represents the inertia weight that reflects the effect of historical particle velocity on current particle velocity; c_1 and c_2 represent the positive constant parameters representing cognitive learning factor and social learning factor, which determine the deviation between individual optimal solution and global optimal solution; r_1 and r_2 represent random variables in a range of $[0, 1]$ for the randomicity of searching; $v_i^t \in [v_{\min}, v_{\max}]$ and $\ell_i^t \in [\ell_{\min}, \ell_{\max}]$ represent the velocity and position of the i-th particle in the i-th iteration, respectively. The fundamental computational procedure consists of the following steps:

Step 1: An appropriate fitness function is chosen to evaluate the effectiveness of potential solutions.

Step 2: The initial search range, along with the initial velocity and position for each particle, is defined.

Step 3: The velocities and positions of particles are calculated in each iteration based on the previously defined parameters.

Step 4: The optimal solutions for individual particles ($pbest$) and the global best solution ($gbest$) are updated using the predefined fitness function, and particle velocities and positions are adjusted accordingly.

In PSO, inertia weight (ω) and acceleration coefficients (c_1 and c_2) play crucial roles in affecting the exploration and exploitation abilities of particles within the solution space. The inertia weight governs the velocity update of the particle, where a higher weight value can boost the global search capability, and a lower weight value can refine the search process. On the other hand, c_1 and c_2 determine how the particle accelerates towards its historical best position and the group's best position, respectively. Careful selection of these parameters is essential to strike a balance between global exploration and local exploitation, thereby enhancing the overall optimization performance.

Remark 5. *During the optimization process, the rationality of parameter selection impacts the convergence speed and accuracy of the algorithm. An excessively large inertia weight can lead to oscillation of particles in the solution space, hindering convergence, whereas an excessively small weight may prematurely trap particles in local optimal solutions. Likewise, an inappropriate setting of acceleration coefficients can alter the search trajectory of particles, diminishing the algorithm's overall performance. Therefore, the careful selection of optimization parameters plays a pivotal role in guaranteeing the effective operation of PSO algorithm.*

B. Parameter design

As previously mentioned, a significant limitation of PSO is its susceptibility to getting stuck in local minima when confronted with complex or multimodal functions

To mitigate premature convergence, numerous studies have been undertaken to devise strategies for circumventing this challenge. The rationale behind the proposed approach is to surmount this limitation. This paper introduces NPSO method. A distinguishing feature of NPSO compared to PSO is that the predefined parameters of NPSO vary nonlinearly with the iteration count. The schematic representation of NPSO is depicted in Fig. 5.

1) Inertia weight: The inertia weight is a parameter that signifies the effect of past particle velocity on the present particle velocity . When set to a high value, the system exhibits improved global search ability, albeit at the cost of extended convergence time. Conversely, a low inertia weight accelerates convergence but may lead to local optimization. Considering these factors, the inertia weight can be determined as follows:

$$\omega = (1 - \omega_{max}) + \frac{\omega_{max} - \omega_{min}}{exp(2 * k/G)^4} \tag{11}$$

where k represents the current iteration step and G represents the maximum iteration step.

Remark 6. *In the initial stage, ω is set to a higher value to improve the global search capability and prevent the algorithm from getting trapped in local optima. As the number of iterations progresses, ω is gradually decreased to expedite the convergence process.*

2) Learning factor: The learning factor plays a significant role in affecting the global search capability of the algorithm. To guarantee that the algorithm demonstrates a robust global search ability in its early stages and efficiently converges towards the globally optimal value later, a dynamic adjustment strategy has been devised as follows:

$$c_1 = c_{1u} - \frac{c_{1u} - c_{1d}}{exp(k/G)} \tag{12}$$

$$c_2 = c_{2d} + \frac{c_{2u} - c_{2d}}{exp(k/G)} \tag{13}$$

In the initial phase, when $c_1 > c_2$, it indicates that the effect of an individual particle's optimal value exceeds that of the group's optimal value, thus promoting an extensive search. Conversely, in the subsequent phase, when $c_2 > c_1$, it signifies a stronger focus on the group's optimal value, allowing for localized exploration and the attainment of a globally optimal solution with increased precision.

3) Fitness function: The fitness function aimed at minimizing both velocity mutation and velocity error plays a crucial role as an indicator of the proximity of a particular solution to the optimal one. To attain the desired optimization result, the fitness function is formulated by taking into account both the joint error and the velocity mutation.

$$F = \log_{10}(\int \delta/1e^{-3} + 1) + \log_{10}(\int |e|/1e^{-3} + 1) \tag{14}$$

where $\int \delta$ represents the absolute integral of velocity mutation in entire motion period; and $\int |e|$ represents the absolute integral of joint error.

