arXiv:2304.08638v1 [cs.MA] 17 Apr 2023

具有安全保证的深度连续变形协调与优化

Harshvardhan Uppaluru 和 Hossein Rastgoftar

摘要——**本文提出并介绍了一种新颖策略，用于安全协调具有“局部变形”能力的大规模多智能体团队。根据我们提出的方法，多智能体协调被定义为一个多层变形问题，并被表述为深度神经网络（DNN）优化问题。所提出的 DNN 由 p 个隐藏层组成，每个隐藏层都包含代表独特智能体的人工神经元。此外，根据隐藏层 k($k = 1, \cdots, p-1$)中智能体的期望位置，规划隐藏层 $k+1$ 中智能体的期望变形。与现有的神经网络学习问题不同，我们提出的神经网络优化以边界智能体随时间不变的参考位置作为输入，并根据智能体团队构型的期望轨迹训练权重，同时通过上下界约束权重，以确保智能体间避碰。我们通过仿真展示大规模四旋翼无人机团队协调跟踪期望椭圆轨迹的结果，以验证所提出的方法。**

I. 引言

最初受到自然现象的启发，编队飞行和多智能体系统（MAS）中的协同控制在过去 20 年里一直是引人入胜且重要的研究领域。对 MAS 的研究带来了有趣的理论问题，也展现了其在广泛情境中的潜在实际用途。MAS 编队飞行和协同控制可通过集中式或分布式方式实现。集中式技术采用中央计算机控制 MAS 中的每个智能体。然而，分布式技术（又称分布式控制）允许每个智能体进行机载计算，并在相邻智能体之间共享信息 [1]。对于多智能体协同控制，分布式方法具有诸多优点，例如运营成本低、系统要求少、鲁棒性强、适应性高且可扩展性灵活。

A. 相关工作

从监视 [2], [3]到编队飞行 [3], [4], 救援任务 [5], 野生动物监测与探索 [6], 精准农业 [7],协同载荷递送 [8], [9], 以及危险环境感知[10],等应用场景中，已有多种多智能体协调技术得到研究并被提出。

一组智能体作为单个虚拟刚体的粒子，采用虚拟结构的集中式方法

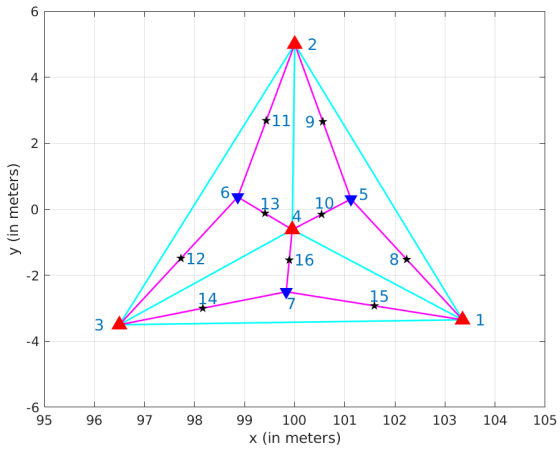
*本研究得到美国国家科学基金会第 2133690 号和第 1914581 号资助。作者来自美国亚利桑那州图森市 85721，亚利桑那大学航空航天与机械工程系可扩展运动与韧性通行实验室。 邮箱： {huppaluru, hrastgoftar}@arizona.edu

（VS） [11], [12]。VS 能够维持智能体之间的刚性几何关系，并作为刚体沿给定方向和朝向运动。一致性 [13], [14] 是研究最为深入的协作控制方法之一。在这种方法中，一组智能体仅通过与邻居通信，就能就某些关注量达成一致。它是一种分布式协调与控制方法，大致分为两类：无领导者一致性（即无领导者的一致性） [15], [16] 以及领导者—跟随者一致性（即有领导者的一致性） [17], [18]。

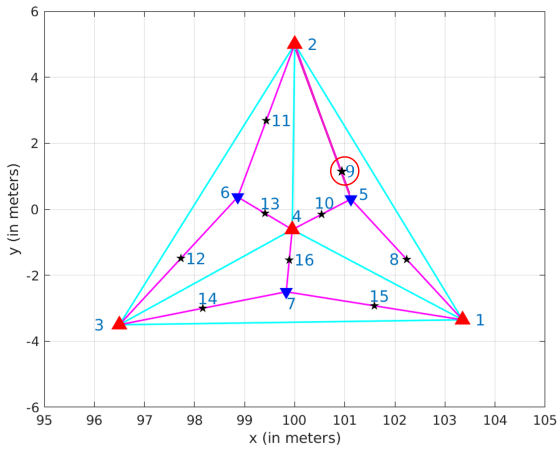
另一种去中心化领导者—跟随者方法称为包容控制 [19], [20], [21], [22],，该方法通过多个领导者实现所有智能体的集体运动。跟随者智能体通过与入邻居智能体进行局部通信来获取期望位置，所有智能体均被限制在由几何约束定义的特定区域内。一种近期的多智能体协调方法称为齐次（或仿射）变换，其原理基于连续介质力学，将系统中的智能体视为可变形体中经历齐次变换的粒子 [23], [24], [25],[26], [27], [28], [29]。该技术确保系统中的所有智能体始终处于包络边界内，并允许包络边界平移、旋转和剪切，同时确保避碰。齐次变换通过确保智能体间避碰，推进了包容控制。为在 n维（$n = 1, 2, 3$）中实现期望的齐次变换， $n+1$ 领导者通过局部通信与跟随者智能体进行通信。

