

doi: 10.7690/bgzdh.2020.11.012

基于 Henon 混沌映射的多目标粒子群算法改进分析

柴志君, 欧阳中辉, 李 钊

(海军航空大学岸防兵学院, 山东 烟台 264000)

摘要: 为解决多目标粒子群优化算法对初值敏感性较差和局部搜索能力不强的问题, 提出一种改进的混沌多目标粒子群优化算法。根据多目标优化问题的概念, 利用混沌对初值敏感、随机的特性, 确定种群初始值, 通过引入 Henon 混沌映射, 将改进后的算法应用到 3 个典型的多目标测试函数, 并与 NSGA II 和 MOPSO 算法进行对比。仿真结果表明: 改进的多目标粒子群算法在收敛性、分布性和均匀性等均有提高, 具有可行性和优越性。

关键词: 多目标优化; 粒子群算法; Henon 映射; Pareto 解; 评价指标

中图分类号: TP301.6 **文献标志码:** A

Improved MOPSO Optimization Based on Henon Chaotic Mapping

Chai Zhijun, Ouyang Zhonghui, Li Zhao

(School of Coast Defense, Navy Aviation University, Yantai 264001, China)

Abstract: In order to solve the problem of poor sensitivity to initial value and weak local search ability of multi-objective particle swarm optimization (MOPSO) algorithm, an improved chaotic multi-objective particle swarm optimization algorithm is proposed. According to the concept of multi-objective optimization problems, the chaos is sensitive to initial value, the characteristics of random initial values determine population, by introducing Henon chaotic map, the improved algorithm is applied to the three typical test functions, and compared with the NSGA II and MOPSO algorithm. The simulation results show that the improved multi-objective particle swarm optimization algorithm has better convergence, distribution and uniformity, and is feasible and superior.

Keywords: multi-objective optimization; particle swarm optimization; Henon map; Pareto solutions; evaluation index

0 引言

粒子群优化算法具有形式简洁、收敛快速和参数调节机制灵活等优点, 同时一次运行可得到多个解, 且能逼近非凸或不连续的 Pareto 最优前端, 因而被认为是求解多目标优化问题最具潜力的方法之一。当粒子群优化算法从单目标问题扩展到多目标问题时, Pareto 最优解集的存储与维护、全局和个体最优解的选择以及开发与开采的平衡等问题亦随之出现。因此, 很多学者都致力于改善 MOPSO (multi objective particle swarm optimization) 算法的性能, 如采用惯性权重法、压缩因子法、混合法、空间邻域法、社会趋同法、动态目标函数法、协同法、结合复杂系统的自组织临界性等^[1]。这些改进措施主要是从改善算法本身着手, 每种改进方法只是从某一方面提高了 MOPSO 算法的优化性能, 而笔者构建的 MOPSO 算法融入 Henon 混沌映射的优化方法, 在提高 MOPSO 解的精度以及提高整个算法的优化求解效率时, 并不着手改善 MOPSO 自身算法的参数设置, 而是在整个 MOPSO 优化迭代过程中,

将优化变量空间转化到混沌空间, 利用混沌具有遍历性等特点获得优化变量空间中的一个优化值。

1 相关概念

1.1 多目标优化问题相关概念

在生产管理和国防建设过程中, 解决遇到的问题往往要考虑多个目标, 并在给定的区间尽可能达到最优, 即多目标最优化问题^[2-3]。

多目标优化问题 (multi-objective optimization problem, MOP) 一般可表述^[4]为:

$$\left. \begin{array}{l} \min \quad \{f_1(X), f_2(X), \dots, f_i(X), \dots, f_N(X)\} \\ \text{s.t.} \quad g_j(X) \leq 0, \quad h_j(X) = 0 \\ \quad \quad j = 1, 2, \dots, M \\ \quad \quad X_L \leq X \leq X_U \end{array} \right\} \quad (1)$$