Remark 7. *The development of a fitness function aimed at minimizing both velocity mutation and velocity error plays a critical role in optimizing the movement of a robotic arm. By incorporating joint error and velocity mutation throughout the entire motion sequence, the fitness function guarantees that the optimization procedure focuses on achieving seamless and precise movements. Appropriately assigning weights to these elements provides the flexibility to emphasize various aspects of the motion, thereby enhancing the efficiency and dependability of the robotic system.*

Remark 8. *The advantage of utilizing the logarithmic function as an optimization algorithm stems from its stable, monotonic increasing nature and convexity. These characteristics enable the algorithm to circumvent local optimal solutions and enhance the convergence speed of the algorithm [25]. The determination of the base of the logarithmic function is typically achieved through a trial-and-error method.*

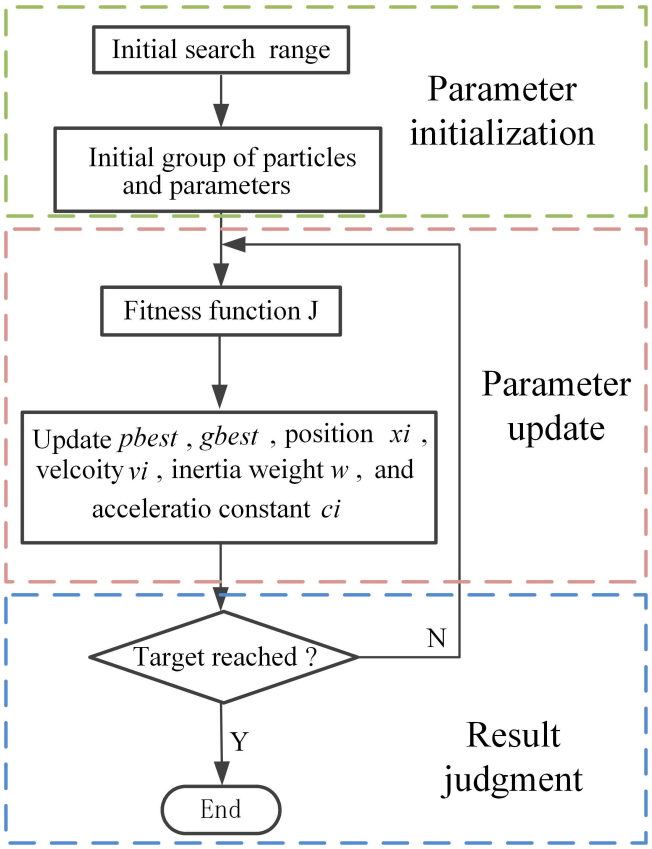


Fig. 5. Flowchart of NPSO Algorithm

Based on the analysis presented in Sections 3 and 4, the algorithm flow chart derived from the integration of V-DLS and NPSO as proposed in this study is illustrated in Fig. 6

The initial process of the NPSO is carried out according to the description provided in Section 4.2. The optimized λ_{max} will be determined upon completion of the iterations. As the manipulator is in motion, the controller receives the desired velocity command. The monitored joint velocities are utilized to compute the joint velocity change index using Eq. (7). Subsequently, the updated λ is determined using Eq. (8) and then inserted into J. Finally, the modified λ is obtained.

V. Simulation and Experimental Results

In this section, both simulations and experiments are carried out to validate the efficacy of the proposed algorithm. The simulation is executed utilizing MATLAB 2023 Robotic Tools on a Windows 11, 64-bit computing system. Diverse scenarios are examined to assess the algorithm's performance across varying conditions. Essential parameters are modified to analyze their effect on the results. The experimental setup replicates the conditions utilized in the simulation to maintain consistency. Data is gathered and examined to contrast the results obtained from both methodologies. The results illustrate the algorithm's capacity to accomplish the

