

# 基于INOA算法的塑料激光焊接熔深控制

陈 飞, 周俊杰

江西理工大学电气工程与自动化学院, 江西 赣州

收稿日期: 2026年3月23日; 录用日期: 2026年4月29日; 发布日期: 2026年5月11日

## 摘 要

针对熔深控制系统非线性、滞后性的特点, 本文引入FOPID控制器和预测算法DMC进行控制, 同时采用改进星鸦算法对控制器参数进行迭代寻优, 以解决参数整定不合理导致的控制性能不佳、超调量过大、耗时过长等问题。星鸦算法的改进主要包括两个方面: 一是针对初始种群多样性不足的缺陷, 采用拉丁超立方抽样的方法, 保障初始种群在解空间内的分布多样性; 二是针对算法容易陷入局部最优的问题, 在探索与开发阶段引入全局引导机制, 提升算法全局寻优性能。仿真结果表明, INOA-DMC-FOPID预测控制策略可以有效提高控制效果, 能够满足焊接实际生产要求, 具备一定的工程实用性。

## 关键词

FOPID控制器, 塑料激光焊接, INOA算法

# Penetration Depth Control of Plastic Laser Welding Based on INOA Algorithm

Fei Chen, Junjie Zhou

School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou Jiangxi

Received: March 23, 2026; accepted: April 29, 2026; published: May 11, 2026

## Abstract

In view of the nonlinear and time-delay characteristics of the weld penetration control system, this paper introduces a Fractional Order PID controller and the Dynamic Matrix Control prediction algorithm for system control. Meanwhile, an improved Nutcracker Optimization Algorithm is adopted to carry out iterative optimization of the controller parameters, so as to solve the problems including poor control performance, excessive overshoot, and long settling time caused by improper parameter tuning. The improvements of the NOA mainly include two aspects: First, aiming at the defect of insufficient diversity of the initial population, Latin Hypercube Sampling is used to ensure

the uniform and diverse distribution of the initial population in the solution space. Second, to address the problem that the algorithm is prone to fall into local optimum, a global guidance mechanism is introduced in the exploration and exploitation phases to enhance the global optimization performance of the algorithm. The simulation results show that the proposed INOA-DMC-FOPID predictive control strategy can effectively improve the control performance, meet the actual requirements of welding production, and has certain engineering practicability.

## Keywords

FOPID Controller, Plastic Laser Welding, INO Algorithm

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着焊接技术不断进步、生产需求持续增长、设备成本稳步下降,原本主要用于金属材料焊接的激光焊接技术,已成功扩展至塑料材料焊接领域[1]。更重要的是,塑料材料具备突出的轻量化优势,在工业制造中逐步替代传统金属,尤其在迅猛发展的汽车行业已有成熟广泛的应用,因此塑料激光焊接技术也得到了长足发展并日趋普及[2]。如今我国汽车与塑料制品产销量均稳居世界首位,两大产业互为促进、形成良性循环。

在汽车轻量化零部件、电子精密外壳等产品的焊接过程中,熔深控制精度是决定产品品质的核心指标之一。熔深偏差超标极易引发焊接失效,甚至导致产品报废。传统激光焊接熔深控制系统中多采用PID控制,其根据输出反馈的熔深值与设定值偏差,实时调节激光功率等因素实现熔深的闭环稳定控制。然而,在熔深控制过程具有非线性和滞后性的背景下,该控制器的控制效果欠佳,易出现熔深不足、过熔或工件变形等问题。

为实现熔深的高精度控制,本文选用分数阶PID控制器(Fractional Order Proportional-Integral-Derivative Controller, FOPID) [3]替代传统PID控制器。与PID控制器相比,FOPID控制器具有更高的控制自由度,可以更有效地抑制系统的稳态误差,提高熔深轨迹的跟踪精度。FOPID控制器和传统PID存在同样的问题:参数整定难度大,靠手动调试很难达到理想的控制效果。随着优化算法的不断发展,当前已有多种优化算法可用于解决该问题,例如粒子群算法(Particle Swarm Optimization, PSO) [4]、灰狼算法(Grey Wolf Optimizer, GWO) [5]、鲸鱼优化算法(Whale Optimization Algorithm, WOA) [6],均能为FOPID控制器进行参数寻优。星鸦算法(Nutcracker Optimization Algorithm, NOA),是Mohamed和Abdel-Basset等人于2023年基于星鸦觅食规律提出的一种元启发式算法[7],具备优异的寻优性能,可以应用于FOPID控制参数的优化。在此基础上,针对时滞对控制效果的负面影响,采用动态矩阵控制方法(Dynamic Matrix Control, DMC) [8]建立系统预测模型,通过对熔深的动态变化趋势进行预测,补偿时滞带来的控制偏差,实现熔深的高精度最优控制。

## 2. 控制算法

### 2.1. FOPID 控制器

FOPID控制器由I. Podlubny教授于1994年首次系统提出。与传统整数阶PID控制器相比,FOPID

控制器将积分阶次  $\lambda$ 、微分阶次  $\mu$  从固定整数扩展为可独立整定的实数, 具备更大的参数调节自由度, 其传递函数表达式如式(1)所示。

$$G(s) = K_p + K_I s^{-\lambda} + K_D s^\mu \quad (1)$$

式中:  $\lambda$  和  $\mu$  取(0,2)内的任意实数;  $K_p$ 、 $K_I$ 、 $K_D$  分别为比例系数、积分系数与微分系数。基于该传递函数搭建的分数阶控制系统结构图 1 所示。