式中: X 为优化的设计变量; $f(X)$ 为目标函数; $g_j(X)$ 和 $h_j(X)$ 为约束函数; N 和 M 分别为目标函数和约束函数的数量; X_L 和 X_U 分别为设计变量的下限和上限。

多目标优化问题因各目标相互之间存在制约和冲突, 无法像单目标优化问题一样求得唯一的最优

收稿日期: 2020-07-02; 修回日期: 2020-08-20

作者简介: 柴志君(1989—), 男, 山东人, 硕士, 从事军用仿真技术研究。E-mail: 275131218@qq.com。

解，而是得到 Pareto 最优解集或最优非劣 (non-dominated) 解集^[5]。

1.2 粒子群优化算法

粒子群优化 (particle swarm optimization, PSO) 算法是一种源于对鸟类捕食行为研究的进化计算技术。在 PSO 模型中，个体 (称为粒子) 会根据自身经验和群体经验调整运动状态，表现出一定的智能性。每个粒子的位置代表函数的一个解，且任意粒子都用一个速度来决定它的飞行方向和移动位移。在每次迭代过程中，粒子通过跟踪局部极值 p_{best} 和全局极值 g_{best} 2 个极值对自身位置进行更新。

Kennedy 和 Eberhart 提出对粒子操作的公式^[6]如下：

$$\left. \begin{aligned} v_{is}(t+1) &= v_{is}(t) + c_1 r_{1s}(t)(p_{is}(t) - x_{is}(t)) + \\ &\quad c_2 r_{2s}(t)(p_{gs}(t) - x_{is}(t)) \\ x_{is}(t+1) &= x_{is}(t) + v_{is}(t+1) \end{aligned} \right\} \quad (2)$$

式中， c_1 和 c_2 均为学习因子，并分别负责调节粒子向 p_{best} 和 g_{best} 的学习步长。2 个学习因子均为非负常数，一般设 $c_1=c_2=2$ ； r_{1s} 和 r_{2s} 都是 $[0,1]$ 之间的随机数。

1.3 混沌序列

混沌理论^[7] (chaos theory) 是一种非线性理论，具有非线性、初值敏感性、随机性和遍历性等特点，能按其自身的规律在一定范围内不重复地遍历所有状态。Henon 映射在 2 维空间产生，是一种典型的离散混沌映射，其动力学公式如下：

$$\left. \begin{aligned} x_{n+1} &= 1 + y_n - ax_n^2 \\ y_{n+1} &= bx_n \end{aligned} \right\} \quad (3)$$

从上式可知， x_0, y_0, a 和 b 4 个参数决定 Henon 映射的状态，较 1 维混沌映射更复杂。研究结果表明：当 $a=1.4, b=0.3$ 时，函数进入混沌状态，生成的混沌序列具有很强的随机性^[8]。

2 基于 Henon 映射的多目标粒子群算法

多目标粒子群初值敏感性强。在决策空间内随机生成的初始种群时，传统的 MOPSO 虽然具备一定的随机性，但分布较为散乱。为均匀地在全部决策空间内产生初始种群，并增加其随机性，在种群初始值的确定中引入 Henon 混沌映射^[9]。

引入 Henon 混沌映射的多目标粒子群算法 (Henon chaotic MOPSO, H-CMOPSO) 在保证种群多样性的前提下，能够提高收敛速度，极大地缩短其寻优时间。算法具体步骤如下：

1) 初始化。根据均匀概率随机取值，在决策空间中得出所有粒子的位置，粒子的速度均取 0。令粒子的个体向导为该粒子的位置，档案集为空。

2) 种群评价。求得种群粒子位置在目标函数 Q_1, Q_2, \dots, Q_M 上的取值，根据取值范围，将其平均分割成 n 块，共计 $10n$ 个网格。

3) 粒子的个体向导更新。在种群粒子的目标值向量支配其个体向导或两者互不支配的情况下，均将个体向导更新为粒子当前位置；否，则不予更新。

4) 快速非支配解集构造。删掉种群中不满足约束 $Q_1 \leq Q_1^{\max}$ 或 $Q_M \geq Q_M^{\max}$ 的粒子后，与档案集合并。

