一种基于GOA的多推进器系统水下航行器容错轨迹跟踪控制，且不考虑执行器饱和

Danjie Zhu，*IEEE*会员，Lei Wang、Hua Zhang和Simon X. Yang，*IEEE*高级会员

arXiv:2301.01827v1 [cs.RO] 4 Jan 2023

摘要—本文提出一种智能容错控制（FTC）策略，用于解决水下航行器（UV）在推进器损坏（功率损失）情况下的轨迹跟踪问题，同时解决航行器物理约束导致的执行器饱和问题。在所提出的控制策略中，轨迹跟踪分量由改进反步算法和滑模控制构成，前者控制速度变化，后者计算力矩/力输出；容错分量则基于蚱蜢优化算法（GOA）构建。该算法收敛速度快、精度令人满意，可计算优化后的推进器推力重新分配方案，以补偿不同故障情况下的功率损失。本文展示了不同故障情况下有无环境扰动的仿真结果，并与其他传统容错控制方法进行比较，从而验证了所提出的基于GOA的容错轨迹跟踪设计的有效性和鲁棒性。

给从业者的说明：本文的研究动机是，当水下航行器（UV）的推进器系统发生功率损失时，其轨迹跟踪过程中会出现执行器饱和问题。采用传统方法时，航行器的物理约束会影响容错轨迹跟踪性能，因为这些方法在控制过程中可能会提出过高的运动学/动力学要求，从而导致跟踪轨迹偏离。因此，本文结合改进的反步法与蚱蜢优化（GOA）来消除过度要求，其中改进的反步法用于缓解速度跳变（运动学输出），而面对推进器故障情况时，GOA 用于控制推进力（动态输出）。这项研究为行业从业者提供了新的思路：通过嵌入优化算法改进航行器的控制设计，能够避免故障情况下因驱动指令无法满足要求而造成的跟踪偏差。此外，针对本文研究的用于大坝检测的特定类型水下航行器，本文选取了贴近实际大坝检测的仿真场景，例如三维折线轨迹跟踪以及频繁出现的水下航行器单故障情况，可为相关领域的从业者提供参考。未来将开展该水下航行器的水下实验，并纳入更多实际环境因素。

索引词——**执行器饱和、反步控制、容错控制、蚱蜢优化、轨迹跟踪、水下航行器。**

I. 引言

THE 航行器控制研究已扩展至各种条件，在这些条件下，容错被广泛纳入相应的控制设计，这称为容错控制（FTC） [1, 2]。数十年来，科学家一直在航空航天器、陆地车辆和工业制造等多个领域开展容错控制研究 [3–10]。以往研究中，容错控制通常用于减轻突发误差，并在不同领域的设备发生不可避免的损坏时提供最可行的解决方案 [11]。然而，由于水下环境和水下航行器系统带来的复杂性，针对水下航行器（UV）的容错控制研究尚未得到充分探索 [12–15]。

本世纪以来，已有相关研究提出了容错控制（FTC）方法 [3, 4, 6, 16]。基于这些研究，有人提出并认可了一种解决水下航行器容错控制问题的方法：配置数量超过自由度（DOF）数量的推进器，这称为推进器重构 [17, 18]。例如，当航行器推进器发生意外故障时，安装在航行器上的推进器数量超过其自由度数量（六个：纵荡、横荡、垂荡、横滚、俯仰和偏航），因而具有足够的调节余地，可在相应自由度上提供所需推力。为在实际情况下应用推进器配置理论，研究者提出了加权伪逆矩阵法，将故障情况量化为损伤程度，并作为输入构建推进器配置模型 [19]。采用该方法，容错控制过程大大简化，因为可以通过加权伪逆矩阵模型直接求得所需的推进器推力。然而，推进器输出的物理约束很少得到考虑，因而引发了过驱动车辆问题 [20, 21]。此外，这些研究大多致力于消除故障情况引起的静态误差。然而，在水下航行器的实际应用中，实时实现对航行器输出的动力学控制至关重要；这通常指水下航行器的轨迹跟踪控制 [22–24]。

因此，为解决过驱动问题，同时实现水下航行器轨迹跟踪的鲁棒性，我们将优化方法与跟踪控制相结合，在跟踪过程中遇到故障情况时，将航行器的动态输出优化至允许范围内。在当前主流优化算法中，遗传算法的迭代耗时较长，

本研究得到自然科学与工程研究理事会（NSERC）及中国国家重点研发计划项目的支持。Danjie Zhu 和 Simon X. Yang 任职于圭尔夫大学工程学院先进机器人与智能系统（ARIS）实验室，地址为加拿大安大略省圭尔夫市 N1G 2W1（电子邮箱：{danjie；syang}@uoguelph.ca）；Lei Wang 和 Hua Zhang 任职于中国船舶科学研究中心水下工程研究所，地址为中国无锡市 214082（电子邮箱：13771115826@163.com；zhanghua702@126.com）。

这不适用于既要求快速又要求可行解的水下航行器容错控制 [25–27]。神经网络对数据输入的选择要求较高，而水下航行器在遇到故障情况时无法提供所需数据；贪心算法也存在类似问题，因为它需要先分解数据再进行处理 [28–30]。因此，群智能算法凭借灵活的数据输入方式和快速的收敛速度，成为解决水下航行器容错控制应用问题的更优选择 [31–33]。Zhu 团队已将基于粒子群优化（PSO）的容错控制应用于无人水下航行器。尽管该方法获得了令人满意的力矩输出，但传统 PSO 方法的实时反馈能力较差，不符合水下航行器容错控制的在线要求 [34, 35]。本研究选择了一种名为蚱蜢优化算法（GOA）的先进群智能方法，因为它能够在快速收敛与优化结果精度之间取得令人满意的平衡，且易于实现 [36]。该算法通过依据各搜索代理的位置简单更新迭代，实现快速收敛，大幅提升优化效率，并为水下航行器容错控制提供实时反馈的可能性 [37, 38]。与此同时，GOA 通过优化算法中嵌入的限制条件，能够在可接受的驱动约束范围内提供准确的容错结果，从而灵活地解决水下航行器容错控制中的过驱动问题 [39, 40]。

本文提出的控制策略将先进的群智能算法（GOA）与容错轨迹跟踪控制相结合，以解决先进水下航行器在无执行器饱和情况下的故障问题。通过识别并量化多推进器系统的损坏程度（功率损失），再利用GOA方法高效地重新分配推力，可快速收敛并获得精度令人满意的可行解。该策略为遭受突发损坏的水下航行器建立了系统的故障识别和高效误差消除流程；同时，它首次实现了GOA方法在特定多推进器系统水下航行器容错控制中的应用。Durham提出的约束控制分配方法等传统方法基于线性代数的基本概念，并通过确定可达力矩空间的边界表面来进行分配，但该边界表面难以处理 [41]。常用于构建水下航行器容错控制的方法，如T-近似或S-近似，无法彻底解决执行器饱和问题，因为航行器约束不可避免地会带来误差 [34]。因此，GOA方法快速收敛且精度理想，有助于及时修正动态控制器发出的指令，从而实现水下航行器轨迹跟踪的实时容错控制[36]。

本文其余部分安排如下。首先，介绍先进水下航行器“YuLong”的运动学与动力学模型，并基于无人水下载具推进器配置定义力矩-力转换与归一化。随后，阐述水下航行器的容错轨迹跟踪问题，并说明其约束条件。接着，提出基于蚱蜢优化的容错轨迹跟踪控制，其中采用改进反步算法构成运动学控制分量；采用滑模控制作为动力学控制分量；并利用蚱蜢优化算法，通过重新配置推进器力输出，消除不同故障情况下推进器不同损伤程度所产生的误差，从而构成容错分量。在最后一节中，通过跟踪期望的折线或螺旋轨迹，对所提出的基于蚱蜢优化的容错控制的有效性进行评估，条件是假设一个推进器（单故障）或两个推进器（双故障）发生损坏。

II. 水下航行器模型与问题描述

本节研究一种名为“YuLong”的典型水下航行器，并以具体方程的形式给出其机器人模型及轨迹跟踪问题的描述。

A. “玉龙”无人艇模型

本节研究一种名为“YuLong”的典型水下航行器，并以具体方程的形式给出其机器人模型及容错轨迹跟踪问题的描述。

1）运动学与动力学模型：“玉龙”无人艇是中国船舶科学研究中心水下工程研究所设计的最新型大坝检测水下航行器之一，潜深可达3000米。其示意图如图1所示，多推进器系统结构如图2所示。作为专为检测深水区域大坝并确保其处于良好状态而设计的水下航行器，其运行的鲁棒性和效率对大坝检测设备至关重要。

水下航行器的六个自由度（DOF）包括纵荡、横荡、垂荡、横摇、俯仰和偏航。由于在实际导航过程中，横摇和俯仰这两个自由度对水下航行器的影响极小，因此可以忽略（图1）。因此，为了建立轨迹跟踪模型并确保水下航行器可控运行，通常只考虑纵荡、横荡、垂荡和偏航这四个自由度。基于这四个自由度，对于水下航行器的运动学方程，速度向量$\mathbf{v}$ 可转换为轨迹向量 $\dot{\mathbf{p}}$的时间导数，即

$$\dot{\mathbf{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \mathbf{J}(\mathbf{p})\mathbf{v} == \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ r \end{bmatrix}, \tag{1}$$

其中，$\mathbf{J}$ 是根据水下航行器艇体的物理结构推导出的变换矩阵，$[u\ v\ w\ r]^T$ 表示水下航行器所选四个坐标轴上的速度（见图1）。