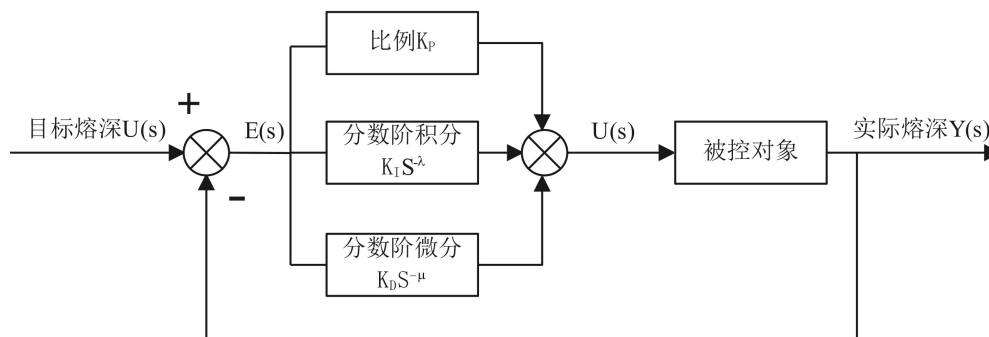


Figure 1. Controller block diagram  
图 1. FOPID 控制器结构图

## 2.2. DMC 算法原理

DMC 以阶跃响应模型预测未来输出为核心, 以历史与当前时刻的系统输入输出偏差为依据, 完成未来时域内的输出偏差超前预判; 同时采取反馈校正和有限时域滚动优化的思想, 完成当前控制周期内最优控制增量的求解, 并据此计算出在控制时域内的最优控制序列。

### (1) 预测模型

DMC 首先采集被控对象单位阶跃响应的采样值  $a_i = a(iT)$ ,  $i=1,2,\dots$ , 其中  $T$  是采样周期。根据采样值建立模型的预测向量为  $a_i = [a_1 \cdots a_N]^T$ ,  $N$  表示建模时域长度。基于线性系统叠加原理, 结合预测时域  $P$ 、控制时域  $M$  (满足  $M \leq P \leq N$ ), 构建核心预测方程:

$$Y_m(k) = Y_0(k) + A\Delta U(k) \quad (2)$$

$$Y_m(k) = [Y_m(k+1), \dots, Y_m(k+p)]^T \quad (3)$$

$$Y_0(k) = [Y_0(k+1), \dots, Y_0(k+p)]^T \quad (4)$$

$$\Delta U(k) = [\Delta U(k,k), \dots, \Delta U(k+M-1,k)]^T \quad (5)$$

$$A = \begin{bmatrix} a_1 & & & & \\ a_2 & a_1 & & & \\ \vdots & \vdots & & & \\ a_p & a_{p-1} & \cdots & a_{p-M+1} & \end{bmatrix} \quad (6)$$

式中:  $Y_m(k)$  是  $k$  时刻模型未来  $P$  步预测输出向量;  $Y_0(k)$  是模型无新增控制量时的初始输出向量;  $\Delta U(k)$  为待优化的  $M$  步控制增量序列;  $A$  为由阶跃响应系数构成的三角动态矩阵, 可定量刻画控制增量对模型未来输出的动态影响, 为滚动优化提供核心输入输出映射。在对系统输出响应曲线进行观察后, 确定  $T=0.2s$ ,  $N=40$ ,  $P=30$ ,  $M=10$ 。

## (2) 滚动优化

DMC 采用滚动时域优化策略, 区别于传统最优控制的全局离线优化模式, 可在每个采样周期内, 基于被控系统当前状态, 在线求解有限预测区间内的最优控制问题。该算法以预测输出对参考轨迹的平滑跟踪、抑制控制量的剧烈波动为双目标, 构建加权二次型性能指标式  $\min J(k)$ ; 通过优化指标, 确定出未来  $M$  步的控制增量序列  $W(k)$ , 使系统未来输出的预测值最大程度贴合参考轨迹, 其优化求解的核心方程如下:

$$\min J(k) = \|W(k) - Y_m(k)\|_Q^2 + \|\Delta U(k)\|_R^2 \quad (7)$$

$$W(k) = [W(k+1) \quad W(k+2) \quad \cdots \quad W(k+P)] \quad (8)$$

式中,  $W(k)$  是参考轨迹向量,  $Q$ 、 $R$  分别是跟踪误差权重矩阵与控制增量权重矩阵。最优控制序列求解完成后, 控制器仅将首项控制增量作用于被控对象, 下一采样周期则基于更新后的系统状态开展新一轮滚动优化。

## (3) 反馈校正

反馈校正是 DMC 保证闭环鲁棒性的核心, 用于修正模型失配、外部扰动及测量噪声带来的预测误差。每个采样周期内, 通过对比被控对象实际输出  $Y(k+1)$  与预测值  $Y_m(k+1|k)$ , 实时校正预测误差, 计算过程见式(9)。

$$E(k+1) = Y(k+1) - Y_m(k+1|k) \quad (9)$$

基于得到的预测误差, 我们用  $N$  维权系数校正向量  $h = [h_1, h_2, \dots, h_N]^T$  来修正预测值, 整个校正过程是实时滚动进行的, 具体公式如式(10)所示:

$$Y_p(k+1) = Y_m(k+1) + hE(k+1) \quad (10)$$

其中  $h$  是校正系数, 一般取 1; 校正后的输出预测向量  $Y_p(k+1)$  经过移位处理后, 就可以作为下一个采样周期滚动优化的初始预测向量, 向量表达式见式(11), 式中  $S$  为移位矩阵。

$$Y_p(k+2) = SY_p(k+1) \quad (11)$$

$$S = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (12)$$

## 2.3. NOA 算法

NOA 算法的迭代寻优过程是模仿星鸦觅食规律形成的, 分为两个核心策略阶段: 春季觅食与食物储存策略、冬季食物缓存搜索与恢复策略。在算法单代迭代过程中, 先按照预设的选择概率确定本次迭代采用的寻优策略, 再从该策略对应的两种更新模式中择一完成种群内个体的位置更新。