5) 用非支配集 ND 更新档案集。当档案集数大于规模数 (这里为 $10n$) 时，则根据档案集粒子的网格密度，由大到小依次删除至档案规模不超过 $10n$ 。

6) 当算法迭代次数大于 300 时，停止搜索并将档案集输出；次数不到 300 时，转 7)。

7) 根据式(5)，进行混沌扰动搜索，生成新的多样化且分布均匀的种群替换原种群。

8) 将网格密度最小的粒子记做全局向导，在有多个的情况下，随机选择其中一个。然后转 2)。

3 实验结果与分析

3.1 评价方法

多目标优化算法的性能评价通常涉及 2 个方面：一方面获得分布尽可能广的 Pareto 前端，另一方面是获得尽可能多的 Pareto 解。笔者选用以下 5 个量化评价标准，在很大程度上可以反映多目标优化算法的性能^[10-12]。

1) 容错率评价。

Veldhuizen 提出的容错率 (error rate, ER) 用来检验解不在 Pareto 前沿上的占比，定义如下：

$$ER = \sum_{i=1}^n P_i / n, \quad P_i = \begin{cases} 0 & \text{解在 Pareto 上} \\ 1 & \text{解不在 Pareto 上} \end{cases} \quad (4)$$

ER 值为 0，说明得出的所有非劣解都分布在 Pareto 前沿上。ER 的值越小，说明求得的解集越好。

2) 收敛性评价。

Veldhuizen 和 Lamont 提出的一种世代距离 (generational distance, GD) 用来估算算法得到的非劣解集与 Pareto 前沿的距离，其形式如下：

$$GD = \sqrt{\sum_{i=1}^n d_i^2} / n \quad (5)$$

式中： n 为所获得非劣解的个数； d_i 为第 i 个非劣解到已知 Pareto 最优解集的最小距离。GD 的值越小，

表明算法的收敛性越好。

3) 分布性评价。

Deb 等提出了分布性评价指标 (diversity measure, DM), 可以对非劣解的分布性广泛程度进行评估, DM 的计算公式如下:

$$DM = \frac{d_f + d_1 + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_1 + (n-1)\bar{d}} \quad (6)$$

式中: d_f 为极值点到非劣解的距离; d_1 为边界点到非劣解的距离。如果 $d_f = d_1 = 0$, $d_i = \bar{d}$, 说明算法求得的解完全覆盖在真实 Pareto 前沿面上, 是一种较为理想的状况。

4) 均匀性评价。

采用非劣解间距度量 (spacing, SP) 对解集分布的均匀程度进行评估, 其计算公式如下:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (7)$$

式中: n 为获得的非劣解个数; $d_i = \min(\sum_{k=1}^m |f_k(x_i) - f_k(x_j)|)$ (其中, $i, j = 1, \dots, n, i \neq j, k$ 为目标函数的个数); d_i 为相邻最近非劣解对应的目标向量之间的距离; \bar{d} 为 d_i 的平均值。