B. 主要贡献

尽管齐次变换协调能够使智能体间距离发生激进且安全的变化，但它存在“变形均匀性”问题。这是因为在任意时刻 t，整个智能体团队构型的变形都由一个雅可比矩阵决定，该矩阵可根据唯一的旋转角、剪切变形角以及轴向变形[30]来指定。因此，整个多智能体系统布局中的旋转、轴向变形和剪切变形必须保持一致。为了解决这一变形均匀性问题，本文将基于单个雅可比矩阵指定的现有齐次变换协调扩展为深度连续变形协调（DCDC），使我们能够规划智能体团队的安全“局部变形”，而无需改变智能体间距离



(a)



(b)

图1。（a） $N = 16$ 四旋翼无人机团队参考构型的一个示例。如第 II 节所定义，我们有 $\mathcal{B} = \{1, 2, 3\}$、$\mathcal{C} = \{4\}$ 和 $\mathcal{I} = \{5, \cdots, 16\}$。集合 $\mathcal{L}_1 = \{1, 2, 3, 4\}$ 由红色三角形表示，$\mathcal{L}_2 = \mathcal{L}_1 \cup \{5, 6, 7\}$ 由蓝色三角形表示，$\mathcal{L}_3 = \{8, \cdots, 16\}$ 由黑色星形表示。最后，$\mathcal{V} = \{1, \cdots, 16\}$ 是用于标识团队中所有四旋翼无人机的集合。（b）通过调整权重 $\beta_{9,\ 5,\ 3}$ 和 $\beta_{9,\ 2,\ 3}$，使智能体 9 发生局部变形。

所有智能体（见图1(a)、(b)）。

DCDC 将多智能体协调定义为一种通过神经网络构建的多层连续体变形，其中 (i) 每个神经元代表一个不同的智能体，(ii) 输入层接收边界智能体随时间不变的参考位置。在我们提出的神经网络中，该网络具有 p 个隐藏层，第 $k+1$ 层智能体的期望变形基于属于隐藏层 k 的智能体的期望位置（$k = 1,\ \cdots,\ p-1$）。不同于基于训练数据确定权重和偏置的现有神经网络学习方法，我们提出的神经网络优化基于单个输入和随时间变化的输出来分配权重和偏置。具体而言，我们的目标是通过获取权重和偏置来规划多智能体团队的变形隐藏层，使智能体团队的实际位置构型与期望位置构型之间的误差最小化，同时保证任意两个智能体之间的智能体间避碰。我们通过获得隐藏层权重和偏置的等式约束与不等式约束，给出了确保智能体间避碰的保证条件。

本文组织如下：首先介绍预备知识（第 II 节），随后详细描述我们提出的方法（第 III 节）。在介绍仿真结果（第五节）之前，我们先给出安全保证条件（第四节）。第六节总结全文。

II. 预备知识

考虑一个由 N 个智能体组成的团队（见图1），它们作为可变形体的粒子，在三维运动空间中协同运动。我们用集合 $\mathcal{V} = \{1, \cdots, N\}$ 表示团队中的所有智能体。集合 $\mathcal{V}$ 表示为

$$\mathcal{V} = \mathcal{B} \bigcup \mathcal{C} \bigcup \mathcal{I}, \tag{1}$$

其中，$\mathcal{B}$ 表示位于智能体团队构型边界上的主要领导智能体，单元素集合 $\mathcal{C}$ 表示位于由 $\mathcal{B}$ 定义的凸包内部的单个智能体，而 $\mathcal{I}$ 表示其余智能体，它们位于由 $\mathcal{B}$ 定义的凸包内部。不失一般性，我们对智能体进行编号，使集合 $\mathcal{B} = \{1, \cdots, N_L - 1\}$、$\mathcal{C} = \{N_L\}$ 和 $\mathcal{I} = \{N_L + 1, \cdots, N\}$ 分别表示边界智能体、核心智能体和内部智能体。集合 $\mathcal{V}$ 也可表示为