### 2.3.1. 种群初始化

NOA 算法的初始种群, 采用可行解空间内均匀随机方式生成, 作为迭代寻优的初始起点, 具体生成公式如下。

$$X_j^i = L_j + (U_j - L_j)rand \quad (13)$$

式中:  $X_j^t$  是第  $j$  维的位置,  $U_j$  和  $L_j$  是第  $j$  维的上下限。

### 2.3.2. 觅食、储存策略

(1) 觅食策略, 也称探索阶段 1

在探索阶段 1, 星鸦首先会检测初始位置, 如果有食物, 则带到储藏区, 若没有, 则到区域内的其它松树上寻找。对该行为的建模如下:

$$\overline{X_i^{t+1}} = \begin{cases} X_{i,j}^t & \tau_1 < \tau_2 \\ X_{m,j}^t + \gamma(X_{A,j}^t - X_{B,j}^t) + \mu(r^2 * U_j - L_j) & t \leq T_{\max}/2 \\ X_{C,j}^t + \mu(X_{A,j}^t - X_{B,j}^t) + \mu(r_1 < \delta)(r^2 * U_j - L_j) & \text{其他} \end{cases} \quad (14)$$

$X_{i,j}^t$  为第  $t$  代第  $i$  个星鸦的第  $j$  维;  $X_{m,j}^t$  是第  $j$  维平均值;  $\gamma$  是根据莱维飞行产生的随机数;  $A$ 、 $C$  和  $B$  是从种群中随机选择的三个体;  $r$ ,  $r_1$  为  $[0, 1]$  范围内的随机实数。

$$\mu = \begin{cases} \tau_3 & r_1 < r_2 \\ \tau_4 & r_2 < r_3 \\ \tau_5 & r_1 < r_3 \end{cases} \quad (15)$$

$\tau_3$  为 0~1 之间的随机数;  $\tau_4$  是基于正态分布生成的数;  $\tau_5$  为莱维飞行生成的数;  $r_2$ ,  $r_3$  为  $[0, 1]$  范围内的随机实数。

(2) 储存策略, 也称开发阶段 1

星鸦将觅食阶段获得的食物运到不同储藏区, 其该行为的建模如下:

$$\overline{X_i^{t+1}(\text{new})} = \begin{cases} X_{i,j}^t + \mu * (X_{best}^t - X_i^t) * |\lambda| + r_1 * (X_A^t - X_B^t) & \tau_1 < \tau_2 \\ X_{best}^t + \mu * (X_{A,j}^t - X_{B,j}^t) & \tau_1 < \tau_3 \\ X_{best}^t * l & \text{其他} \end{cases} \quad (16)$$

$\overline{X_i^{t+1}(\text{new})}$  是当代个体的新位置;  $\lambda$  是莱维飞行产生的随机数;  $l$  是从 1 到 0 线性递减的因子。

### 2.3.3. 缓存搜索、恢复策略

(1) 参考点计算

星鸦在冬天来临时会以两个空间参考点为标记, 找回此前储藏的食物, 食物储藏位置的参考点矩阵定义如式(17)所示。为进一步优化星鸦寻食过程的探索能力, 通过式(18)与式(19)两个更新方程, 完成两个空间参考点的生成。

$$\begin{bmatrix} \overline{RP_{1,1}^t} & \overline{RP_{1,2}^t} \\ \vdots & \vdots \\ \overline{RP_{i,1}^t} & \overline{RP_{i,2}^t} \\ \vdots & \vdots \\ \overline{RP_{N,1}^t} & \overline{RP_{N,2}^t} \\ \vdots & \vdots \end{bmatrix} \quad (17)$$

第一层参考点为:

$$\overline{RP_{i,1}^t} = \begin{cases} X_i^t + \alpha * \cos \theta * (\overline{X_A^t} - \overline{X_B^t}) + \alpha * RP & \theta = \pi/2 \\ X_i^t + \alpha * \cos \theta * (\overline{X_A^t} - \overline{X_B^t}) & \text{其他} \end{cases} \quad (18)$$

第二层参考点为:

$$\overline{RP}_{i,2} = \begin{cases} X_i^t + \left\{ \alpha * \cos \theta * \left[ (\bar{U} - \bar{L}) * \tau_3 + \bar{L} \right] + \alpha * RP \right\} * \bar{U}_2 & \theta = \pi/2 \\ X_i^t + \alpha * \cos \theta * \left[ (\bar{U} - \bar{L}) * \tau_3 + \bar{L} \right] * \bar{U}_2 & \text{其他} \end{cases} \quad (19)$$

式中:  $\overline{RP}_{i,1}$  和  $\overline{RP}_{i,2}$  表示当前第  $t$  代第  $i$  个星鸦的缓存位置。 $\alpha$  是从 1 线性减到 0 的数;  $\theta$  是在  $[0, \pi]$  范围内的随机值;  $U_2$  是 0 或 1 的随机数。

(2) 缓存搜索策略, 也称探索阶段 2

冬季来临后, 星鸦会从储藏模式转换到搜索模式, 开启第二次探索, 根据参考点和最优位置找回食物。

$$\overline{X}_i^{t+1} = \begin{cases} X_{i,j}^t & \tau_3 < \tau_4 \\ X_{i,j}^t + r_1 (X_{best,j}^t - X_{i,j}^t) + r_2 (\overline{RP}_{i,1}^t - X_{i,j}^t) & \tau_5 < \tau_6 \\ X_{i,j}^t + r_1 (X_{best,j}^t - X_{i,j}^t) + r_2 (\overline{RP}_{i,2}^t - X_{i,j}^t) & \text{其他} \end{cases} \quad (20)$$