5) CPU Time 评价: 算法运行时间, s 。

3.2 测试函数及实验结果

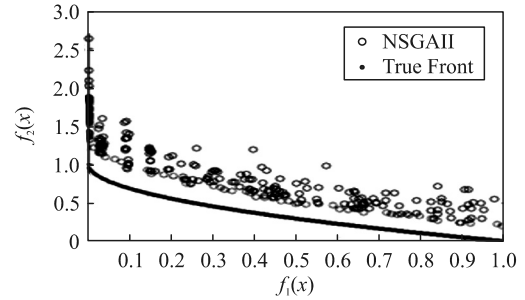
从 Zitzler 和 Deb 提出的标准函数中选择 3 个典型的测试函数, 对笔者提出的 H-CMOPSO 算法同一般 MOPSO 算法和第二代非劣解排序遗传算法 (NSGA II) 进行对比测试。各种算法的群体规模均为 200, 最大进化代数均为 150, H-CMOPSO 算法 Henon 映射 $a=1.4, b=0.3$; MOPSO 算法的学习因子 $c_1 = c_2 = 0.5$, 惯性因子 $\omega = 0.1 + \text{rand}(\omega_0 = 0.1)$, 且 $0.1 \leq \omega \leq 0.8$; NSGA II 算法的变异率 $p_m = 0.1$, 灾变率 $p_z = 0.1$, 交叉率 $p_c = 0.7$ 。

1) 选择 ZDT1 作为第 1 个测试函数, 表达式为:

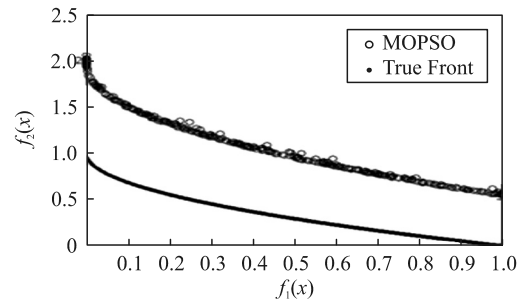
$$\left. \begin{aligned} f_1(x) &= x_1 \\ f_2(x) &= g(x)[1 - \sqrt{x_1/g(x)}] \\ g(x) &= 1 + 9 \left(\sum_{i=2}^n x_i / (n-1) \right) \\ x_1 &\in [0, 1], x_i = 0, i = 2, 3, \dots, n \end{aligned} \right\} \quad (8)$$

该函数是一个凸函数, n 取 30, 分别运行 MOPSO、NSGA II 和 H-CMOPSO 3 种算法 150 代,

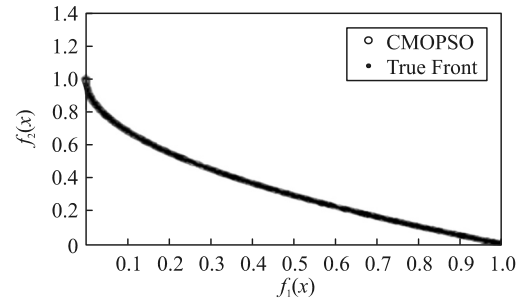
可得 3 种算法产生的 Pareto 前沿如图 1 所示, 几种评价指标的计算测试结果如表 1 所示。



(a) NSGA II 产生的 Pareto 前沿



(b) MOPSO 产生的 Pareto 前沿



(c) H-CMOPSO 产生的 Pareto 前沿

图 1 3 种算法产生的 Pareto 前沿

表 1 对 ZDT1 的测试结果对比

算法	ER	GD	DM	SP	T
MOPSO	1.00	0.048 966	0.476 225	0.236 922	10.760 340
NSGA II	0.92	0.032 369	0.419 082	0.203 561	9.573 628
H-CMOPSO	0	0.000 169	0.380 520	0.189 341	12.277 200

从图中可以看出: NSGA II 算法获得的非劣解数量少, 且距离真实 Pareto Front 较远; MOPSO 算法虽然获得非劣解数量比 NSGA II 算法多, 但距离真实 Pareto Front 更远; H-CMOPSO 算法获得的非劣解数量多, 且非劣解的前沿分布比较均匀。