$$\mathcal{V} = \bigcup_{k=1}^{p} \mathcal{L}_k \tag{2}$$

子集 $\mathcal{L}_1$ 至 $\mathcal{L}_p$，其中 $\mathcal{L}_1 = \mathcal{B} \bigcup \mathcal{C}$。我们注意到，子集 $\mathcal{L}_k$ 是 $\mathcal{L}_{k+1}$ 的直接领导者，对于 $k = 1,\ \cdots,\ p-1$，这意味着，属于 $\mathcal{L}_{k+1}$ 的智能体的期望位置是根据集合 $\mathcal{L}_k$ 中智能体的期望位置定义的。$\mathcal{L}_2$ 至 $\mathcal{L}_{p-1}$ 定义为内部领导者，而集合 $\mathcal{L}_p$ 定义了纯跟随者；它们不作为属于 $\mathcal{L}_1$ 至 $\mathcal{L}_{p-1}$ 中任何智能体的直接领导者。此外，$\mathcal{L}_1$ 至 $\mathcal{L}_{p-1}$ 满足以下条件：

$$\bigwedge_{k=1}^{p-1} (\mathcal{L}_k \subset \mathcal{L}_{k+1}). \tag{3}$$

智能体 $i \in \mathcal{L}_{k+1}$ 的期望位置具体定义为

$$\mathbf{p}_i(t) = \sum_{j \in \mathcal{L}_k} \beta_{i,j,k}(t) \mathbf{p}_j(t), \qquad i \in \mathcal{L}_{k+1},\ k = 1, \cdots, p-1, \tag{4}$$

其中 $\beta_{i,\ j,\ k}(t) \in [0, 1]$，并且

$$\sum_{j \in \mathcal{L}_k} \beta_{i,j,k}(t) = 1, \qquad i \in \mathcal{L}_{k+1},\ k = 1, \cdots, p-1. \tag{5}$$

安全条件将在第四节中说明。请注意，属于 $\mathcal{L}_1$ 的领导者是主领导者，它们会按照定义的期望轨迹独立移动

$$\mathbf{p}_i(t) = \alpha_i(t)\mathbf{a}_i + \mathbf{d}(t), \qquad \forall i \in \mathcal{L}_1, \tag{6}$$

其中，$\mathbf{d}(t)$表示智能体团队构型在惯性坐标系中的标称位置，$\mathbf{a}_i$ 是主要领导者 $i \in \mathcal{L}_1$的常数参考位置。注意，核心领导者 $i \in \mathcal{C}$ 的参考位置为$\mathbf{0}$，即$\mathbf{a}_{i,0} = \mathbf{0}$。

III. 方法

我们研究由 N个智能体组成的团队在有限时间区间 $[t_0, t_f]$内的深度连续体变形协调，并提出一种基于神经网络（NN）的优化方法，通过指定第一层的 α_i(t)，以及每一层 $k = 1, \cdots, p-1$的$\beta_{i,j,k}$(t)，获得智能体团队的期望变形（见图2）。图2展示了所提出的具有 p 个隐藏层的神经网络示意图。请注意，隐藏层1至 p 中的人工神经元分别表示由 $\mathcal{L}_1$ 至 $\mathcal{L}_p$定义的智能体。

输入层：如图2所示，输入层生成边界智能体和核心智能体的参考位置，定义为 $\mathcal{B} \cup \mathcal{C}$。请注意，参考位置不随时间变化。

隐藏层：第一个隐藏层（记为 $\mathcal{L}_1$）接收边界智能体的参考位置。具体而言，神经元 $j \in \mathcal{L}_1$ 表示智能体 $j \in \mathcal{B} \cup \mathcal{C}$，接收参考位置 $\mathbf{a}j$，并使用方程 (6) 返回 $\mathbf{p}j(t)$。请注意，方程 (6) 中使用的刚体位移向量 $\mathbf{d}(t)$ 对层$\mathcal{L}_1$中的所有神经元具有相同的偏置。

每个隐藏层 $k \in \{2, \cdots, p\}$ 从前一个隐藏层 $k-1$ 接收期望位置，并返回由 $\mathcal{L}_k$定义的智能体的期望位置。具体而言，神经元 $i \in \mathcal{L}_k$ 接收$\mathcal{L}_{k-1}$ 中智能体的期望位置，并使用 (4) 返回 $\mathbf{p}_i(t)$。请注意，第 2 层至第 p 层隐藏层中的神经元偏置为 $\mathbf{0}$。

输出层：输出层由单个神经元组成，用于对智能体团队配置的期望位置取平均值。因此，连接隐藏层 p 中的神经元与输出层的边所关联的权重为 $\frac{1}{|\mathcal{L}_p|}$，其中 $|\mathcal{L}_p|$ 表示集合 $\mathcal{L}_p$的基数。