在实际的水下航行器系统中，水动力阻力、阻尼、升力、科氏力和向心力、重力和浮力、推进器作用力以及环境扰动等多种复杂的非线性力会作用于

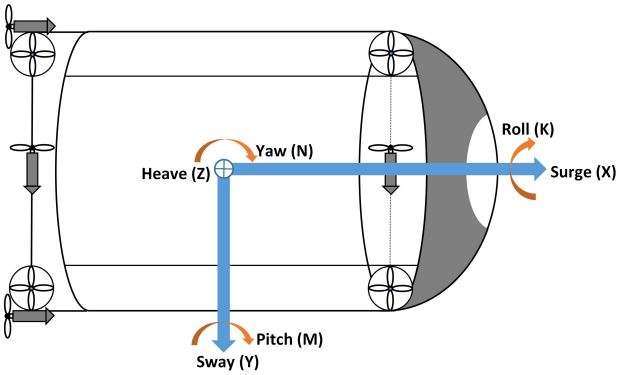


图1.“玉龙”无人艇的参考坐标系和六个自由度（x、y、z、k、m 和 n）（俯视图）。

航行器。考虑到这些力的来源及其影响，可将一般动力学模型写为

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{g}(\mathbf{p}) = \boldsymbol{\tau}, \tag{2}$$

其中，M为刚体与附加质量之和的惯性矩阵；**C(v)**为刚体与附加质量之和的科里奥利与向心矩阵；**D(v)**为二次与线性阻力矩阵；**g(p)**为重力与浮力矩阵；而 $\boldsymbol{\tau}$ 为推进器输入的力矩向量。如前一节所述，本研究仅针对特定模型 YuLong 水下航行器考虑四个状态。推进器输入的力矩向量表示为

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_x & \tau_y & \tau_z & \tau_n \end{bmatrix}, \tag{3}$$

其中，x、y 和 z 分别表示水下航行器在纵荡、横荡和垂荡方向上的线位移，而 n 表示水下航行器在偏航方向上的角位移（见图1）。

对于“玉龙”无人艇，指定以下参数值：惯性矩阵$\mathbf{M} = [42\ \ 0\ \ 0\ \ 0;\ \mathbf{0\ 153\ 0\ 0};\ \mathbf{0\ 0\ 141\ 0};\ \ 0\ \ 0\ \ 0\ \ 100]^T$；科里奥利与向心矩阵$\mathbf{C(v)} = \mathbf{J}^{-1}\mathbf{M}\dot{\mathbf{J}}\mathbf{J}^{-1}$，其中$\mathbf{J}^{-1}$表示**J**的逆矩阵，而 $\dot{\mathbf{J}}$表示**J** 的导数；二次与线性阻力矩阵$\mathbf{D(v)} = [42+69u\ \ 0\ \ 0\ \ 0;\ \ 0\ \ 319+245v\ \ 0\ \ 0;\ \ 0\ \ 0\ \ 272+86w\ \ 0;\ 0\ \ 0\ \ 0\ \ 33+4r]^T$。此外，当整个系统处于平衡状态时，作用于航行器上的引力与浮力相互抵消。

2）力矩-力转换与归一化：根据“玉龙”航行器推进系统的物理结构（见图2），其力矩向量与推进器推力之间的关系为

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \\ \tau_n \end{bmatrix} = \begin{bmatrix} T_1 + T_2 \\ T_7 + T_8 \\ T_3 + T_4 + T_5 + T_6 \\ T_1 + 1.4 \times T_8 - T_2 - 1.4 \times T_7 \end{bmatrix}, \tag{4}$$

其中，T_1、T_2、T_3、T_4、T_5、T_6、T_7、T_8 是安装在航行器载体本体周围的八个推进器产生的力。

这八个推进器属于同一类型，且假设其最大力 T_m相同。因此，根据式（4）可以推导出力矩向量 $\boldsymbol{\tau_m}$ 的最大值：

$$\boldsymbol{\tau_m} = \begin{bmatrix} \tau_{xm} \\ \tau_{ym} \\ \tau_{zm} \\ \tau_{nm} \end{bmatrix} = \begin{bmatrix} 2T_m \\ 2T_m \\ 4T_m \\ (2+2.8)T_m \end{bmatrix}. \tag{5}$$

基于式（4），将式（5）的两边除以最大力矩，以将输出限制在 -1 到 1 的范围内，并令 $\boldsymbol{\tau} = \boldsymbol{\tau}/\boldsymbol{\tau_m}$，$\mathbf{T} = \mathbf{T}/\mathbf{T_m}$，该向量变换为

$$\begin{bmatrix} \overline{\tau}_x \\ \overline{\tau}_y \\ \overline{\tau}_z \\ \overline{\tau}_n \end{bmatrix} = \begin{bmatrix} \tau_x/\tau_{xm} \\ \tau_y/\tau_{ym} \\ \tau_z/\tau_{zm} \\ \tau_n/\tau_{nm} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{5}{24} & -\frac{5}{24} & 0 & 0 & 0 & 0 & -\frac{7}{24} & \frac{7}{24} \end{bmatrix} \begin{bmatrix} T_1/T_m \\ T_2/T_m \\ T_3/T_m \\ T_4/T_m \\ T_5/T_m \\ T_6/T_m \\ T_7/T_m \\ T_8/T_m \end{bmatrix}$$

$$= \overline{\mathbf{B}} \begin{bmatrix} T_1/T_m \\ T_2/T_m \\ T_3/T_m \\ T_4/T_m \\ T_5/T_m \\ T_6/T_m \\ T_7/T_m \\ T_8/T_m \end{bmatrix} = \overline{\mathbf{B}} \begin{bmatrix} \overline{T}_1 \\ \overline{T}_2 \\ \overline{T}_3 \\ \overline{T}_4 \\ \overline{T}_5 \\ \overline{T}_6 \\ \overline{T}_7 \\ \overline{T}_8 \end{bmatrix}. \tag{6}$$

上述方程可简写为

$$\begin{aligned} \overline{\boldsymbol{\tau}} &= \overline{\mathbf{B}}\,\overline{\mathbf{T}} \\ \overline{\mathbf{T}} &= \overline{\mathbf{B}}^{-1}\overline{\boldsymbol{\tau}}, \end{aligned} \tag{7}$$

其中，$\overline{\mathbf{B}}^{-1}$ 是**B**的广义逆矩阵。

因此，实现了推进器推力与力矩之间的转换并进行了归一化。式（6）中的所有力矩和力均归一化至 -1 到 1 的范围（$-1 \le \boldsymbol{\tau} \le 1$ 和 $-1 \le \mathbf{T} \le 1$），以便在跟踪控制过程中进行直接、简化的展示。

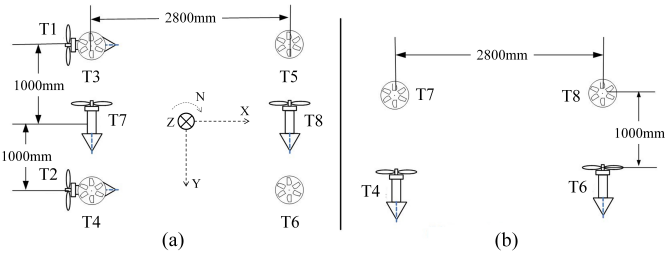


图2.“玉龙”无人艇推进系统中的推进器分布。(a) 俯视图，(b) 侧视图。

B. 问题陈述

本小节对“Yulong”航行器推进器系统的容错控制问题进行建模和说明，并介绍控制过程中的要求与约束。

1）水下航行器轨迹跟踪中的容错问题：为实现水下航行器的精准、高效控制，需要考虑推进器系统的故障情况，即一个或多个推进器可能发生故障。通过推导归一化期望力矩 $\boldsymbol{\tau}_d$，提出相应的控制策略，以维持航行器按预期运动并保持姿态。在故障条件下，力矩输出 $\boldsymbol{\tau}$ 由推进器推力的组合动作实现，并通过近似/优化方法进行分配。当推进器的工作状态发生变化时，近似/优化方法会推导出最优的重新分配方案，并在此过程中消除力矩误差。

对于采用多推进器系统的水下航行器，理想的容错控制可通过以下方式实现：

$$||e|| = ||\overline{\boldsymbol{\tau}}_d - \overline{\boldsymbol{\tau}}|| \ \rightarrow \ 0\,, \tag{8}$$

$$||\theta_e|| = \arccos \frac{\overline{\boldsymbol{\tau}}_d \bullet \overline{\boldsymbol{\tau}}}{||\overline{\boldsymbol{\tau}}_d|| \cdot ||\overline{\boldsymbol{\tau}}||} \ \rightarrow \ 0\,, \tag{9}$$

其中，$||e||$ 是通过期望力矩与航行器实际力矩之差的向量范数得到的幅值误差；$||\theta_e||$ 是方向误差，通过计算期望力矩与实际力矩的乘积与其向量范数之比的反余弦得到。除了幅值误差，方向误差对于水下航行器的轨迹跟踪控制也很重要，因为航行器的运动也取决于其动态输出（力矩）沿坐标轴的方向位移。航行器可能具有可接受的幅值动态误差，但同时出现过大的方向误差，从而导致不理想的跟踪结果。

因此，在设计容错控制时，应尽可能减小误差值，以获得良好的控制效果。

2）水下航行器容错控制的约束：在实际应用中，航行器的物理约束总会限制控制效果。水下航行器无法提供无限大的驱动输入（例如力矩/力）来完成导航，因此会产生执行器饱和问题。为实现可靠且可控的导航过程，通常会对水下航行器施加驱动限制。航行器推进器的最大力由载体本体可提供的最大力矩决定，是航行器控制问题中的关键约束。

为评估约束的影响，本文在仿真部分引入最大力矩 $\boldsymbol{\tau_m}$（见图3）。根据式（6）的定义，归一化力矩 $\boldsymbol{\tau}$和归一化力$\mathbf{T}$ 应限制在−1至1之间。在仿真过程中，这两个变量用于量化约束对轨迹跟踪的影响。