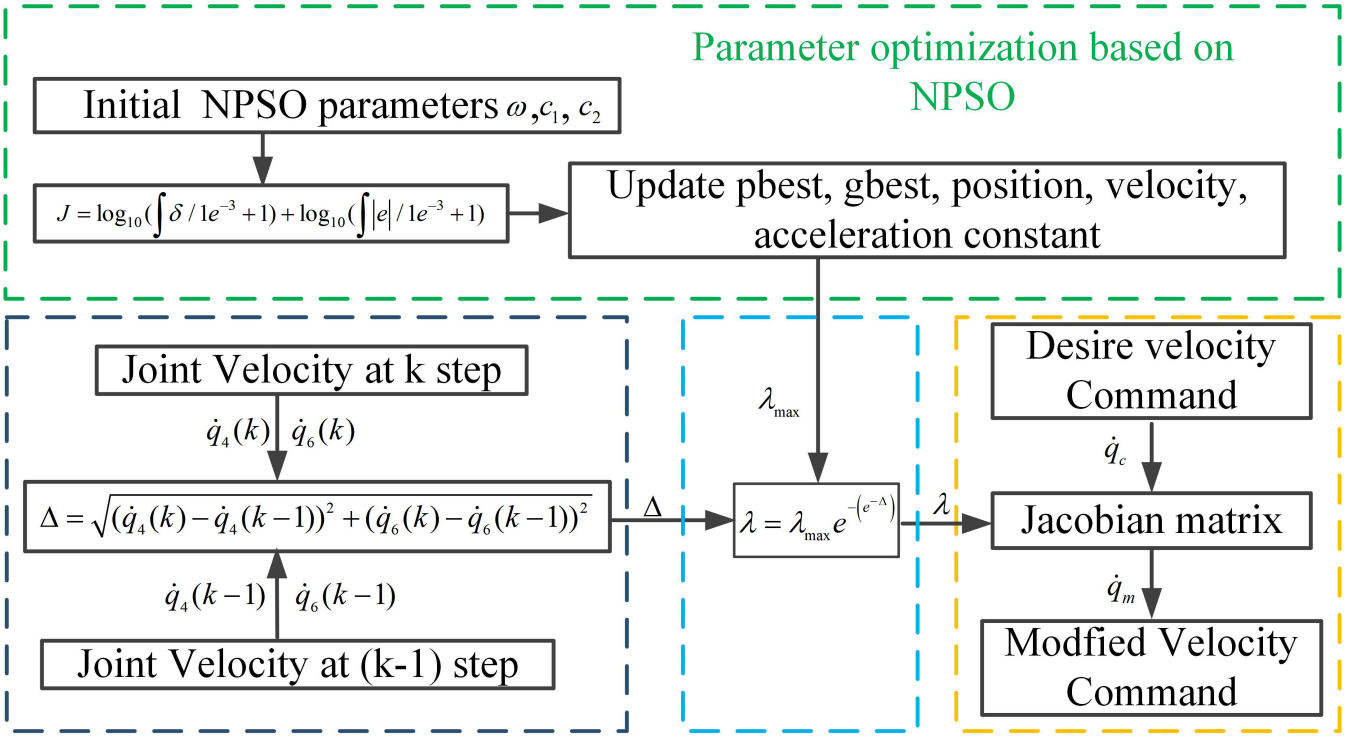


Fig. 6. Flow diagram of V-DLS combined with NPSO

intended goals, emphasizing its resilience and effectiveness. A comprehensive analysis is presented to elucidate the observations and address any disparities between the simulation and experimental data.

A. Simulation verification

1) Design procedure: To verify the effectiveness of the proposed method, the simulation experiment setup is conducted as follows:

Step 1: The initial position of the robot arm is set away from the singular area;

Step 2: The robot arm is directed to cross the singular area along a straight line at a predetermined constant velocity, with the end effector moving along the negative direction of Y-axis of the inertial coordinate system (Fig. 7);

Step 3: The end-effector velocity tracking error and the joint angle mutation are measured.

The initial coordinates of the end effector in the inertial system are set as (x, y, z).

$x = 0.7954m, y = -0.2873m, z = 1.19m$

The joint angles corresponding to the kinematics of the robotic arm are expressed as follows:

$\theta_1 = -28.8°, \theta_2 = -86.4°, \theta_3 = -3.6°$

$\theta_4 = 90°, \theta_5 = 25.2°, \theta_6 = 0°$

The initial search range of the NPSO algorithm is determined through a process of trial and error as $\lambda_{max} \in [0, 0.5]$, ω_{min} = 0.1, ω_{max} = 0.9, c_{1d} = c_{2d}= 0.5, and c_{1u} = c_{2u} = 2, where iteration times is 50s. According to the optimization algorithm described in this study, the optimal value of λ_{max} are 0.05, 0.08 and 0.12, respectively, which associated with various end-effector velocities 0.02 m/s, 0.015m/s and 0.01m/s.