$C$  是随机选择的个体;  $X_{best,j}^t$  为第  $t$  代最优个体。

(3) 恢复策略, 也称开发阶段 2

在冬季的开发阶段 2, 星鸦以参考点为搜索依据: 在第一个参考点和第二个参考点中选择最优搜索, 未找到食物储藏位置, 则返回原处。搜索食物位置的更新方程如下。

$$\overline{X}_i^{t+1} = \begin{cases} \text{式(22)} & f(\text{式(22)}) < f(\text{式(23)}) \\ \text{式(23)} & \text{其他} \end{cases} \quad (21)$$

$$\overline{X}_i^{t+1} = \begin{cases} \overline{X}_i^t & f(\overline{X}_i^t) < f(\overline{RP}_{i,1}^t) \\ \overline{RP}_{i,1}^t & \text{其他} \end{cases} \quad (22)$$

$$\overline{X}_i^{t+1} = \begin{cases} \overline{X}_i^t & f(\overline{X}_i^t) < f(\overline{RP}_{i,2}^t) \\ \overline{RP}_{i,2}^t & \text{其他} \end{cases} \quad (23)$$

## 2.4. INOA 算法

### 2.4.1. LHS 采样初始化

由于 NOA 初始值的随机性, 很难保证初始群体在求解空间中的平均分布, 容易出现局部极值, 收敛速度较慢。针对这一问题, 利用拉丁超立方体采样(Latin Hypercube Sampling, LHS)的多层次均匀抽样的特点, 产生更均匀分布和更优的初始群体, 为算法高效寻优奠定基础[9]。

LHS 采样针对给定的  $d$  维输入变量空间  $X = (x_1, x_2, \dots, x_d)$ , 设每个输入变量  $x_j$  存在一个定义域:  $x_j \in [x_j^{\min}, x_j^{\max}]$ ,  $j$  是 1 到  $d$  的整数, 从该  $d$  维空间中采样  $N$  个样本点形成的样本矩阵如下:

$$S = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_d^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \dots & x_d^{(N)} \end{bmatrix} \quad (24)$$

LHS 的核心原理, 是通过分层抽样保证样本的空间均匀性: 对  $d$  维输入空间的每个变量维度, 均将其定义域等分为  $N$  个子区间, 且确保每个样本在任一维度上仅落入唯一的子区间, 实现每个子区间内有且仅有一个样本点。对第  $j$  个变量, 其定义域  $[x_j^{\max}, x_j^{\min}]$  被均分为  $N$  段后的区间宽度为:

$$\Delta_j = \frac{x_j^{\max} - x_j^{\min}}{N} \quad (25)$$

对每一个变量维度, 均需构造两组独立的排列序列  $\pi_j = \{\pi_j, \dots, \pi_j^{(N)}\}$ : 第一组为集合  $\{0, 1, \dots, N-1\}$  的无重复随机排列; 第二组排列  $u_j = \{u_j^{(1)}, \dots, u_j^{(N)}\}$  的每个元素, 均取自  $[0, 1]$  区间内服从均匀分布、且两两互不相同的随机数。基于上述排列, 即可计算得到第  $i$  个星鸦个体在第  $j$  个变量维度上的坐标值  $x_j^{(i)}$ 。

$$x_j^{(i)} = x_j^{\min} + (\pi_j^{(i)} + u_j^{(i)}) \Delta_j \quad (26)$$

排列  $\pi_j$  用于界定试验样本点所属的超立方空间, 即其可确定  $j$  维变量下样本点对应的均匀分段区间; 而随机扰动项  $u_j$  的引入, 能够精准锁定试验样本点在超立方体内部的具体位置, 进一步提升每个超立方体内采样的随机均匀性, 避免样本点集中在立方体边界。

#### 2.4.2. 全局引导机制

针对 NOA 算法迭代后期易陷入局部最优、全局搜索能力弱的问题, 在算法中引入全局最优引导机制。在 NOA 算法的策略更新阶段, 为星鸦个体引入速度项  $V_i^{t+1}$  作为全局引导项, 明确种群个体的迭代方向与位移步长, 改进后的星鸦个体位置更新表达式如式(27)所示。

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (27)$$

式中速度项, 将种群全局最优  $gbest_t$  扩展为外部档案中的全局最优解集合, 通过档案中的非支配解引导星鸦个体搜索, 弥补原算法仅依赖个体历史最优  $pbest_i$  引导的搜索局限。该融合算法既保留 NOA 收敛快、精度高的优势, 又显著提升全局寻优能力, 避免迭代陷入局部最优。全局引导的速度项表达式如式(28)所示。

$$V_i^{t+1} = wV_i^t + c_1r_1(pbest_i - X_i^t) + c_2r_2(gbest_i - X_i^t) \quad (28)$$

$$w = w_{\max} - (w_{\max} - w_{\min})(t/T) \quad (29)$$

式中:  $V_i^t$  为第  $i$  个星鸦第  $t$  代的速度;  $w$  为惯性权重, 在迭代过程中线性递减;  $c_1$ 、 $c_2$  为个体与社会学习因子, 取值为 2;  $r_1$ 、 $r_2$  为  $[0, 1]$  内随机数;  $T$  为最大迭代次数。

该全局最优引导机制以 NOA 原生迭代逻辑为基础, 针对其寻优性能瓶颈进行优化, 与原生算法实现深度融合: 不改变原迭代范式与种群分工规则, 仅通过引入速度引导项完成位置更新规则的修正, 完整保留算法核心框架与基础寻优特性; 依托全局最优  $gbest_t$  扩展为外部档案中的全局最优解集合, 对原生搜索机制形成有效补充, 将原算法仅依赖个体历史最优  $pbest_i$  引导优化为跨迭代多维度引导, 缓解局部最优滞留与早熟收敛问题, 实现算法全局探索与局部开发能力的动态协同, 有效维持种群多样性。