III. 基于GOA的容错轨迹跟踪控制（GFTC）设计

系统的基本控制架构如图3所示。控制策略由两部分组成：（1）轨迹跟踪部分，由基于水下航行器位置误差的外环辅助运动学控制器和基于速度状态向量的内环动力学力矩控制器构成；（2）容错控制部分，用于识别水下航行器推进系统的故障情况，并基于蚱蜢优化算法重新分配推进器力输出，以消除故障情况造成的影响。此外，还施加了最大推进器推力输出$\mathbf{T}_m$（见图3中的粗框）。针对水下环境扰动仿真，考虑了海流对力矩输出的影响。此外，还纳入了航行器位置形成过程中的环境噪声。本节将介绍控制设计的详细内容。

A. 轨迹跟踪控制的组成部分

本节将介绍构成水下航行器运动学控制和动力学控制的方法，并给出详细方程及稳定性分析。

1) 误差受限反步控制：假设水下航行器的最大速度向量为$\mathbf{v_m} = [v_{xm} \quad v_{ym} \quad v_{zm} \quad v_{\psi m}]^T$，输入$\mathbf{e}(t)$为期望轨迹与实际轨迹之间的误差。定义一个传递函数，以便在可接受范围内处理输入轨迹误差，

$$\mathbf{v}_e = f(\mathbf{e})\mu\mathbf{v}_m = \frac{\mathbf{e}(t)}{|\mathbf{e}(t)| + 1}\mu\mathbf{v}_m\,, \tag{10}$$

其中$\mathbf{v_m}$ 表示水下航行器各自由度在相应坐标轴上的最大速度；μ为根据变化要求选取的正数，本研究中取值为0.5。

因此，当$\mathbf{e}(t) \ \rightarrow 0$时，$\mathbf{v_e} = f(\mathbf{e})\mu\mathbf{v_m} \ \rightarrow \ 0$；当$\mathbf{e}(t) \ \rightarrow \infty$时，$\mathbf{v_e} = f(\mathbf{e})\mu\mathbf{v_m} \rightarrow$ ，受限最大速度 $\mu\mathbf{v_m}$。基于这一限制，传递函数$\mathbf{v}_e$ 将其控制输出限制在更小的范围内，同时使输入误差更快收敛。

收敛后的处理输出$\mathbf{v_e}$ 不应超过水下航行器速度的最大可能值，而 $f(\mathbf{e})$的定义为水下航行器在初始阶段提供了平滑过渡。因此，可以缓解水下航行器的速度突变问题。四个自由度上的向量$\mathbf{v_e}$ 可写为$\mathbf{v}_\mathbf{e} = [v_{ex} \quad v_{ey} \quad v_{ez} \quad v_{en}]^T$。此外，在反步法中，基于李雅普诺夫方法为每个子系统设计控制函数，并将其组合形成完整的控制律 [42]。因此，根据式（1）以及反步法的定义，反步法控制律中的误差变量为

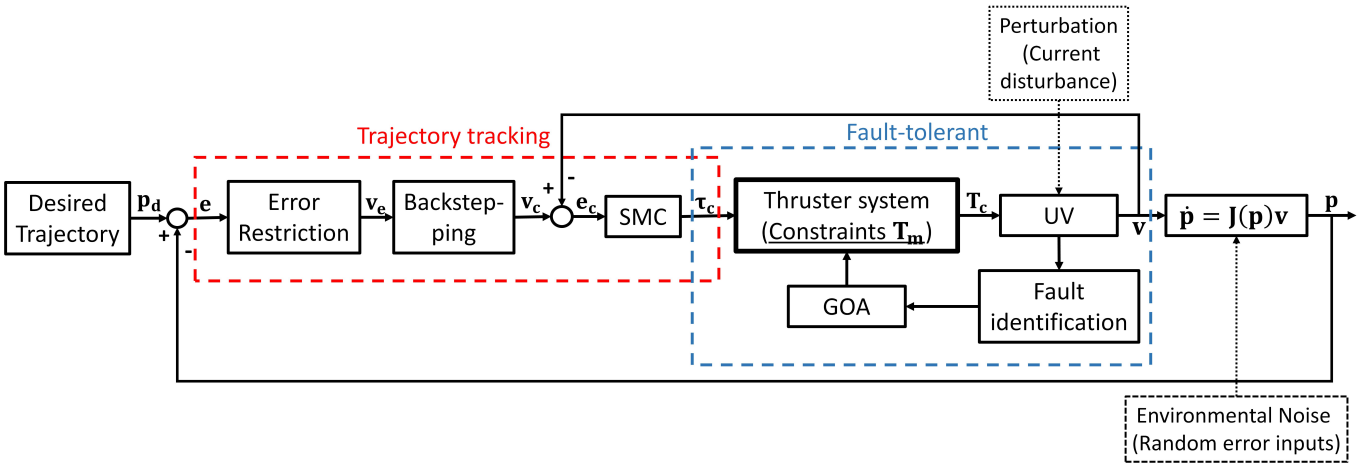


Fig. 3. 为水下航行器设计的容错轨迹跟踪控制示意图 .

将式（10）中处理后的受限输出代入后，可推导出反步控制的控制律为

$$\mathbf{v_c} = \begin{bmatrix} u_c \\ v_c \\ w_c \\ r_c \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} k(v_{ex}\cos\psi + v_{ey}\sin\psi) + u_d\cos v_{e\psi} - v_d\sin v_{e\psi} \\ k(-v_{ex}\sin\psi + v_{ey}\cos\psi) + u_d\sin v_{e\psi} - v_d\cos v_{e\psi} \\ w_d + k_z v_{ez} \\ r_d + k_\psi v_{e\psi} \end{bmatrix}.$$

其中， k、 k_z 和 k_ψ 为正数。

然后，将处理后的控制速度$\mathbf{v_c}$ 传递给水下航行器，并通过水下航行器的动力学模型进行计算，使其跟上期望轨迹。此外，通过构造李雅普诺夫函数$\Gamma_0 = \frac{1}{2}($ $e_x^2+e_y^2+e_z^2+e_\psi^2$)可以证明改进后的反步控制具有稳定性，其导数小于等于零（见附录A）。

2）滑模控制：为设计滑模控制，应引入期望动力学(s)。根据式（2），水下航行器动力学系统关于速度$\mathbf{v}$为二阶系统，因此动力学可设计为

$$\mathbf{s} = \left[\frac{d}{dt} + \lambda\right]^2 \int \mathbf{e_v} dt = \dot{\mathbf{e}}_\mathbf{v} + 2\lambda\mathbf{e_v} + \lambda^2 \int \mathbf{e_v} dt, \tag{12}$$

其中，$\frac{d}{dt}$ 为导数算子；$\mathbf{e_v}$ 表示由控制速度给出的误差（见图3），$\mathbf{e_v} = \mathbf{v_c} - \mathbf{v}$；且$\lambda > 0$ 为正参数 [43]。

然后对$\mathbf{s}$求导，可得

$$\dot{\mathbf{s}} = \ddot{\mathbf{e}}_\mathbf{v} + 2\lambda\dot{\mathbf{e}}_\mathbf{v} + \lambda^2\mathbf{e_v}, \tag{13}$$

其中 $\dot{\mathbf{e}}_\mathbf{v} = \dot{\mathbf{v}}_\mathbf{c} - \dot{\mathbf{v}}$

为了使系统状态与期望动力学保持一致，式（13）应等于零。这意味着系统状态位于完美跟踪的滑模面上。在同时，代入水下航行器动力学模型的方程（式（2））,

$$\begin{aligned} \dot{\mathbf{s}} &= \ddot{\mathbf{e}}_\mathbf{v} + 2\lambda\dot{\mathbf{e}}_\mathbf{v} + \lambda^2\mathbf{e_v} = 0 \\ \ddot{\mathbf{e}}_\mathbf{v} &+ 2\lambda(\dot{\mathbf{v}}_\mathbf{c} - \dot{\mathbf{v}}) + \lambda^2\mathbf{e_v} = 0 \\ \ddot{\mathbf{e}}_\mathbf{v} &+ 2\lambda(\dot{\mathbf{v}}_\mathbf{c} - (\boldsymbol{\tau} - \mathbf{Cv} - \mathbf{Dv} - \mathbf{g})\mathbf{M}^{-1}) + \lambda^2\mathbf{e_v} = 0 \\ \boldsymbol{\tau} &= \mathbf{M}(\dot{\mathbf{v}}_\mathbf{c} + \frac{\ddot{\mathbf{e}}_\mathbf{v}}{2\lambda} + \frac{\lambda}{2}\mathbf{e_v}) + \mathbf{Cv} + \mathbf{Dv} + \mathbf{g}. \end{aligned} \tag{14}$$

标准滑模控制律定义为

$$\boldsymbol{\tau} = \hat{\boldsymbol{\tau}} + \boldsymbol{\tau_c}, \tag{15}$$

其中 $\hat{}\boldsymbol{\tau}$ 表示主要控制律，它是连续的，并基于模型设计，旨在使轨迹始终保持在滑模面上。 $\boldsymbol{\tau_c}$ 表示切换控制律，用于处理模型不确定性。当轨迹偏离控制时，使用 $\boldsymbol{\tau_c}$ 将轨迹推回滑模面，以继续实现令人满意的跟踪。对于式(14)，假设基于误差加速度反馈控制进行简化 $\ddot{\mathbf{e}}_\mathbf{v} \approx -k\dot{\mathbf{e}}_\mathbf{v}$，以降低计算复杂度，则主要控制律中的估计项 $\hat{}\boldsymbol{\tau}$可推导为

$$\hat{\boldsymbol{\tau}} = \hat{\mathbf{M}}(\dot{\mathbf{v}}_\mathbf{c} + \frac{-k\dot{\mathbf{e}}_\mathbf{v}}{2\lambda} + \frac{\lambda}{2}\mathbf{e_v}) + \hat{\mathbf{C}}\mathbf{v} + \hat{\mathbf{D}}\mathbf{v} + \hat{\mathbf{g}}, \tag{16}$$