Remark 9. *The λ_{max} computed by NPSO exhibits an decreasing trend with the rise in the end effector velocity. As the velocity escalates, the system becomes more responsive to variations in J coefficients, leading to a decrease in λ_{max}.*

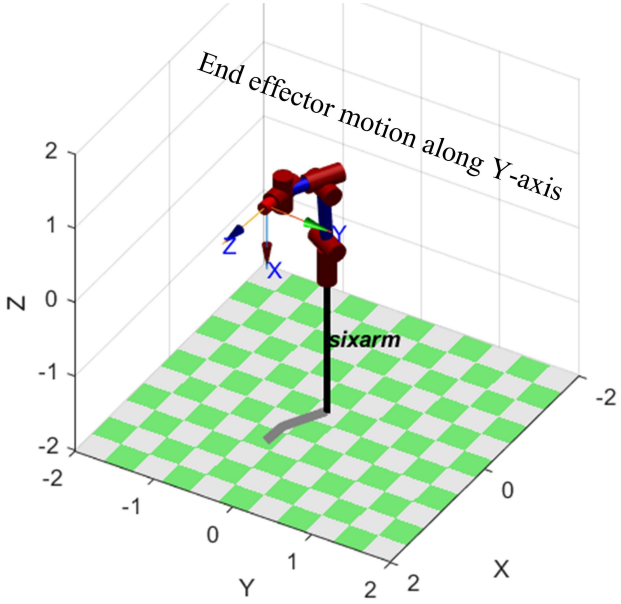


Fig. 7. Initial pose of the manipulator

2) Simulation results: In this section, the performance of the traditional DLS method and V-DLS is compared based on the velocity parameters outlined above. To effectively showcase the superiority of the algorithm proposed in this study, three scenarios are employed for simulation verification.

Case1: Velocity at 0.02m/s

The λ calculated at 0.02 m/s using V-DLS is illustrated in Fig. 8. In Fig. 8, the value of λ is approximately 0.0185.

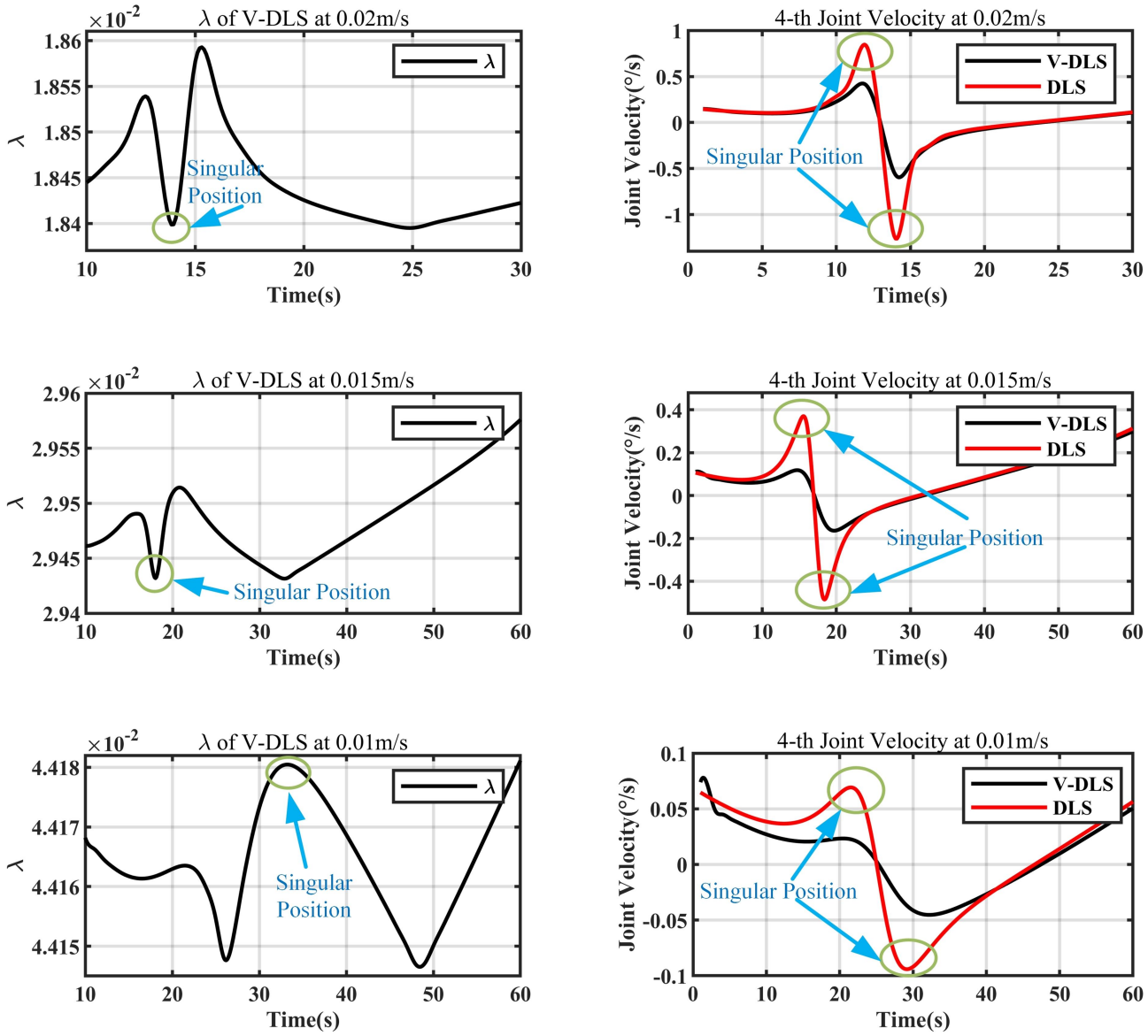


Fig. 8. Comparative simulation analysis of adaptive damping optimization in singular configuration control