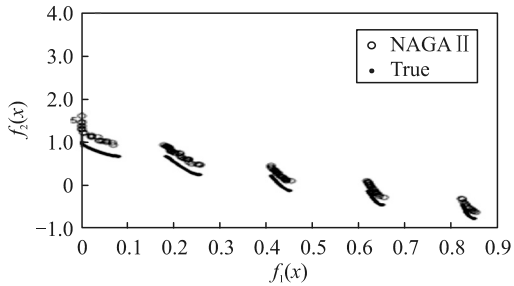
从表中可以看出: 笔者提出的算法对 ZDT1 测试函数都有很好的性能度量, H-CMOPSO 算法的 ER 值最低, 说明该算法所得的非劣解在近似 Pareto 最优解集中所占的比重最大; MOPSO 算法的 ER 值为 1, 说明该算法所得的非劣解不在近似 Pareto 最优解集中; H-CMOPSO 的 GD、DM 和 SP 数值均小于其他 2 种算法, 说明该算法的收敛性、分布性

和均匀性都有非常好的效果。从 CPU Time 可以看出，在运算速度方面，其他 2 种算法由于 H-CMOPSO。

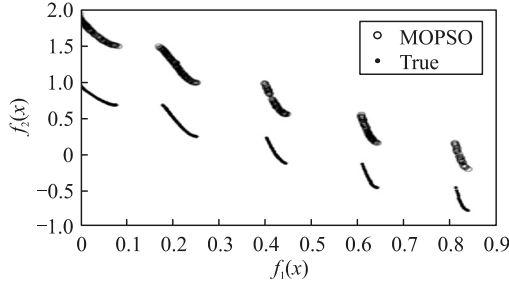
2) 选择 ZDT3 作为第 2 个测试函数，表达式为：

$$\left. \begin{aligned} f_1(x) &= x_1 \\ f_2(x) &= [1 - \sqrt{x_1/g(x)} - (x_1/g(x))\sin(10\pi x_1)] \\ g(x) &= 1 + 9 \sum_{i=2}^n x_i / (n-1) \\ x_i &\in [0,1], x_i = 0, i = 2,3,\dots,n \end{aligned} \right\} \quad (9)$$

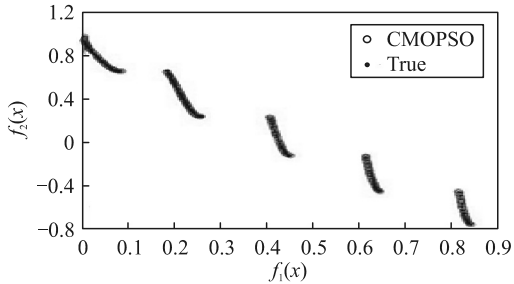
该函数是不连续、凸函数， n 取 30，分别运行 MOPSO、NSGA II 和 H-CMOPSO 3 种算法 150 代，可得 3 种算法产生的 Pareto 前沿如图 2 所示，几种评价指标的计算测试结果如表 2 所示。对图 2 和表 2 进行分析，可得出与 ZDT1 测试基本相同的结果。



(a) NSGA II 产生的 Pareto 前沿



(b) MOPSO 产生的 Pareto 前沿



(c) H-CMOPSO 产生的 Pareto 前沿

图 2 3 种算法产生的 Pareto 前沿

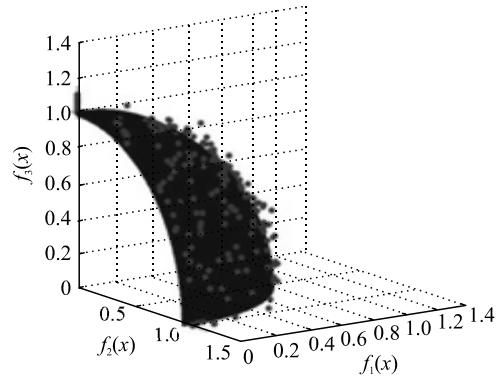
表 2 对 ZDT3 的测试结果对比

算法	ER	GD	DM	SP	T
MOPSO	1.00	0.060 545	0.562 611	0.279 269	10.262 350
NSGA II	0.69	0.034 652	0.490 423	0.239 709	9.432 923
H-CMOPSO	0	0.000 404	0.400 627	0.199 200	12.786 300