$\hat{}\ \hat{}\ \hat{}$其中 $\mathbf{M}$、$\mathbf{C}$、$\mathbf{D}$、$\hat{}\mathbf{g}$ 分别是$\mathbf{M}$、$\mathbf{C}$、$\mathbf{D}$ 和$\mathbf{g}$的估计值；相应的近似值可通过实际情况获得 [44]。

滑模控制中的切换项 $\boldsymbol{\tau_c}$ 可定义为

$$\boldsymbol{\tau_c} = -\mathbf{K}_1\mathbf{s} - \mathbf{K}_2|s|^r sign(\mathbf{s}), \tag{17}$$

其中， $sign(\mathbf{s})$ 是$\mathbf{s}$的非线性符号函数；$\mathbf{K}_1$ 和$\mathbf{K}_2$ 为正系数，$\mathbf{K}_1 \geq \eta + F$ 和 $\mathbf{K}_2 \geq \eta + F$， η 是设计参数，通常取正常数； $0 < r < 1$；而 F 表示系统实际输出与其估计值之差的上界，

$$F = |f(\mathbf{v}) - \hat{f}(\mathbf{v})|. \tag{18}$$

此外，在控制律中加入自适应变化项 $\dot{\tilde{\tau}}_{est}$，其中 $\tilde{\tau}_{est} = \Gamma\mathbf{s}$ 和 Γ 表示一个正

常数。因此，最终的滑模控制律定义为

$$\begin{aligned}\boldsymbol{\tau} &= \hat{\boldsymbol{\tau}} + \widetilde{\tau}_{est} + \boldsymbol{\tau}_{\mathbf{c}} \\ &= \hat{\mathbf{M}}(\dot{\mathbf{v}}_{\mathbf{c}} + \frac{-k\dot{\mathbf{e}}_{\mathbf{v}}}{2\lambda} + \frac{\lambda}{2}\mathbf{e}_{\mathbf{v}}) + \hat{\mathbf{C}}\mathbf{v} + \hat{\mathbf{D}}\mathbf{v} + \hat{\mathbf{g}} \\ &\quad + \widetilde{\tau}_{est} - \mathbf{K}_1\mathbf{s} - \mathbf{K}_2|s|^r\mathrm{sign}(\mathbf{s})\,. \end{aligned} \tag{19}$$

滑模控制稳定性的详细证明见附录 B。

B. 容错控制的组成部分

容错控制设计主要基于对推进器系统所需推力的调整，各推进器协同工作，以提供所需力矩。推进器推力的调整由蚱蜢优化算法（GOA）计算得出；在完成故障识别后，该算法可有效消除推进系统故障引起的误差。本节将详细介绍控制策略。

1) 加权矩阵： 为量化多推进器系统在故障情况下的损坏程度，引入加权矩阵$\mathbf{W}$ 。矩阵$\mathbf{W}$ 决定推进器的工作状态，通常定义为对角矩阵，

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_8 \end{bmatrix} \tag{20}$$

其中， $w_j > 0$ 是 j^{th} 推进器的权重。如果所有推进器均处于理想工作状态且没有功率损失，$\mathbf{W}$ 将为单位矩阵，这意味着所有 $w_j = 1$。如果任一推进器发生功率损失，其对应的权重将根据损失程度相应降低。例如，当T_1 推进器发生 20 % 的功率损失时，加权矩阵中的 w_2 相应设为 0.8。

由于不同状态下推进器推力与航行器力矩之间的关系已在式（7）中定义并给出，因此故障情况下力矩与推力之间的以下转换关系定义为：

$$\overline{\boldsymbol{\tau}} = \overline{\mathbf{B}}\mathbf{W}\overline{\mathbf{T}}, \tag{21}$$

其中，$\overline{\mathbf{T}}$ 是 UV 工况下的控制参数，通过优化方法求得，例如本研究采用的 GOA 方法。

此外，为与 GOA 方法进行比较，本文采用加权伪逆矩阵法，该方法基于定义的加权矩阵确定：

$$\overline{\mathbf{T}} = \overline{\mathbf{B}}_w^+\overline{\boldsymbol{\tau}}_d = (\mathbf{W}\overline{\mathbf{B}}^T(\overline{\mathbf{B}}\mathbf{W}\overline{\mathbf{B}}^T)^{-1})\overline{\boldsymbol{\tau}}_d, \tag{22}$$

其中，$\overline{\mathbf{B}}_w^+$ 是将损伤信息传递至推进系统并相应进行调整的矩阵。因此，可以推导出故障情况下的推进器力结果。例如，如果推进器 T_1 在遭受 30% 的功率损失后只能提供 70% 的功率，加权伪逆矩阵法将要求 T_1 输出更大的力（$\frac{1}{0.7}\times$ 原始力），这样在功率损失削弱 30% 后仍能达到相同的力。

随后，采用 T-近似或 S-近似方法，使所得推力值处于推进器推力上限范围内，这通常称为伪逆（P-I）矩阵近似。T-近似通过减去或加上状态值大于 1 或小于 -1 的超出部分，将所有归一化力$\mathbf{T}$ 限制在 [-1, 1] 之间，其中

$$\overline{\mathbf{T}}_t = \begin{cases} \overline{T}_i, & \overline{T}_i \in [-1, 1] \\ 1, & \overline{T}_i > 1 \\ -1, & \overline{T}_i < -1 \end{cases}$$

S-近似通过将所有状态下最大的归一化力乘以其倒数比值，实现 [-1, 1] 的限制，其中

$$\overline{\mathbf{T}}_s = \frac{1}{max(\overline{T}_i)}\overline{\mathbf{T}}, i = 1, 2, ..., 8.$$

例如，在 S-近似中，如果某个状态的最大归一化力达到 2，则所有归一化力都会乘以$\frac{1}{2}$ 的比值，以确保它们不超出 -1 到 1 的范围。

在本研究的仿真部分中，采用 T近似方法作为典型的伪逆（P-I）矩阵近似，与所提出的基于 GOA 的容错控制进行比较。由于 T-近似通常产生的误差小于 S-近似，因此它在水下航行器容错控制的实际应用中更为广泛[34, 45]。

2）蚱蜢优化算法：蚱蜢优化算法于 2017 [36] 首次提出。作为一种基于群体智能理论、模仿蚱蜢活动而发展出的算法，GOA 在传统群智能算法之上表现更优，因为它通过在“成虫”和“幼虫”形态之间切换，在收敛速度与精度之间实现了令人满意的平衡。当 GOA 处于“成虫”形态时，会根据每个个体的位置进行全局搜索；个体之间以吸引的方式进行大范围探索，从而实现快速收敛。而当 GOA 处于“幼虫”形态时，则会根据最佳个体缩小搜索范围并保持排斥区域，从而避免陷入局部最小值，确保搜索精度。

根据蚱蜢群的运动，可以定义一个数学模型来描述其群聚行为 [46, 47]

$$X_i = \sum_{j=1, j\neq i}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - G_i + A_i, \tag{23}$$

其中， X_i 表示 i_{th} 蚱蜢的下一个位置；$s(r)$是社会交互函数，其优化形式为$s(r) = 0.5e^{-r/1.5} - e^{-r}$。 $|x_j - x_i|$ 表示当前位置与 i_{th} 蚱蜢和 j_{th} 蚱蜢之间的距离，

$(x_j - x_i)/d_{ij}$ 是从 i_{th} 蚱蜢的位置指向 j_{th} 蚱蜢的单位向量。 G_i 表示作用于 i_{th} 蚱蜢的引力； A_i 表示风平流，假设其方向始终朝向目标 T_i。

基于本控制案例所作的假设，即忽略引力且风力始终指向目标，式（23）可转换为

$$X_i^d = c(\sum_{j=1, j\neq i}^{N} c\frac{ub_d - lb_d}{2} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}}) + T_d \quad (24)$$

其中， c 是一个递减系数，用于缩小舒适区、斥力区和吸引区，其确定方式为 $c = c_{\max} - l(c_{\max} - c_{\min})/L$，$c_{\max}$ 为最大值，$c_{\min}$ 为最小值， l 表示当前迭代次数，L 为最大迭代次数。在本研究中，我们通过试错法设定 $c_{\max} = 1$ 和 $c_{\min} = 0.00001$ 。变量 ub_d 表示该案例的上界，而 lb_d 表示下界，在本设计中分别为1和-1。 T_d 是当前迭代的期望解。这些参数通过提高局部解更新速度随迭代次数增加的幅度，来加快优化的收敛，从而使GOA方法能够高效搜索[48]。

将GOA应用于UV推进器力重分配的伪代码总结如下（见算法1），其中适应度评估由误差评估替代，误差评估见公式（8）和（9）。

Algorithm 1: GOA algorithm embedded in the fault-tolerant component

Input: $\bar{\tau}_d$: desired normalized torque matrix
W: weighting matrix (fault identification)
Output: $\bar{T}$: A vector containing the optimal allocation of the normalized forces for eight thrusters
Initialize the swarm X*i* (*i*=1,2,---,10)
Initialize the c_{max}, c_{min}, and maximum number of iterations L
Calculate the fitness of each search agent to address the best search agent $\bar{T}$ with the minimum errors (Eq. s (8) and (9))
while ($l < L$)
Update c = $c_{max} - l$ (c_{max} - c_{min}) / L
for each search agent
Update the value of the current search agent by Eq. (24)
Retrieve the current search agent if it excess the limitation of [-1, 1]
end for
Update $\bar{T}$ if there is a better solution
$l = l + 1$
end while
Return $\bar{T}$

所提出的GOA方法流程总结如下:

1) 首先初始化群体，设置 10 组由 8 个介于 -1 和 1 之间的随机数构成的数据，表示归一化的八推进器组，并将每组视为一个搜索代理；2) 接下来，根据由式 (8) 和 (9) 组合而成的目标函数计算每个搜索代理的适应度，并将误差最小的个体确定为最优个体，即 $||e|| + ||\theta_e|| \to 0$，每个个体的约束范围设置为 [-1,1];