INOA 算法流程如图 2 所示。

### 2.5. INOA 算法的 DMC-FOPID 的控制器

本文提出的 INOA-DMC-FOPID 复合控制器由预测、控制两部分构成: DMC 环节通过系统输入输出数据超前预测输出值, 补偿系统大滞后特性; INOA 算法完成 FOPID 控制器 5 个核心参数的寻优整定, 熔深控制系统结构如图 3 所示。

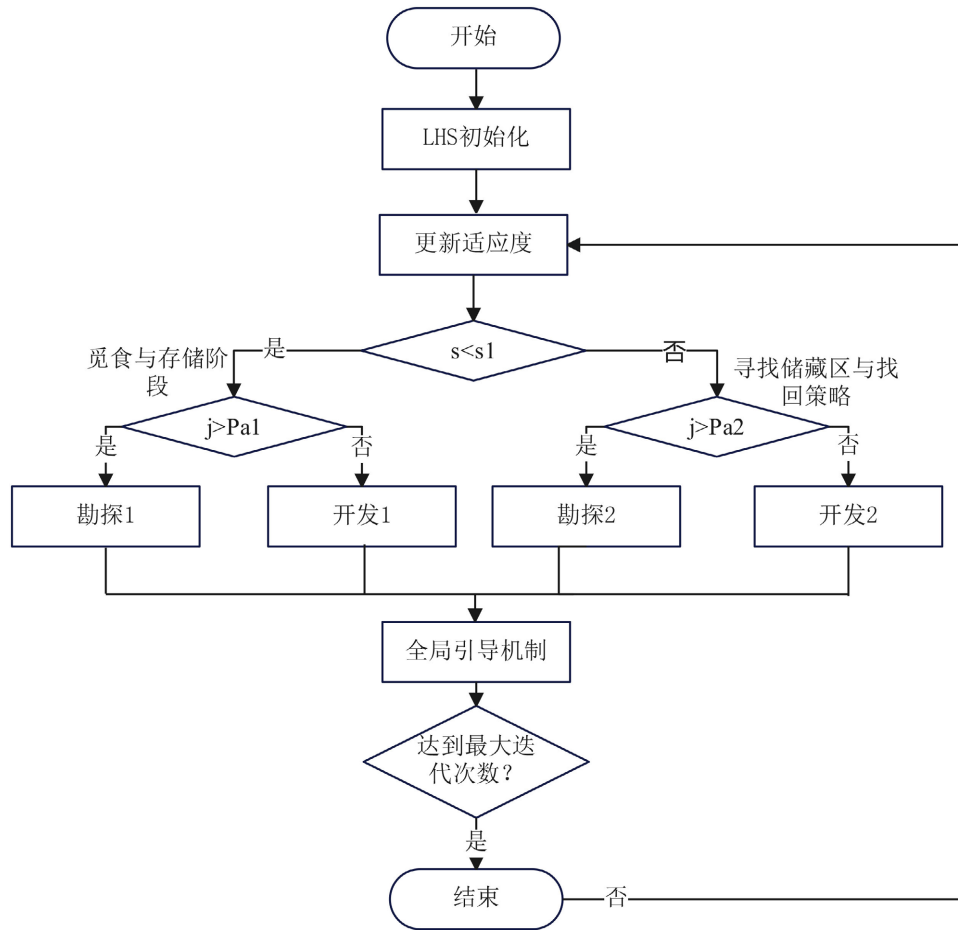


Figure 2. Flowchart of the INOA algorithm  
图 2. INOA 算法流程图

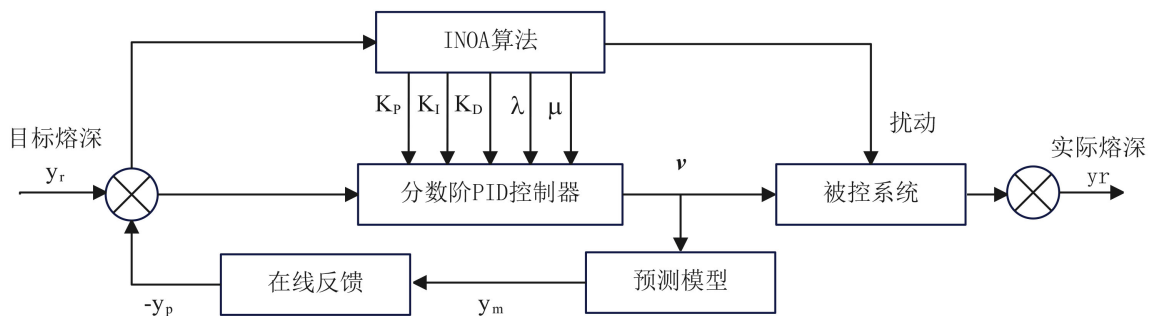


Figure 3. Block diagram of INOA-DMC-FOPID penetration control  
图 3. INOA-DMC-FOPID 熔深控制框图

在熔深控制中, 实际熔深沿参考轨迹向设定值过渡, 经 DMC 预测校正得到预测输出, 其与参考值的偏差经优化后输入 INOA, 在线调整 FOPID 参数, 输出控制量实现熔深稳定跟踪。

## 2.6. 适应度函数

本文采用时间乘绝对误差积分的 ITAE [10]与系统最大超调量作为适应度函数, 在严格约束超调量的同时, 保障控制器兼顾快速性、稳定性与稳态精度。适应度表达式如式(30)所示。

$$Fitness(t) = k_1 \int_0^t \tau |e(\tau)| d\tau + k_2 \cdot overshoot \quad (30)$$