Between $10s$ and $12s$, a singular configuration leads to a sudden fluctuation in λ; from $16s$ to $23s$, λ exhibits a continuous decrease; and from $24s$ to $28s$, λ shows a gradual increase. This increase may be attributed to the necessity of higher joint velocities when the end effector reaches the boundary configuration.

In Fig. 8, the velocity of 4-th joint exhibits a gradual increase near the singular point. A peak value is observed between $10s$ and $15s$, during which the manipulator assumes a singular configuration. The maximum velocities of DLS and V-DLS are recorded as 0.8 m/s and 0.4 m/s, respectively. The results indicate that V-DLS outperforms DLS in mitigating velocity fluctuations.

Case2: Velocity at 0.015 m/s

The λ calculated at 0.015 m/s using V-DLS is illustrated in Fig. 8. In Fig. 8, the value of λ increases to 2.95×10^{-2} between $10s$ and $25s$, indicating the manipulator is approaching a singular configuration. Subsequently, there is a gradual increase from $40s$ to $60s$ as the manipulator moves towards the edge position, necessitating an increase in joint velocity.

The velocity profiles for the joints when the velocity is set to 0.015 m/s are illustrated in Fig.8. Specifically, the velocity of the 4-th joint of DLS reaches 0.35 m/s at the 15th second, whereas the velocity of V-DLS is 0.15 m/s at the same time point. These results indicate that the performance of V-DLS method proposed in this study surpasses that of the conventional DLS.

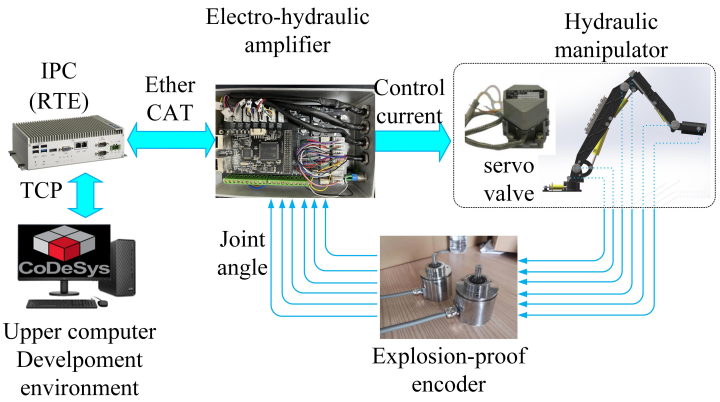
Case3: Velocity at 0.01 m/s

The λ calculated at 0.01 m/s using V-DLS is illustrated in Fig. 8. In Fig. 8, λ attains a maximum value of 4.4×10^{-2} at $35s$, indicating the initiation of the manipulator's entry into the singular configuration. The joint velocity results are depicted in Fig. 8. Analysis of the results presented in Fig. 8. indicates that V-DLS exhibits superior performance in velocity planning.

B. Experimental setup

To assess the proposed methodology, experiments were carried out on the experimental platform illustrated in Fig. 9 (a). The experiment utilized a 6-DOF series hydraulic

(a) 6-DOF hydraulic manipulator



(b) Control schematic diagram

Fig. 9. Experiment setup ([32] and owned by the author(s))

manipulator with an end load of 35 kg and an arm span of 1550 mm. The hardware components of the control system are illustrated in Fig. 9 (b). The platform comprises a hydraulic manipulator and a control module. The hydraulic manipulator is equipped with 6 servo valves (HY110) and 6 rotary encoders (SSI). The control module consists of a host computer, a servo amplifier, and an industrial computer (IPC). The sampling time is set at 250 Hz (4 ms). The real-time control software is developed using Codesys, which provides a real-time environment for control software on the Windows 10 operating system [32].

Fig. 10 depict the experimental procedure at the commencement, intermediate, and concluding positions, respectively. These figures delineate the advancement of the robotic arm's motions during the trial, emphasizing the variations in location and arrangement across the different phases.