3) 根据迭代次数更新参数 c，以加快收敛。如果迭代次数达到最大值，则停止；否则继续更新 c；

4）根据公式（24）更新搜索代理的位置（数值），并将更新后个体的适应度与适应度最佳的个体进行比较。如果更新后的适应度更优，则更新个体的位置；否则不更新。

5）更新迭代次数，并从步骤3）开始重复循环。

C. 扰动分量

由于“玉龙”无人艇UV模型是为大坝检测而设计的，通常在近岸水下环境中运行，因此水流扰动可以表示为波函数的常规组合形式，如 [49–51]

$$\tau_p = A_{p1}\cos(\omega_{p1}t)\sin(\omega_{p2}t) + A_{p2}\cos(\omega_{p3}t)\sin(\omega_{p4}t), \quad (25)$$

其中， A_{p1}、 A_{p2} 和 ω_{p1} 至 ω_{p4} 为随机系数，用于合成水流的随机性，从而适当应对水下环境扰动。 A_{p1} 和 A_{p2} 的取值范围设为四个轴力矩输出的10%，例如，如果 τ_x 输出约为$100N$，则取值范围为 $[-10, 10]$。系数ω_{p1} 至 ω_{p4} 的取值范围为-1至1。

此外，考虑到环境噪声会在位置形成阶段对数据传输造成扰动，仿真中加入了随机误差输入。假定随机误差处于 [-0.1、0.1] 范围内，并由传感器滤除，这与实际UV工况相符。

IV. 仿真结果与分析 YSIS

本节展示并分析所提出的容错控制方法与传统近似方法在折线和螺旋轨迹跟踪中的仿真结果。鉴于单故障（一个推进器损坏）和双故障（两个推进器损坏）较为常见，本文采用这两种故障情况进行仿真。

A. 螺旋轨迹跟踪

本节假设推进器 T_1 发生故障，功率损失为 100%。期望螺旋轨迹的初始位置设为 (0,0,0,0)，而控制轨迹的初始位置设为 (0,10,0,0)。初始位置的差异用于测试两种跟踪策略在某个坐标轴（即 y轴）上存在一定偏差时的修正能力。假设期望轨迹为 $x_d = 10\sin 0.2t, y_d = 10 - 10\cos 0.2t$, $z_d = 0.5t$ 和 $\psi_d = 0.2t$，仿真持续时间为 50 秒。

GFTC（红色虚线）的跟踪结果迅速消除初始误差，并一直跟随期望螺旋轨迹直至仿真结束，尽管此时 T_1 推进器被认为已完全损坏（图4(a)）。在单故障情况下，P-I近似方法（蓝色）无法与期望螺旋线重合，其偏差的突变从开始阶段产生，并在整个过程中导致轨迹畸变。

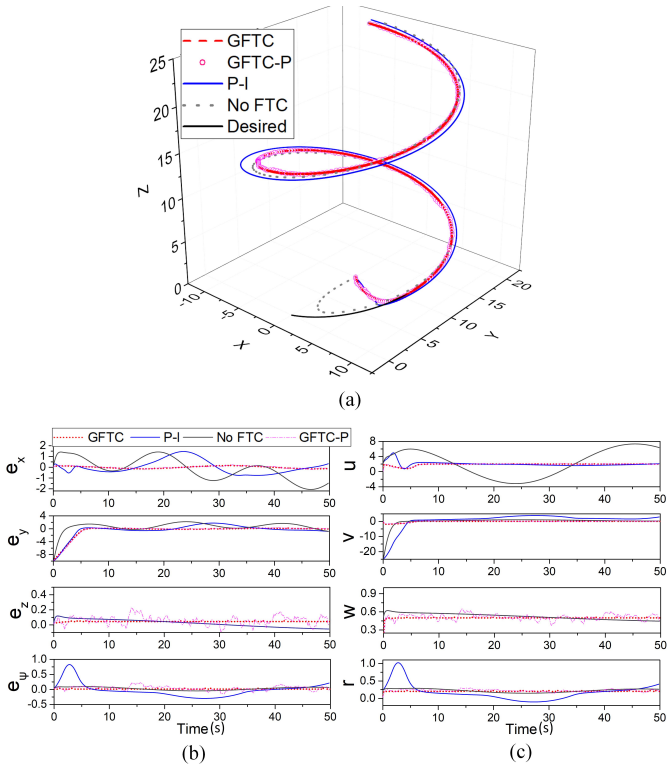


图4. 单故障情况下，使用有或无扰动的GFTC以及基于P-I近似的容错控制进行螺旋轨迹跟踪的结果。(a) 轨迹对比，(b) 跟踪误差对比，(c) 控制速度对比。

因此，在考虑动态约束时，P-I方法无法补偿单个推进器的功率损失，从而导致误差不断增大，并产生图4(b)和(c)所示的带有突变的过大速度；而GFTC则实现了平滑的误差曲线和控制速度曲线，表明该方法具有良好的跟踪性能。此外，GFTC-P（粉色）的结果在处理航行器的位置信息时，考虑了水流和环境噪声引起的扰动影响。在扰动影响下，GFTC-P的结果仍保持了与未受影响的GFTC相似的跟踪轨迹，验证了所提出控制方法的鲁棒性。

表I. 跟踪螺旋线时，单故障情况下有或无扰动的GFTC及基于P-I的容错控制的最大速度

	u_c (m/s)	v_c (m/s)	w_c (m/s)	r_c (米/秒)
GFTC	2.0008	-1.8978	0.5002	0.2016
P-I	4.9719	-25	0.6125	1.0034
GFTC-P	2.0150	-1.8944	0.6404	0.2836

GFTC方法（红色）、基于P-I近似的容错控制（蓝色）、无容错控制的单故障情况（灰色），以及受扰动影响的GFTC-P（粉色）的跟踪误差如图4(b)所示。GFTC方法的误差曲线消除了初始偏差，并迅速收敛至零。基于P-I近似的容错控制误差则出现明显波动，且无法在所有坐标轴上消除；此外，在 ψ轴上，P-I误差曲线的波动甚至比无容错控制时更大，这与图4(a)所示的轨迹跟踪性能相符。这表明，在施加动态约束的单故障条件下，P-I近似容错控制无法维持期望的螺旋轨迹跟踪；而GFTC方法能够完成容错轨迹跟踪任务，并获得令人满意的运动学输出。图4(c)中的速度变化也支持这一结论。P-I方法在x轴和y轴产生了大幅超限速度，所需最大值分别为4.9719 m/s和-25 m/s；而GFTC方法将x轴和y轴的速度有效限制在 [-2, 2]m/s的约束范围内，最大输出分别为2.0008 m/s和-1.8978 m/s（表I）。此外，与P-I方法相比，GFTC在所有坐标轴上的速度曲线都平滑得多。即使在GFTC-P仿真中考虑扰动影响，虽然出现了轻微抖振，误差和控制速度仍成功限制在可接受范围内；x轴和y轴的最大值分别为2.015 m/s和-1.8944m/s，且ψ 轴所需的控制速度远低于前者。因此，即使存在外部扰动，所提出的GFTC方法在单故障情况下跟踪期望轨迹的有效性也得到了验证。

B. 三维折线跟踪

本节采用三维折线作为参考跟踪轨迹，因为“玉龙”无人艇通常沿类似折线的路径航行，以检测大坝损伤。

期望轨迹的初始位置设为 (0，0，0，0)，而控制轨迹的初始位置设为 (0，2.5，0，0)。采用特定的折线函数，仿真持续 20 秒：

$$
\begin{aligned}
&x_d = t, 0 \le t \le 20,\\
&y_d = \begin{cases} t, & 0 \le t \le 5 \\ 5, & 5 < t \le 10 \\ t-5, & 10 < t \le 15 \\ 10, & 15 < t \le 20 \end{cases}\\
&z_d = \begin{cases} t, & 0 \le t \le 5 \\ 5, & 5 < t \le 10 \\ t-5, & 10 < t \le 15 \\ 10, & 15 < t \le 20 \end{cases}\\
&\psi_d = 0.2, 0 \le t \le 20.
\end{aligned}
$$

1) 单故障情况：假设其中一个推进器发生故障，功率损失为100%。跟踪轨迹结果如图5(a)所示。GFTC（红色虚线）在消除y轴上的初始误差后，保留了所需的折线跟踪轨迹，而无需考虑推进器的功率损失。此外，考虑扰动影响的GFTC-P（粉色）结果保持了与未受影响情况相似的跟踪

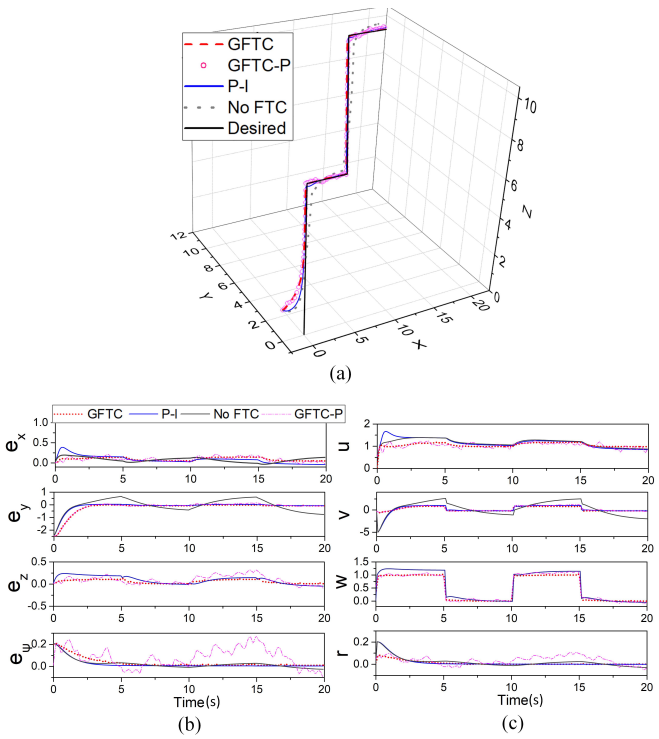


图 5. 单故障情况下，采用有无扰动的 GFTC 和基于 P-I 近似的容错控制进行折线跟踪的结果。(a) 轨迹对比；(b) 跟踪误差对比；(c) 控制速度对比。