式中:  $e(\tau)$  是输出响应曲线与参考值之间的偏差,  $overshoot$  是系统超调量。适应度函数  $Fitness(t)$  的值越小, 输出响应效果越优异。

### 3. 仿真结果分析

基于上海某激光公司提供的激光焊接设备技术规范, 以及该设备量产过程中采集的实际生产运行数据, 完成被控对象的数学模型辨识, 所得系统传递函数如下所示:

$$G(s) = \frac{0.62e^{-1.2s}}{1.693s^2 + 2.75s + 1} \quad (31)$$

操作系统采用 64 位 Windows10, CPU 为 Intel(R) Core(TM) i5-9400 CPU @ 2.90 GHz 2.90 GHz, 内存为 16 GB, 实验仿真平台采用 matlab2024a。

#### 3.1. 多维度基准函数优化结果分析

表 1 给出了 4 种算法 NOA、INOA、IGWO 及 IWOA 在不同维度下 10 次独立实验的运行结果的平均值、方差及最优值对比分析。为了更好地对比 4 种算法的收敛性, 选取两个函数(F9 和 F10)的适应度收敛曲线进行对比分析, 如图 4 所示。

**Table 1.** Comparison results of six sets of test functions

**表 1.** 6 组测试函数的对比结果

测试函数	评价指标	NOA	INOA	IWOA	IGWO
F8	平均值	-3.5139e+03	-4.0215e+03	-3.2127e+03	-3.2166e+03
	最优值	-3.8186e+03	-4.1898e+03	-4.1898e+03	-3.9693e+03
	方差	3.5642e+04	1.4755e+04	1.9571e+05	1.4107e+05
F9	平均值	1.0145e+01	1.2457e-10	5.2862e+00	4.0447e-01
	最优值	6.1823e+00	0.0000e+00	0.0000e+00	0.0000e+00
	方差	8.8233e+00	7.6117e-21	1.3345e+02	1.6358e+00
F10	平均值	2.4828e-04	1.2547e-15	3.2863e-15	1.4495e-10
	最优值	1.0054e-04	3.2145e-16	4.4409e-16	2.5981e-11
	方差	3.3441e-08	5.2049e-31	2.2440e-30	1.6440e-20
F11	平均值	3.6781e-01	8.7245e-11	1.2180e-02	1.5967e-02
	最优值	1.6972e-01	0.0000e+00	0.0000e+00	0.0000e+00
	方差	3.6111e-02	2.7193e-21	1.4835e-03	6.4801e-04
F12	平均值	7.4002e-10	3.2145e-11	6.4358e-03	2.5085e-04
	最优值	1.4186e-11	1.2457e-12	5.4460e-05	9.8388e-05
	方差	4.0458e-19	3.5059e-22	9.8063e-05	1.4259e-08
F13	平均值	3.2011e-03	1.2547e-10	1.3775e-02	4.8180e-03
	最优值	2.0342e-11	5.2147e-12	7.7826e-05	6.9465e-04
	方差	5.1138e-05	7.6117e-21	1.6387e-04	2.2056e-05

续表

dim = 20	F8	平均值	-4.3723e+03	-7.9862e+03	-7.2145e+03	-6.1258e+03
		最优值	-5.4586e+03	-8.3797e+03	-8.3797e+03	-8.3795e+03
		方差	3.1483e+05	1.2403e+05	7.3315e+05	1.0499e+06
	F9	平均值	2.7470e+01	3.2145e-08	5.2147e+00	1.2547e+01
		最优值	1.6105e+01	1.2457e-09	0.0000e+00	0.0000e+00
		方差	4.7693e+01	3.5059e-16	1.8102e+01	4.7230e+01
	F10	平均值	2.1243e+00	2.1457e-13	5.6415e-15	5.4298e-10
		最优值	4.6527e-02	8.7245e-14	4.4409e-16	1.8333e-10
		方差	2.2067e+00	1.5518e-26	4.5097e-30	1.8514e-19
	F11	平均值	1.0088e+00	1.2145e-08	5.0153e-03	1.8762e-02
		最优值	8.4237e-01	1.2457e-09	0.0000e+00	1.1102e-16
		方差	4.0893e-03	5.2500e-17	2.3156e-04	8.9723e-04
	F12	平均值	1.1478e-01	1.2547e-07	8.5737e-03	1.4139e-03
		最优值	2.9095e-04	8.7245e-09	6.7153e-04	5.7435e-04
		方差	2.9125e-02	5.2049e-15	3.9522e-05	4.7136e-07
F13	平均值	2.2131e-01	1.2547e-06	1.0774e-01	1.4075e-02	
	最优值	1.1710e-02	3.2145e-08	3.0485e-02	6.4133e-03	
	方差	3.6473e-02	5.2049e-13	5.3537e-03	7.5098e-05	
dim = 30	F8	平均值	-5.7873e+03	-1.1987e+04	-1.0852e+04	-8.1247e+03
		最优值	-6.0144e+03	-1.2569e+04	-1.2569e+04	-9.0087e+03
		方差	2.0403e+04	2.7193e+05	1.4750e+06	5.2660e+05
	F9	平均值	5.9653e+01	1.2547e-06	1.2547e+01	2.5874e+01
		最优值	3.6316e+01	5.2147e-08	0.0000e+00	0.0000e+00
		方差	1.0681e+02	7.6117e-13	7.6117e+01	1.0499e+02
	F10	平均值	1.3818e+01	1.2547e-12	8.6415e-15	1.0920e-09
		最优值	4.1920e+00	3.2147e-13	4.4409e-16	3.6500e-10
		方差	4.1042e+01	5.2049e-25	9.7561e-30	2.9340e-19
	F11	平均值	1.3230e+00	1.2547e-06	1.2125e-02	4.2158e-2
		最优值	1.1448e+00	5.2147e-08	0.0000e+00	3.3307e-16
		方差	1.7377e-02	5.2049e-13	1.4835e-03	1.2547e-03
	F12	平均值	1.1332e+00	1.2547e-05	1.4622e-02	1.6550e-03
		最优值	1.7323e-01	3.2145e-07	7.8934e-03	1.2687e-03
		方差	6.0366e-01	5.2049e-11	1.7380e-05	1.9929e-07
F13	平均值	1.3064e+01	2.1457e-04	2.6802e-01	4.8711e-02	
	最优值	2.4890e+00	1.2547e-06	9.1461e-02	2.0865e-02	
	方差	2.5082e+01	1.5518e-08	1.5528e-02	7.9914e-04	