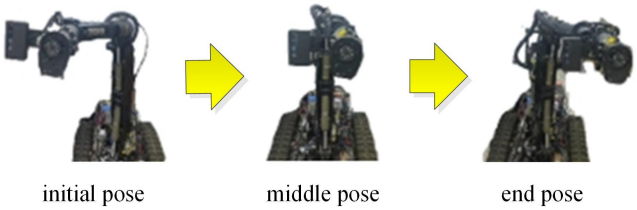


Fig. 10. Triple-phase robot pose transition: from initial to terminal configuration

C. Comparative experimental results

To assess the efficacy and practicality of the developed control algorithm, two methods are compared as follows:

Case 1: Velocity at 0.02 m/s

The λ calculated at 0.02 m/s using V-DLS and the feedback of joint velocity is shown in Fig. 11.

In Fig. 11, as the robot approaches the singular point, the value of λ is approximately 1.84×10^{-2}. Analysis of the angular velocity curve of the 4-th joint reveals that the angular velocity varies within the range of [0.04, -0.02]. Furthermore, the experimental results align with the results obtained from the simulation

Case 2: Velocity at 0.015m/s

The λ calculated at 0.015 m/s using V-DLS and the feedback of joint velocity is shown in Fig. 11.

In Fig. 11, the value of λ is approximately 2.943×10^{-2} when the manipulator is positioned at the singular point. Additionally, Fig. 11 illustrates that the velocity change in the 4-th joint ranges from 0.4°$/s$ to -0.4°$/s$.

Case 3: Velocity at 0.01 m/s

The λ calculated at 0.01 m/s using V-DLS and the feedback of joint velocity is shown in Fig. 11.

In Fig. 11, the λ value at the singular point of the robot arm is approximately 4.414×10^{-2} during the movement. Furthermore, Fig. 11 illustrates the change in velocity of the 4-th joint from 0.1°$/s$ to -0.1°$/s$.

D. Comparative pressure fluctuation

The pressure fluctuation comparison results are presented in Table II. In the experimental scenario without avoidance, the pressure fluctuation measures 0.2 MPa. Conversely, in the scheme incorporating DLS, the pressure fluctuation is reduced to 0.1 MPa, and in V-DLS scheme, the pressure fluctuation further decreases to 0.01 MPa.

TABLE II
PRESSURE FLUCTUATION RESULTS

Method	Fluctuation Amplitude
Without DLS	0.2 MPa
DLS	0.1 Mpa
V-DLS	0.01 Mpa

In summary, the optimization method presented in this paper offers advantages in mitigating singularities in robotic arm systems. By effectively tackling the issues related to singular configurations, this method guarantees a more seamless and stable operation. A key advantage of this strategy lies in its capacity to reduce sudden and drastic variations, or mutations, in joint velocities that commonly occur in proximity to singular configurations. Such abrupt changes in velocity have the potential to induce significant mechanical strain and instability in the hydraulic system, which could result in damage and compromise the manipulator's overall performance and longevity.