轨迹，验证了所提出跟踪控制的鲁棒性。P-I方法（蓝色）未能跟上期望轨迹，尤其是在需要较大动态输入的转弯点处。当存在物理约束（最大扭矩/力）时，P-I方法无法弥补推进力的损失，因此产生了具有较大波动的误差以及过大的控制速度，如图5(b)和(c)所示。

表 II. 单故障情况下跟踪折线时的最大速度

	u_c (m/s)	v_c (m/s)	w_c (m/s)	r_c (米/秒)
GFTC	1.1738	0.8212	1.0012	0.0782
P-I	1.7278	-5.0713	1.2289	0.2
GFTC-P	1.2410	1.2188	1.1192	0.1035

四个轴上的误差如图5(b)所示，GFTC（红色）成功消除了跟踪误差。相比之下，基于P-I近似的容错控制（蓝色）未能消除误差：当轨迹要求急转弯时，P-I方法产生的过大动态输出无法满足施加的动态约束，因此在转弯段造成较大的轨迹偏差。四个轴上的速度变化如图5(c)所示，P-I方法出现剧烈波动，且未能将y轴上的控制速度保持在期望范围内（见表II中的y轴）。P-I方法在y轴上的速度达到惊人的数值−5.0713 m/s，远超航行器预设的−2 m/s至2 m/s期望范围。同时，GFTC成功将运动学输出限制在约束范围内，曲线平滑，波动也小得多。此外，在GFTC-P仿真中考虑扰动时（粉色），尽管出现了抖振问题，但误差仍被限制在可接受范围内；在大多数坐标轴上，其运动学输出的变化范围均小于P-I方法，x轴和y轴上的最大值分别为1.2410 m/s和1.2188 m/s。这些结果表明，所提出的基于GOA的容错控制具有良好的有效性和鲁棒性。

2）双故障情形：本节比较了 GFTC、基于 P-I 近似的 FTC 以及考虑环境扰动的 GFTC 的效果。假设推进系统中的两个推进器（T_1 和 T_8）发生 100% 功率损失。

GFTC 方法（红色虚线）成功消除了 y轴方向上的初始误差，并使跟踪轨迹一直保持期望状态直至结束（图 6(a)）。在扰动影响下，GFTC-P（粉色）的结果与未受影响的 GFTC 轨迹保持相似的跟踪轨迹；在第二个转弯段，其跟踪曲线比第一个转弯段更加平滑，这验证了它在双故障情形下跟踪期望折线的鲁棒性。同时，基于 P-I 近似的 FTC 无法跟踪期望轨迹，与单故障情况相比，甚至出现了更大的偏差（图 5(a)）。这表明，当推进器系统中的功率损失程度不同时，GFTC 方法能够平衡跟踪误差，从而证明了所提出 FTC 方法的鲁棒性。

与单故障情形类似，在双故障情形下，GFTC方法的误差曲线消除了初始偏差，并迅速收敛至零（图 6(b)）。然而，与单故障情形相比，P-I方法在x轴、y轴和ψ 轴上出现了剧烈的误差振荡，其表现甚至比未采用容错控制的双故障情形更差。这表明，基于P-I近似的容错控制会受到水下航行器推进系统损坏程度的严重影响，无法抵消推进器功率损失过大所产生的误差，例如双故障情形。图 6(c) 中的速度变化也支持这一结论：由于推进器功率损失，P-I方法无法将速度限制在允许范围内；表 III 所列的x轴、y轴和 ψ轴最大速度分别高达17.5815m/s、25.9814m/s和1.4485m/s，导致图 6(a) 所示的跟踪彻底失败。无论故障情况如何变化，GFTC方法的速度在整个过程中均保持在允许域内。在实际水下环境中考虑扰动的GFTC-P仿真也成功将误差和控制速度限制在预定范围内，x轴和y轴的最大值分别为1.2473m/s和0.9646m/s。因此，无论发生单故障还是双故障，所提出的GFTC方法在跟踪期望折线方面均得到验证。

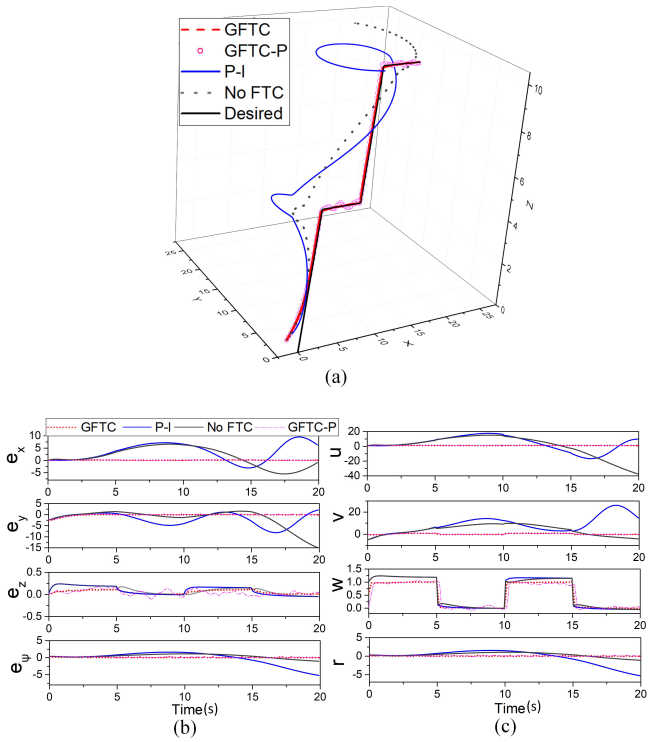


图 6. 双故障情形下，采用有无扰动的 GFTC 和基于 P-I 近似的容错控制进行折线跟踪的结果。(a) 轨迹对比，(b) 跟踪误差对比，(c) 控制速度对比。

表 III. 双故障情形下跟踪折线时的最大速度

	u_c (m/s)	v_c (m/s)	w_c (m/s)	r_c (米/秒)
GFTC	1.1710	0.8158	0.9989	0.0782
P-I	17.5815	25.9814	1.2289	1.4485
GFTC-P	1.2473	0.9646	1.0367	-0.0885

V. 结论

本文通过蚱蜢优化与基于反步法和滑模控制的级联控制（GFTC），解决了“玉龙”水下航行器的容错轨迹跟踪问题。GFTC策略采用改进反步算法来限制运动学输出，并利用蚱蜢优化算法（GOA）在允许域内实现优化的推进器力重新分配。在跟踪折线或螺旋线时遇到故障情况，与传统容错控制（例如基于加权伪逆矩阵近似的方法）相比，GFTC显著减小了轨迹跟踪误差，并消除了执行器饱和问题。此外，本文还验证了所提出的容错控制在存在环境扰动时的鲁棒性，为未来将拓展至实际应用的实验研究奠定了基础。

附录A 误差受限反步控制稳定性证明

根据李雅普诺夫稳定性理论，选取一个特殊的李雅普诺夫函数 Γ_0 ，

$$\Gamma_0 = \frac{1}{2}(e_x^2 + e_y^2 + e_z^2 + e_\psi^2)\,. \tag{26}$$

根据式（1）和（11），可得式（25）的导数，从而证明反步控制系统的稳定性，

$$\begin{aligned}
\dot{\Gamma}_0 =& e_x\,\dot{e}_x + e_y\,\dot{e}_y + e_z\,\dot{e}_z + e_\psi\,\dot{e}_\psi \\
=& e_x\,(\dot{x}_d - \dot{x}) + e_y\,(\dot{y}_d - \dot{y}) \\
&+ e_z(\dot{z}_d - \dot{z}) + e_\psi\,(\dot{\psi}_d - \dot{\psi}) \\
=& e_x\,[(\cos\psi_d u_d - \sin\psi_d v_d) - (\cos\psi u_c - \sin\psi v_c)] \\
&+ e_y\,[(\sin\psi_d u_d + \cos\psi_d v_d) - (\sin\psi u_c + \cos\psi v_c)] \\
&+ e_z(w_d - w_c) + e_\psi\,(r_d - r_c) \\
=& e_x\,[(\cos\psi_d u_d - \sin\psi_d v_d) \\
&- (kv_{ex} + u_d(\cos\psi\cos v_{e\psi} - \sin\psi\sin v_{e\psi}) \\
&+ v_d(\sin\psi\cos v_{e\psi} - \cos\psi\sin v_{e\psi}))] \\
&+ e_y\,[(\sin\psi_d u_d + \cos\psi_d v_d) \\
&- (kv_{ey} + u_d(\sin\psi\cos v_{e\psi} + \cos\psi\sin v_{e\psi}) \\
&- v_d(\sin\psi\sin v_{e\psi} - \cos\psi\cos v_{e\psi}))] \\
&+ e_z(-k_z v_{ez}) + e_\psi\,(-k_\psi v_{e\psi}) \\
\le& -ke_x v_{ex} - ke_y v_{ey} - k_z e_z v_{ez} - k_\psi e_\psi v_{e\psi}\,.
\end{aligned} \tag{27}$$

根据$\mathbf{v_e}$的定义，$\mathbf{e}(t)$符号相同（参见式（10）的定义）；k、k_z、k_ψ 为正常数。认为式（26）的结果小于或等于零，这表明所设计的改进型反步控制器是稳定的。