由表 1 的测试结果分析可知, 相较于 NOA、IGWO 与 IWOA 算法, 本文提出的 INOA 算法在收敛性能与算法稳定性上均表现出更优的特性。

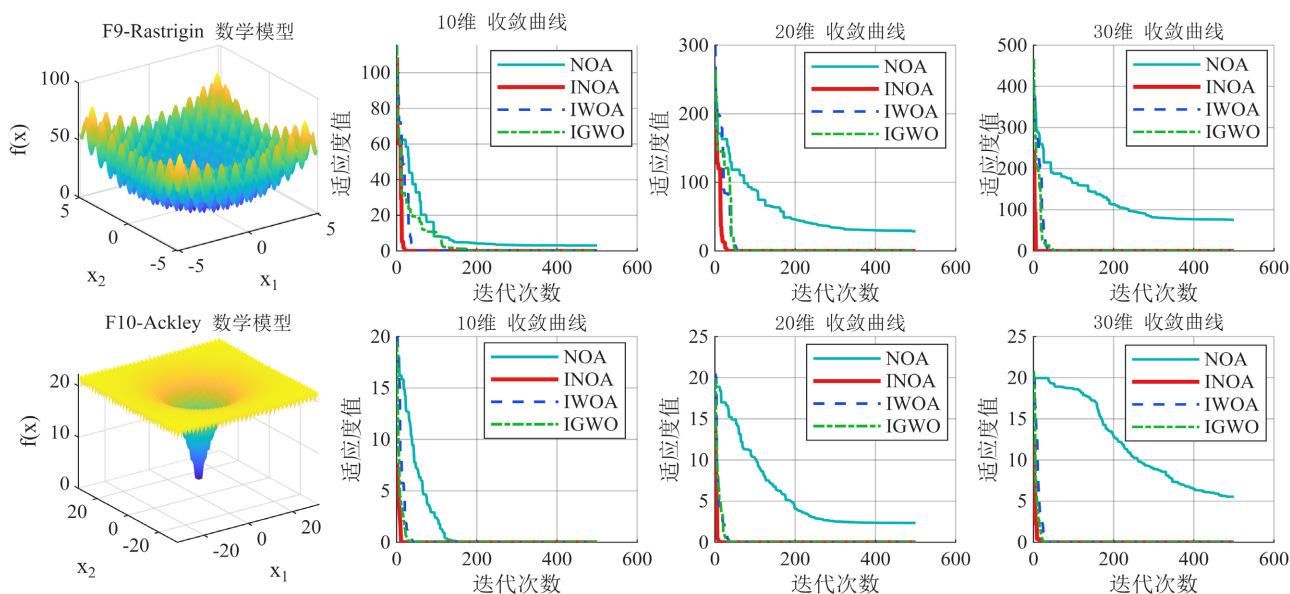


Figure 4. Convergence curves under different dimensions  
图 4. 不同维度下收敛曲线

### 3.2. 改进算法实验分析

为验证 INOA 算法的优化性能, 分别采用 INOA 算法对 FOPID 控制器与 DMC-FOPID 控制器参数进行迭代优化, 并将优化结果与 NOA 算法优化 FOPID 控制器参数的结果进行对比, 核心参数设置为: 种群规模 40, 最大迭代次数 100,  $K_p$ 、 $K_I$ 、 $K_D$  范围[0, 100],  $\lambda$ 、 $\mu$  范围(0, 2), 两种算法的适应度迭代曲线如图 5 所示。

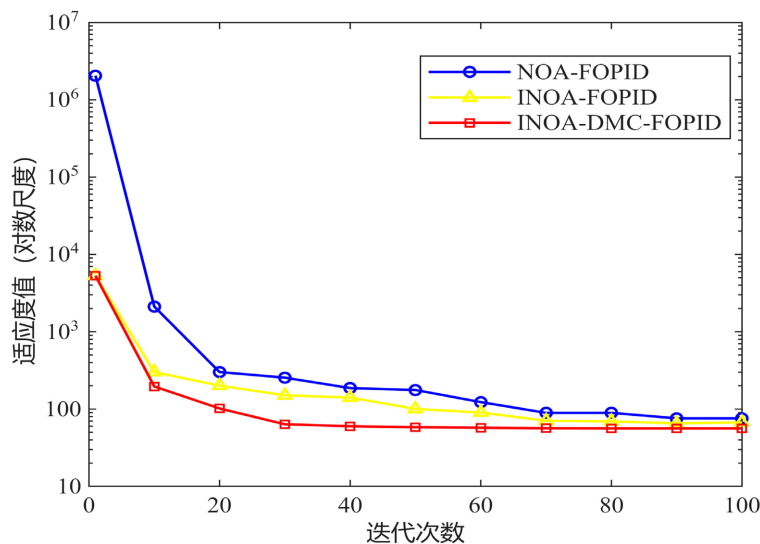


Figure 5. Fitness function value  
图 5. 适应度函数值

由图 5 可知, INOA 算法可搜索到更优的适应度值, 且 INOA 算法迭代至第 60 代即可达到收敛, 而 NOA 算法需迭代至第 80 代收敛, INOA 算法的收敛速度显著更优。