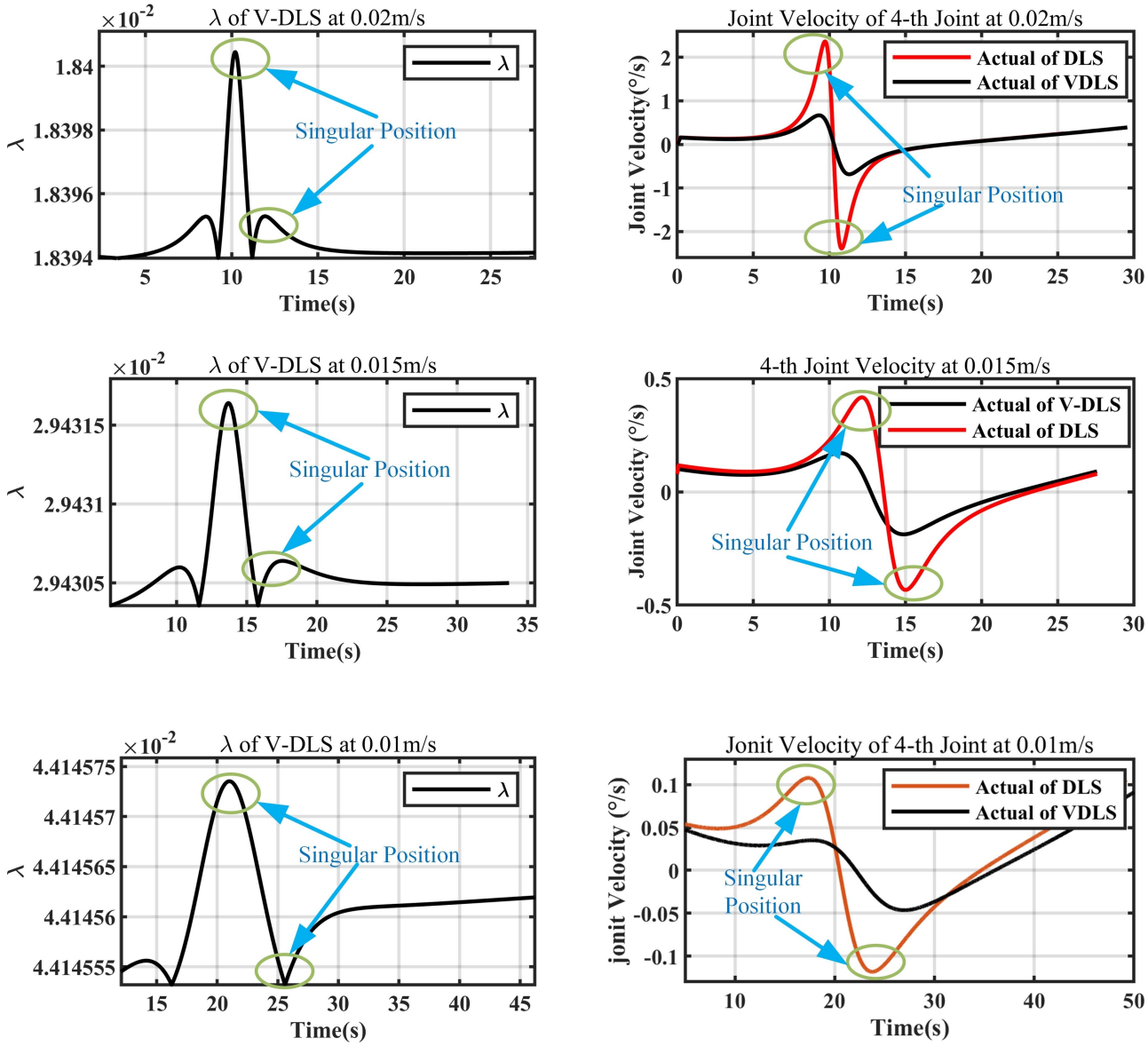


Fig. 11. Experimental validation of adaptive DLS performance in singular configuration control

VI. Conclusion

This study focuses on a self-developed 6-DOF hydraulic manipulator and investigates the effect of the damping coefficient selection on the motion performance of serial manipulators when employing DLS method to address singular configuration issues. The analysis focuses on the singular configurations of the wrist within its operational space based on the robot model. A V-DLS is introduced, which utilizes the rate of change of joint velocity to calculate the damping coefficient and optimizes it through a NPSO algorithm. This methodology effectively mitigates the occurrence of singular configurations in the manipulator. In comparison to conventional DLS methods, the proposed algorithm exhibits enhancements in reducing velocity errors and minimizing sudden velocity changes, thereby ensuring a more stable and precise control of the hydraulic manipulator.

References

[1] H. Wang, Z. Zhou, X. Zhong, and Q. Chen, "Singular Configuration Analysis and Singularity Avoidance with Application in an Intelligent Robotic Manipulator," *Sensors*, vol.22, no.1, pp.1239, 2022.

[2] H. Han and J. Park, "Robot Control near Singularity and Joint Limit Using a Continuous Task Transition Algorithm Regular Paper," *International Journal of Advanced Robotic Systems*, vol.10, pp.10, 2013.

[3] D. Halperin D, L. Kavraki, K. Solovey, "Robotics," *Handbook of discrete and computational geometry*. Chapman and Hall/CRC: pp.1343-1376, 2017.

[4] K. Zheng, R. Sun, F. Li, Y. Liu, D. Li, and R. Song, "Research and application of the centralized drive and control system for a hydraulic manipulator," *Journal of Jilin University (Engineering and technology Edition)*, vol.54, no.11, pp.3358-3371, 2024.

[5] Y. Nakamura, H. Hanafusa and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol.6, no.2, pp.3-15, 1987.

[6] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol.16, no.1, pp.93-101, 1986.

[7] A. Maciejewski and C. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations." *Journal of Robotic systems* vol.5, no.6, pp.527-552, 1988.

[8] L. Phuoc, P. Martinet, S. Lee, and H. Kim, "Damped least square based genetic algorithm with Gaussian distribution of damping factor for singularity-robust inverse kinematics," *Journal of Mechanical Science and Technology*, vol.22, pp.1330-1338, 2008.

[9] S. Buss, J. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol.10, no.3, pp.37-49, 2005.