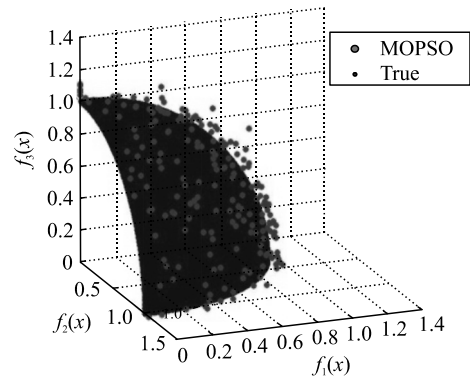
3) 选 DZLT2 作为第 3 个测试函数，表达式为：

$$\left. \begin{aligned} f_1(x) &= \cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)(1+g(x)) \\ f_2(x) &= \cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)(1+g(x)) \\ f_3(x) &= \sin\left(\frac{\pi}{2}x_1\right)(1+g(x)) \\ g(x) &= \sum_{i=3}^m (x_i - 0.5)^2 \\ \text{s.t. } &0 \leq x_i \leq 1, i = 1,2,\dots,n \end{aligned} \right\} \quad (10)$$

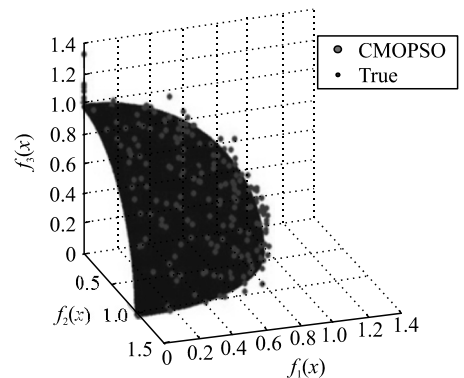
该函数是凸函数， n 取 12，分别运行 MOPSO、NSGA II 和 H-CMOPSO 3 种算法 150 代，可得 3 种算法产生的 Pareto 前沿如图 3 所示，几种评价指标的计算测试结果如表 3 所示。对图 3 和表 3 进行分析，同样得出与 ZDT1 测试基本相同的结果。



(a) NSGA II 产生的 Pareto 前沿



(b) MOPSO 产生的 Pareto 前沿



(c) H-CMOPSO 产生的 Pareto 前沿

图 3 3 种算法产生的 Pareto 前沿

表 3 对 DZLT2 的测试结果对比

算法	ER	GD	DM	SP	T
MOPSO	0.35	0.008 359	0.494 531	0.136 053	10.443 16
NSGA II	0.15	0.011 414	0.554 686	0.136 613	17.535 96
H-CMOPSO	0.12	0.005 460	0.449 102	0.118 802	12.188 59

4 结束语

通过在种群初始值的确定中引入混沌映射，提高 MOPSO 种群的多样性，笔者提出了一种基于 Henon 混沌映射的改进 MOPSO 算法。该算法不仅具有良好的全局优化性能，能够避免陷入局部最优；而且外部存储库能够帮助粒子提供更多的解空间，避免产生大量相同非劣解，收敛性好，非劣解绝大多数都分布在 Pareto 前沿上，维持了决策的多样性，降低了解的错误率。该算法已应用于 3 个典型的多目标测试函数中，与 NSGA II 和 MOPSO 相比，效果更好。

参考文献：

[1] 高飞, 童恒庆. 基于改进粒子群优化算法的混沌系统参数估计方法[J]. 物理学报, 2006, 55(2): 577-582.
 [2] SHI Y, EBERHART R. Parameter selection in particle swarm optimization[C]//Proceedings of the 7th Annual Conference on Evolutionary Programming, Washington DC, 1998: 591-600.
 [3] 杨培宏, 刘连光, 刘春明, 等. 基于粒子群优化算法的电网 GICQ 多目标优化策略[J]. 电力自动化设备, 2017,