附录B 滑模控制稳定性证明

为证明滑模控制的稳定性，构造一个李雅普诺夫函数，

$$\mathbf{V} = \frac{1}{4\lambda}\mathbf{s}^T\mathbf{M}\mathbf{s} + \frac{1}{2}\mathbf{Q}^T\Gamma^{-1}\mathbf{Q}\,, \tag{28}$$

其中$\mathbf{Q} = \tilde{\tau}_r - \tilde{\tau}_{est}$ 且 $\tilde{\tau}_r = \mathbf{M}\dot{\mathbf{v}}_r + \mathbf{C}\mathbf{v}_r + \mathbf{D}\mathbf{v} + \tilde{\mathbf{g}}$。

前文已给出$\mathbf{e_v} = \mathbf{v_c} - \mathbf{v}$，以及$\dot{\mathbf{s}} = \dot{\mathbf{e}}_\mathbf{v} + 2\lambda\mathbf{e_v} + \lambda^2\int\mathbf{e_v}dt$，由此可推导出两个方程：

$$\mathbf{v} = \mathbf{v_c} - \frac{\mathbf{s} - \dot{\mathbf{e}}_\mathbf{v} - \lambda^2\int\mathbf{e_v}dt}{2\lambda}, \tag{29}$$

$$\dot{\mathbf{v}} = \dot{\mathbf{v}}_\mathbf{c} - \frac{\dot{\mathbf{s}} - \ddot{\mathbf{e}}_\mathbf{v} - \lambda^2\mathbf{e_v}}{2\lambda}, \tag{30}$$

因此，可定义以下各项，

$$\mathbf{v_r} = \mathbf{v_c} + \frac{\dot{\mathbf{e}}_\mathbf{v} + \lambda^2\int\mathbf{e_v}dt}{2\lambda}, \tag{31}$$

$$\dot{\mathbf{v}}_\mathbf{r} = \dot{\mathbf{v}}_\mathbf{c} + \frac{\ddot{\mathbf{e}}_\mathbf{v} + \lambda^2\mathbf{e_v}}{2\lambda}. \tag{32}$$

代入式（2）后，

$$\begin{aligned}&\mathbf{M}\frac{\dot{\mathbf{s}}}{2\lambda}+\mathbf{C}\frac{\mathbf{s}}{2\lambda}\\&=\mathbf{M}(\dot{\mathbf{v}}_\mathbf{c}+\frac{\ddot{\mathbf{e}}_\mathbf{v}+\lambda^2\mathbf{e}_\mathbf{v}}{2\lambda})+\mathbf{C}(\mathbf{v}_\mathbf{c}+\frac{\dot{\mathbf{e}}_\mathbf{v}+\lambda^2\int\mathbf{e}_\mathbf{v}dt}{2\lambda})\\&+\mathbf{D}\mathbf{v}+g-\tau=\mathbf{M}\dot{\mathbf{v}}_\mathbf{r}+\mathbf{C}\mathbf{v}_\mathbf{r}+\mathbf{D}\mathbf{v}+g-\tau.\end{aligned}\tag{33}$$

根据前述定义，式(27)的导数可简化为：

$$\begin{aligned}\dot{\mathbf{V}}&=\frac{1}{4\lambda}(\mathbf{s}^T\dot{\mathbf{M}}\mathbf{s}+\dot{\mathbf{s}}^T\mathbf{M}\mathbf{s}+\mathbf{s}^T\mathbf{M}\dot{\mathbf{s}})\\&+\frac{1}{2}\dot{\mathbf{Q}}^T\Gamma^{-1}\mathbf{Q}+\frac{1}{2}\mathbf{Q}^T\Gamma^{-1}\dot{\mathbf{Q}}\\&=\frac{1}{2\lambda}\mathbf{s}^T(\mathbf{M}\dot{\mathbf{s}}+\mathbf{C}\mathbf{s})+\frac{1}{2}\dot{\mathbf{Q}}^T\Gamma^{-1}\mathbf{Q}+\frac{1}{2}\mathbf{Q}^T\Gamma^{-1}\dot{\mathbf{Q}}\\&=\mathbf{s}^T(\mathbf{M}\dot{\mathbf{v}}_\mathbf{r}+\mathbf{C}\mathbf{v}_\mathbf{r}+\mathbf{D}\mathbf{v}+g-\tau)+\dot{\mathbf{Q}}^T\Gamma^{-1}\mathbf{Q}.\end{aligned}\tag{34}$$

代入式（19）后，

$$\begin{aligned}\dot{\mathbf{V}}&=\mathbf{s}^T(\mathbf{M}\dot{\mathbf{v}}_\mathbf{r}+\mathbf{C}\mathbf{v}_\mathbf{r}+\mathbf{D}\mathbf{v}+g-\tau)\\&+(\dot{\tilde{\tau}}_r-\dot{\tilde{\tau}}_{est})^T\Gamma^{-1}\mathbf{Q}\\&=-\mathbf{s}^T(\mathbf{K}_1\mathbf{s}+\mathbf{K}_2|\mathbf{s}|^r sign(\mathbf{s}))+(\dot{\tilde{\tau}}_r)^T\Gamma^{-1}\mathbf{Q}.\end{aligned}\tag{35}$$

由于水下航行器速度较慢，动态项 $\tilde{\tau}_r$ 有界，且 $\mathbf{s}^T\ \ _1+\ \ _2|\ |^r\ \ \ \ \ \ \geq(\mathbf{Ks}\ \mathbf{K}s\mathrm{sign}(\mathbf{s}))\ (\tilde{\tau}_r)^{\mathrm{T}}\Gamma^{-1}\mathbf{Q}$。在设计阶段为$\mathbf{K}_1$，$\mathbf{K}_2$ 和 Γ 赋予足够大的值后，可实现$\dot{\mathbf{V}}\leq 0$，并确保$\mathbf{V}$ 有界，从而得出$\mathbf{Q}$有界的结论。随后设计一个新的李雅普诺夫函数如下：

$$\mathbf{V}_2=\frac{1}{4\lambda}\mathbf{s}^T\mathbf{M}\mathbf{s},\tag{36}$$

其导数可推导为：

$$\dot{\mathbf{V}}_2=\mathbf{s}^T(\mathbf{Q}-\mathbf{K}_1\mathbf{s}-\mathbf{K}_2|s|^r sign(s)),\tag{37}$$

其中 $0<r<1$。假设 $||Q||<a$,

$$\dot{\mathbf{V}}_2\leq\frac{1}{2}||s||^2+\frac{1}{2}a-\lambda_{min}(\mathbf{K}_1)||s||^2-\lambda_{min}(\mathbf{K}_2)||s||^{1+r},\tag{38}$$

选择 $\mathbf{K}_1$，当 $\lambda_{min}(\mathbf{K}_1)>\frac{1}{2}+\beta$ 时，其中 $\beta>0$,

$$\dot{\mathbf{V}}_2\leq-\beta||s||^2-\lambda_{min}(\mathbf{K}_2)||s||^{1+r}+\frac{1}{2}a,\tag{39}$$

这使得李雅普诺夫函数在有限时间内收敛到接近零的范围，且 $\mathbf{s}$ 在有限时间内收敛到接近零的范围。因此，可以认为李雅普诺夫定理的假设条件得到了满足，从而证明了所设计滑模控制的稳定性。

参考文献

[1] T. Baraniuk、R. Simoni 和 L. Weihmann，“自主水下航行器的容错架构”，载于*2018*年 *IEEE/OES* 自主水下航行器研讨会论文集，葡萄牙波尔图，2018年，第 1–6 页。[2] C. Huang、F. Naghdy、H. Du 和 H. Huang，“容错线控转向系统：综述”，《控制年度评论》，第 47 卷，第 98–111 页，2019年。[3] X. Qi、J. Qi、D. Theilliol、Y. Zhang、J. Han、D. Song 和 C. Hua，“单旋翼飞行器故障诊断与容错控制方法综述”，*J. Intell. Robot. Syst.*，第 73 卷，第 1–4 期，第 535–555 页，2014年 1 月。[4] K. Lu、Y. Xia、C. Yu 和 H. Liu，“执行器饱和和故障条件下刚性航天器的有限时间跟踪控制”，*IEEE Trans. Autom. Sci. Eng.*，第 13 卷，第 1 期，第 368–381 页，2016年 1 月。[5] Z. Mao、X. Yan、B. Jiang 和 M. Chen，“存在执行器故障和不确定性的高速列车自适应容错滑模控制”，*IEEE Trans. Intell. Transp. Syst.*，第 21 卷，第 6 期，第 2449–2460 页，2020年 7 月。