### 3.3. 脉冲响应分析

为量化检验 INOA 算法在熔深控制系统中的综合性能, 在 MATLAB/Simulink 仿真模型中开展实验, 将 NOA 算法与 INOA 算法迭代寻优得到的 FOPID 参数输入模型后, 与人工调试参数的 PID、FOPID 控制方法进行仿真对比, 设定期望熔深值为  $180 \text{ mm} \cdot e^{-2}$ , 不同控制方法的脉冲响应曲线如图 6 所示。

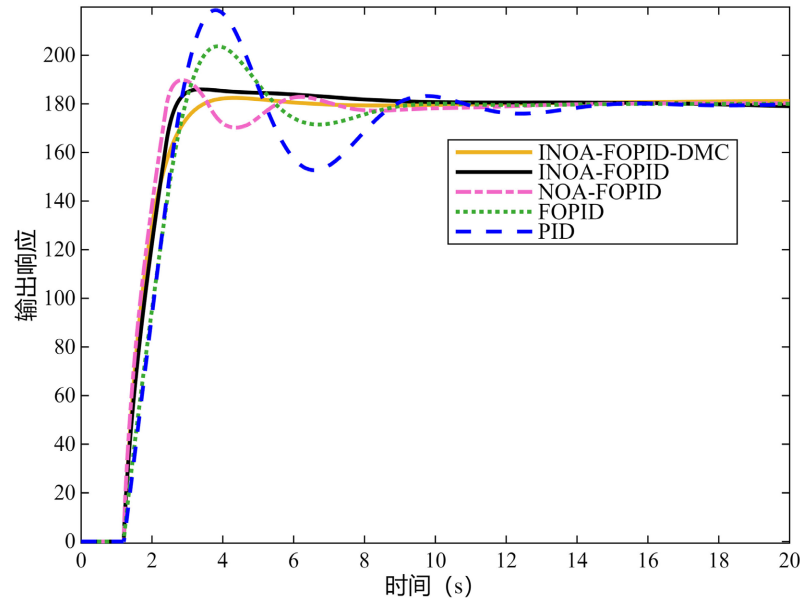


Figure 6. Output response comparison of different algorithms  
图 6. 各算法输出响应对比

阶跃响应数据如表 2 所示。

Table 2. System resulting data of standard experiment  
表 2. 标准试验系统结果数据

算法	调节时间(s)	超调量(%)	峰值时间(s)
INOA-FOPID-DMC	3.14	1.11	4.32
INOA-FOPID	4.51	3.21	3.39
NOA-FOPID	5.23	5.56	2.86
FOPID	8.15	13.2	3.86
PID	9.67	21.4	3.81

由结果可知, 相较于 NOA 算法, INOA 算法仅峰值时间有所增加, 其余各项指标均优于 NOA 算法; 与人工调试参数的 PID、FOPID 控制方法相比, INOA 算法在各项性能指标上均表现更优。

## 4. 结语

针对具有非线性、滞后性的熔深控制系统, 本文提出一种基于动态矩阵预测 DMC 与 INOA 算法优化 FOPID 参数的熔深控制策略。在同等仿真条件下, 将该策略与 PID、FOPID、NOA-FOPID、INOA-

FOPID 四种控制方法进行仿真对比试验。结果表明, 与对照组控制方法相比, INOA-DMC-FOPID 的综合控制性能更优秀, 可以有效降低系统超调量, 同时具备更强的抗干扰能力与鲁棒性, 能够使激光焊接过程中的熔深稳定满足实际生产需求。

## 参考文献

- [1] 宋天虎. 开创我国焊接行业的新局面[J]. 电焊机, 2020, 50(9): 1-10.
- [2] 焦俊科, 江桦锐, 周广兵, 等. PMMA 激光穿透焊接的实验研究[J]. 激光与光电子学进展, 2013, 50(5): 150-156.
- [3] El-Khazali, R. (2012) Fractional-Order  $PI^{\lambda}D^{\mu}$  Controller Design. *Program of the 5th Symposium on Fractional Differentiation and Its Applications*, Nanjing, 14-17 May 2012, 85-86.
- [4] Poli, R., Kennedy, J. and Blackwell, T. (2007) Particle Swarm Optimization. *Swarm Intelligence*, **1**, 33-57. <https://doi.org/10.1007/s11721-007-0002-0>
- [5] 张晓凤, 王秀英. 灰狼优化算法研究综述[J]. 计算机科学, 2019, 46(3): 30-38.
- [6] 许德刚, 王再庆, 郭奕欣, 等. 鲸鱼优化算法研究综述[J]. 计算机应用研究, 2023, 40(2): 328-336.
- [7] Abdel-Basset, M., Mohamed, R., Jameel, M. and Abouhawwash, M. (2023) Nutcracker Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Global Optimization and Engineering Design Problems. *Knowledge-Based Systems*, **262**, Article ID: 110248. <https://doi.org/10.1016/j.knosys.2022.110248>
- [8] Yin, F., Sun, J., Peng, W., Wang, H., Yang, J. and Zhang, D. (2017) Dynamic Matrix Predictive Control for a Hydraulic Looper System in Hot Strip Mills. *Journal of Central South University*, **24**, 1369-1378. <https://doi.org/10.1007/s11771-017-3541-6>
- [9] 王东莹, 孙法省. 计算机试验的设计理论和建模方法[J]. 曲阜师范大学学报(自然科学版), 2025, 51(4): 9-19+141.
- [10] 冯新强, 韦根原. 基于 ITAE 的时滞过程内模 PID 滤波器参数优化[J]. 电力科学与工程, 2015, 31(10): 40-43.