37(3): 93-99.
 [4] LORENZ E N. Deterministic nonperiodic flow[J]. Journal of the Atmospheric Sciences, 1963, 20(2): 130-141.
 [5] KENNEDY J, EBERHART R C. Particle Swarm Optimization[C]. Proc. IEEE International Conference on Neural Networks, Piscataway, NJ: IEEE Service Center, 1995: 1942-1944.
 [6] 张国平, 王正欧, 袁国林. 求解一类组合优化问题的混沌搜索法[J]. 系统工程理论与实践, 2001, 21(5): 102-105.
 [7] 管春阳, 高飞. 一种基于混沌序列的加密算法[J]. 北京理工大学学报, 2003, 23(3): 363-366.
 [8] XU H L, QIAN X, ZHANG L. Study of ACO Algorithm Optimization Algorithm Based on Improved Tent Chaotic Mapping[J]. Journal of Information and Computational Science, 2012 9(6): 1653-1660.
 [9] 吴文欢, 张少辉, 李巍, 等. 分阶段进化的粒子群优化算法[J]. 重庆理工大学学报(自然科学版), 2012(6): 67-70.
 [10] ADIL B. Design optimization with chaos embedded great deluge algorithm[J]. Applied Soft Computing, 2012(12): 1055-1067.
 [11] 黄波, 张林华, 段婉玉. 一种基于 Henon 映射的图像加密算法[J]. 重庆工商大学学报(自然科学版), 2013, 30(11): 62-65.
 [12] 王洪刚, 马良, 李高雅. 多目标微粒群优化算法[J]. 计算机工程与应用, 2008, 44(34): 64-66.

(上接第 31 页)

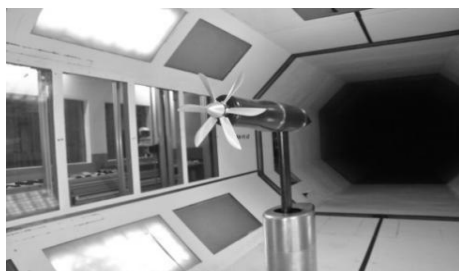


图 8 单桨试验

6 结论

通过对单机螺旋桨带动力试验系统的研制，充分利用现有设备，以极低成本增加了零拉力控制。运行结果表明：系统控制精度高、电磁兼容性好、自动化程度高，显著增强了系统性能，提高了试验能力及试验效率，并通过对系统功能的拓展，成功应用于其他旋转试验中，具有较强的工程价值，试验准备时间由 48 h 缩短至 12 h，大幅提高了单机螺旋桨带动力试验效率。

参考文献：

[1] 王勋年. 低速风洞试验[M]. 北京: 国防工业出版社, 2002: 28-29.
 [2] 霍国, 康洪铭, 刘晓林, 等. FL-12 风洞螺旋桨带动力试验系统[J]. 兵工自动化, 2018, 37(6): 23-27.
 [3] 邓磊, 段卓毅, 钱瑞战, 等. 螺旋桨滑流对短舱/机翼构型尾迹流场的影响[J]. 航空学报, 2019, 40(5): 1-11.
 [4] 霍国, 王政, 刘晓林, 等. 单机螺旋桨带动力控制系统改进实施方案[Z]. 绵阳, 2019: 1-3.
 [5] CT. Unidrive SP 1-9 型感应电机及伺服电机通用交流变频器[S]. 2005.
 [6] 霍国, 康洪铭, 李涵, 等. 风洞螺旋桨试验电机现状及发展方向[J]. 兵工自动化, 2016, 35(11): 20-23.
 [7] 叶松波, 王政. 零拉力控制在螺旋桨带动力试验中的应用[Z]. 绵阳, 2005: 5-7.
 [8] 焦悦, 商亚丹, 董波, 等. 能耗制动中制动电阻的分析计算[J]. 重型机械, 2013(2): 81-84.
 [9] 霍国. 4 米×3 米风洞螺旋桨带动力控制系统研制[D]. 绵阳: 西南科技大学, 2017.