[6] R. T. Meyer、S. C. Johnson、R. A. DeCarlo、S. Pekarek 和 S. D. Sudhoff，《混合动力电动汽车容错控制》，J. Dyn. Syst-T. ASME，第140卷，第2期，第1–12页，2018年。[7] R. Xiong、Q. Yu 和 W. Shen，《电动汽车锂离子电池传感器故障诊断与容错技术综述》，载于《2018年IEEE工业电子与应用会议》（ICIEA），中国武汉，2018年，第406–410页。[8] J. Gong 和 M. Yang，《基于虚拟可重构电路与神经网络架构的进化容错方法》，IEEE进化计算汇刊，第22卷，第6期，第949–960页，2018年12月。[9] G. Zhang、H. Zhang、X. Huang、J. Wang、H. Yu 和 R. Graaf，《针对特定执行器故障的独立驱动后轮轮毂电机电动汽车主动容错控制》，IEEE Trans. Control Syst. Technol.，第24卷，第5期，第1557–1572页，2016年。[10] X. Lang、T. Yang、Z. Wang、C. Wang、S. Bozhko 和 P. Wheeler，《面向多电飞机应用的先进发电中心容错控制》，IEEE Trans. Transp. Electrification，第1页，文章待刊。DOI: dx.doi.org/10.1109/TTE.2021.3093506，2021年。[11] X. Cao、Y. Tian、X. Ji 和 B. Qiu，《制动执行器故障情况下自动驾驶车辆路径跟随的容错控制器设计》，IEEE Trans. Transp. Electrification，第7卷，第4期，第2530–2540页，2021年。[12] M. Seto 和 K. Svendsen，《自主水下航行器：设计与实践》，英国工程技术学会，2020年，第419–445页，第“先进自主水下航行器故障管理”章。[13] J. Kadiyam、A. Parashar、S. Mohan 和 D. Deshmukh，《采用四个可旋转推进器的水下机器人的执行器容错控制研究》，Ocean Eng.，第197卷，第961–979页，2020年。[14] G. Zhu、Y. Ma、Z. Li、R. Malekian 和 M. Sotelo，《具有输入饱和的欠驱动多水面航行器事件触发自适应神经容错控制》，IEEE Trans. Intell. Transp. Syst.，第23卷，第7期，第7045–7057页，2022年。[15] J. Li、Z. Xu、D. Zhu、K. Dong、Z. Z. Tao Yan 和 S. X. Yang，《仿生智能及其在机器人学中的应用：综述》，Intelligence & Robotics，第1卷，第1期，第58–83页，2021年。[16] T. Li、G. Li 和 Q. Zhao，《自适应容错随机形状控制及其在粒子分布控制中的应用》，IEEE Trans. Syst. Man Cybern., Syst.，第45卷，第12期，第1592–1604页，2015年12月。[17] L. Martynova 和 M. Rozengauz，《自主水下航行器运动控制系统重构方法》，《陀螺仪与导航》，第11卷，第3期，第244–253页，2020年。[18] X. Liu、M. Zhang 和 F. Yao，《自主水下航行器自适应容错控制与推进器故障重构》，Ocean Eng.，第155卷，第10–23页，2018年5月。[19] E. Omerdic 和 G. Roberts，《开放式水下航行器的推进器故障诊断与补偿》，Control Eng. Pract.，第12卷，第12期，第1575–1598页，2004年12月。[20] T. Podder、G. Antonelli 和 N. Sarkar，《推进器冗余条件下自主水下航行器的容错控制：仿真与实验》，载于《2000年IEEE国际机器人与自动化会议》（ICRA）论文集，第2卷，美国加利福尼亚州旧金山，2000年，第1251–1256页。[21] S. Zhang、Y. Wu、X. He 和 Z. Liu，《具有输出约束的移动双柔性机械臂协同容错控制》，IEEE Trans. Autom. Sci. Eng.，2021年。[22] C. Shen、Y. Shi 和 B. Buckham，《采用基于李雅普诺夫的模型预测控制的自主水下航行器轨迹跟踪控制》，IEEE Trans. Ind. Electron.，第65卷，第7期，第5796–5805页，2018年。[23] Q. Wen、R. Kumar 和 J. Huang，《最优容错控制综合框架：最大化故障前行为，同时最小化故障后行为》，IEEE Trans. Syst. Man Cybern., Syst.，第44卷，第8期，第1056–1066页，2014年8月。[24] D. Zhu、T. Yan 和 S. X. Yang，《无人水下航行器的运动规划与跟踪控制：技术、挑战与展望》，Intelligence & Robotics，第2卷，第3期，第200–222页，2022年。[25] U. Sinha、M. Cashman 和 M. Cohen，《使用遗传算法优化数据驱动应用中的配置》，载于《第12届国际搜索式软件工程研讨会论文集》，瑞士Cham，2020年，第137–152页。[26] S. Mohanty 和 S. Mohanty，《基于遗传算法的基序搜索问题：综述》，载于《2020年第3届国际智能计算与信息学会议论文集》。智能创新、系统与技术（SIST），第1卷，新加坡新加坡，2020年，第719–731页。[27] D. Gong、J. Sun 和 Z. Miao，《用于区间多目标优化问题的基于集合的遗传算法》，IEEE Trans. Evol. Comput.，第22卷，第1期，第47–60页，2018年2月。[28] C. Lai、H. Ibrahim、M. Abdullah、J. Abdullah、S. Suandi 和 A. Azman，《用于卷积神经网络应用的脑电图数据转换方法文献综述》，载于《2019年国际机器人、视觉、信号处理与电力应用会议论文集：推动可持续发展研究与创新》。电气工程讲义（LNEE），新加坡新加坡，2019年，第521–527页。[29] A. Asmara，《人工神经网络用于预测测试的可能性：系统性文献综述与案例研究》，J. Phys., Conf. Ser.，第1456卷，第12 052–12 061页，2020年。[30] S. Hulyalkar 和 H. Khanuja，《使用贪心算法从评论和微型评论中生成最优解》，载于《2017年国际计算、通信、控制与自动化会议》（ICCUBEA）论文集，印度浦那，2017年，第1–6页。[31] P. Guo、M. Liu 和 Z. Xue，《基于粒子群优化的节能容错静态

云端实时任务调度算法》，载于《第4届国际计算机与通信会议》（ICCC）论文集，中国成都，2018年，第2537–2541页。[32] S. Celtek、A. Durdu 和 M. Ali，《采用群体优化方法的实时交通信号控制》，Measurement，第166卷，第108 206–108 213页，2020年。[33] J. Wu、C. Song、J. Ma、J. Wu 和 G. Han，《强化学习与粒子群优化支持多自主水下航行器的实时救援任务分配》，IEEE Trans. Intell. Transp. Syst.，第23卷，第6807–6820页，2021年7月。[34] D. Zhu、Q. Liu 和 Z. Hu，《基于量子行为粒子群优化的多推进器载人潜水器容错控制算法》，Int. J. Control，第84卷，第11期，第1817–1829页，2011年11月。[35] W. Guo、J. Li、G. Chen、Y. Niu 和 C. Chen，《无线传感器网络中经粒子群优化的实时容错任务分配算法》，IEEE Trans. Parall. Distr. Syst.，第26卷，第12期，第3236–3249页，2015年。[36] S. Saremi、S. Mirjalili 和 A. Lewis，《蚱蜢优化算法：理论与应用》，Adv. Eng. Softw.，第105卷，第30–47页，2017年。[37] G. Wang、A. Heidari、M. Wang、F. Kuang、W. Zhu 和 H. Chen，《混沌弧自适应蚱蜢优化》，IEEE Access，第9卷，第17 672–17 706页，2021年。[38] J. Xia、Z. Wang、D. Yang、R. Li、G. Liang、H. Chen、A. A. Heidari、H. Turabieh、M. Mafarja 和 Z. Pan，《采用对立蚱蜢优化的支持向量机性能优化及其在急性阑尾炎诊断中的应用》，Computers in Biology and Medicine，第143卷，第1页，文章待刊。DOI：dx.doi.org/10.1016/j.compbiomed.2021.105206，2022年。[39] J. Wu、H. Wang、N. Li、P. Yao、Y. Huang、Z. Su 和 Y. Yu，《采用自适应蚱蜢优化算法，在城市环境中对多架太阳能无人机进行目标跟踪的分布式轨迹优化》，Aerosp. Sci. Technol.，第70卷，第497–510页，2017年11月。[40] P. Mishra、V. Goyal 和 A. Shukla，《智能计算与通信进展》，网络与系统讲义。Springer，新加坡，2020年，第179–188页，章节《用于求解数值优化问题的改进蚱蜢优化算法》。[41] W. Durham，《约束控制分配》，J. Guid. Control Dyn.，第16卷，第4期，第717–725页，1993年7月。[42] D. Zhu 和 B. Sun，《基于仿生模型的无人水下航行器混合滑模跟踪控制》，Eng. Appl. Artif. Intell.，第26卷，第10期，第2260–2269页，2013年11月。[43] Y. Shtessel、D. Foreman 和 C. Tournes，《传统滑模控制与二阶滑模控制中的稳定裕度》，载于《2011年IEEE决策与控制会议论文集》，美国佛罗里达州奥兰多，2011年，第4604–4609页。[44] J. Zand，《检查级遥控潜水器的导航增强与脐带缆管理》，硕士论文，机械工程，维多利亚大学，加拿大维多利亚，2005年。[45] D. Zhu、L. Wang、Z. Hu 和 S. X. Yang，《基于蚱蜢优化的多推进器系统载人潜水器容错控制算法》，Ocean Eng.，第242卷，2021年。[46] S. J. Simpson、A. R. Mccaffery 和 B. F. H ̈AGELE，《沙漠蝗相变的行为分析》，Biological Reviews，第74卷，480，2010 4，461no. pp. – 。[47] S. M. Rogers、T. Matheson、E. Despland、T. Dodgson 和 S. J. Simpson，《机械感觉诱导的沙漠蝗（沙漠蝗）行为群居化》，Journal of Experimental Biology，第206卷，4002，2003 22，3991 no. Pt pp. – 。[48] Y. Meraihi、A. Gabis、S. Mirjalili 和 A. Ramdane-Cherif，《蚱蜢优化算法：理论、变体与应用》，IEEE Access，第9卷，第50 001–50 024页，2021年。[49] Y. Zhao、X. Qi、Y. Ma、Z. Li、R. Malekian 和 M. Sotelo，《采用平滑收敛深度强化学习的欠驱动无人水面艇路径跟随优化》，IEEE Trans. Intell. Transp. Syst.，第22卷，第10期，第6208–6220页，2021年。[50] Z. Peng、D. Wang、Z. Chen、X. Hu 和 W. Lan，《不确定动力学下自主水面航行器编队的自适应动态面控制》，IEEE Trans. Control Syst. Technol.，第21卷，第2期，第513–520页，2013年。[51] P. Gong、Z. Yan、W. Zhang 和 J. Tang，《基于不确定动力学下MPC双闭环的自主水下航行器轨迹跟踪控制》，Ocean Eng.，第265卷，2022年。