[10] R. Jin, P. Rocco and Y. Geng, "Cartesian trajectory planning of space

robots using a multi-objective optimization," *Aerospace Science and Technology*, vol.108, pp.106360, 2021

[11] Y. Zang, W. Hui, and Q. Yong, "GIL partial discharge localization method based on 3D optical fingerprint and NPSO-KELM," *Proceedings of the CSEE*, vol.40, no.20, pp.6754-6764, 2020.

[12] M. Alhussein and S. Haider, "Improved particle swarm optimization based on velocity clamping and particle penalization," *2015 3rd international conference on artificial intelligence, modelling and simulation (AIMS)*, pp.61-64, 2005.

[13] P. Kalra, P. Mahapatra and D. Aggarwal, "An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots," *Mechanism and Machine Theory*, vol.41, pp.1213-1229, 2006.

[14] S. Tabandeh, C. Clark, W.A Melek, "A genetic algorithm approach to solve for multiple solutions of inverse kinematics using adaptive niching and clustering," *2006 IEEE International Conference on Evolutionary Computation*, pp.1815-1822, 2006.

[15] A. Dobnikar, N. Steele, and D. Pearson, "A niched-penalty approach for constraint handling in genetic algorithms," *Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Portorož, Slovenia*, pp.235-243, 1999.

[16] H. Huang , S. Xu, and C. Wu, "A hybrid swarm intelligence of artificial immune system tuned with Taguchi–genetic algorithm and its field-programmable gate array realization to optimal inverse kinematics for an articulated industrial robotic manipulator," *Advances in Mechanical Engineering*, vol.8, no.1, 2016.

[17] R. Köker, "A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization," *Information Sciences*, vol.222, pp.528-543, 2013.

[18] Z. Zhou, H. Guo, and Y. Wang, "Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm," *International Journal of Advanced Robotic Systems*, vol.15, no.4, 2018.

[19] G. Zhao, P. Zhang, G. Ma, and W. Xiao, "System identification of the nonlinear residual errors of an industrial robot using massive measurements," *Robotics and Computer-Integrated Manufacturing*, vol.59, pp.104-114, 2019.

[20] A. Almusawi, L. Dülger and S. Kapucu, "A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242)," *Computational Intelligence and Neuroscience*, vol.1, 2016.

[21] X. Wang, X. Liu, and L. Chen, "Deep-learning damped least squares method for inverse kinematics of redundant robots," *Measurement*, vol.171, 2021.

[22] S. Tejomurtula, and S. Kak, "Inverse kinematics in robotics using neural networks," *Information Sciences*, vol.116, no.2, pp.147-164, 1999.

[23] B. Karlik, and S. Aydin, "An improved approach to the solution of inverse kinematics problems for robot manipulators," *Engineering Applications of Artificial Intelligence*, vol.13, no.2, pp.159-164, 2000.

[24] M. Krishnan, and S. Ashok, "Kinematic analysis and validation of an industrial robot manipulator," *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*, pp.1393-1399, 2019.

[25] Y. Zheng, "Study on control System Design and Stability Control for Mobile Hydraulic Manipulator," *ShanDong University* , 2023.

[26] O. Omisore, S. Han, and L. Ren, "Deeply-learnt damped least-squares (DL-DLS) method for inverse kinematics of snake-like robots," *Neural Networks*, vol.107: PP.34-47, 2018.

[27] R. Köker, T. Çakar, and Y. Sari, "A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators," *Engineering with Computers*, vol.30 pp.641-649, 2014.

[28] H. Liu, and T. Zhang, "A new approach to avoid singularities of 6-DOF industrial robot," *2010 IEEE International Conference on Mechatronics and Automation*, pp.247-251, 2010.

[29] M. Safeea, R. Bearee, and P. Neto, "A modified DLS scheme with controlled cyclic solution for inverse kinematics in redundant robots," *IEEE Transactions on Industrial Informatics*, vol.17, no.12, pp.8014-8023, 2021.

[30] M. Tucker, and N. Perreira, "Generalized inverses for robotic manipulators," *Mechanism and Machine Theory*, vol.22, no.6, pp.507-514, 1987.

[31] A. Colomé, and C. Torras, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Transactions On Mechatronics*, vol.20, no.2, pp.944-955, 2014.

[32] Y. Zheng, R. Sun, F. Li, Y. Liu, R. Song, and Y. Li, "Parameter identification and position control for helical hydraulic rotary actuators based on particle swarm optimization," *Mechatronics*, vol.94, 2023.

[33] Y. Chang, C. Ko, "A PSO method with nonlinear time-varying evolution based on neural network for design of optimal harmonic filters," *Expert Systems with Applications*, vol.36, no.3, pp.6809-6816, 2009.

[34] N. Rokbani, and A. Alimi, "Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis," *Procedia Engineering*, vol.64, pp.1602-1611, 2013.