损失函数：本文使用残差平方和作为时刻 t 的损失函数，由下式给出：

$$\mathbf{Loss}(t) = \left\| \frac{1}{|\mathcal{L}_p|} \sum_{i \in \mathcal{L}_p} \mathbf{p}_i(t) - \mathbf{d}(t) \right\|^2, \qquad t \in [t_0, t_f]. \tag{7}$$

神经网络训练：为了训练神经网络，我们在任意时刻 $t \in [t_0, t_f]$ 确定正权重参数 $\alpha_i(t)$ 和 $\beta_{i,j,k}(t)$。为确保智能体团队协同的安全性，$\alpha_i(t)$ 和 $\beta_{i,j,k}(t)$ 受特定上下界的约束，这些上下界在第四节中确定。

IV. 安全保证条件

在继续之前，我们分别将智能体 $i \in \mathcal{V}$ 的期望位置、实际位置和参考位置表示为 $\mathbf{p}_{[x_{i,d}(t)\ y_{i,d}(t)\ z_{i,d}(t)]^T}$，$\mathbf{r}_i(t) = [x_i(t)\ y_i(t)\ z_i(t)]^T$，以及 $\mathbf{a}_{i,0} = [x_{i,0}\ y_{i,0}\ z_{i,0}]^T$。为得到安全保证条件，我们作出以下假设：

假设 1。每个智能体 $i \in \mathcal{V}$ 都可以执行适当的轨迹跟踪控制，使实际轨迹 $\mathbf{r}_i(t)$ 与期望轨迹 $\mathbf{p}_i(t)$ 之间的偏差保持有界，并满足以下方程

$$\bigwedge_{i \in \mathcal{V}} \bigwedge_{q \in \{x,y,z\}} |q_{i,d}(t) - q_i(t)| \le \delta, \qquad \forall t, \tag{8}$$

其中 $\bigwedge_{i \in \mathcal{V}}$ 表示“包含所有”，$\delta > 0$ 为常数。

假设 2。每个智能体 $i \in \mathcal{V}$ 都可以被边长为 2ϵ的盒子包围。

为确保智能体团队连续体变形的安全性，必须满足以下两项安全要求：

安全条件1。对于任意两个不同的智能体 i 和 j，若满足以下条件，则可保证智能体间避碰

$$\bigwedge_{i=1}^{N-1} \bigwedge_{j=i+1}^{N} \bigwedge_{q \in \{x,y,z\}} |q_i(t) - q_j(t)| > 2\epsilon, \qquad \forall t, \tag{9}$$

安全条件2。每个跟随者 $i \in \mathcal{L}_{k+1}$ 都位于由 $\mathcal{L}_k$定义的凸包内。若第 $\mathcal{L}_{k+1}$层的智能体包含在由 $\mathcal{L}_k$定义的凸包内，则称这些智能体满足该条件

$$\bigwedge_{k=1}^{p-1} \bigwedge_{i \in \mathcal{L}_{k+1}} \bigwedge_{j \in \mathcal{L}_k} (\beta_{i,j,k}(t) \ge 0), \qquad \forall t. \tag{10a}$$

$$\bigwedge_{k=1}^{p-1} \bigwedge_{i \in \mathcal{L}_{k+1}} \left(\sum_{j \in \mathcal{L}_k} \beta_{i,j,k}(t) = 1 \right), \qquad \forall t. \tag{10b}$$

若满足 [30]，则安全条件1成立

$$\bigwedge_{i=1}^{N-1} \bigwedge_{j=i+1}^{N} \bigwedge_{q \in \{x,y,z\}} |q_{i,d}(t) - q_{j,d}(t)| > 2(\delta + \epsilon), \qquad \forall t. \tag{11}$$

请注意，方程 (11) 为智能体间避碰提供了一个充分的安全条件。

为确保安全条件1和2成立，我们通过以下条件约束$\alpha_{i,k}$ 和 $\beta_{i,j,k}$

$$\bigwedge_{i \in \mathcal{L}_1} (\alpha_{min} \le \alpha_i(t)), \qquad \forall t \tag{12}$$

$$\bigwedge_{k=1}^{p-1} \bigwedge_{j \in \mathcal{L}_k} \bigwedge_{i \in \mathcal{L}_{k+1}} (\beta_{k,min} \le \beta_{i,j,k} \le \beta_{k,max}). \tag{13}$$

我们首先为每一层 $k = 1, \cdots, p-1$指定 $\beta_{k,min}$ 和$\beta_{k,max}$。然后，对于每一层 $k = 1, \cdots, p-1$给定的 $\beta_{k,min}$和 $\beta_{k,max}$，我们通过求解以下优化问题得到 α_{min}：

$$\alpha_{min} = \min_{\alpha \in (0,1)} \alpha \tag{14}$$

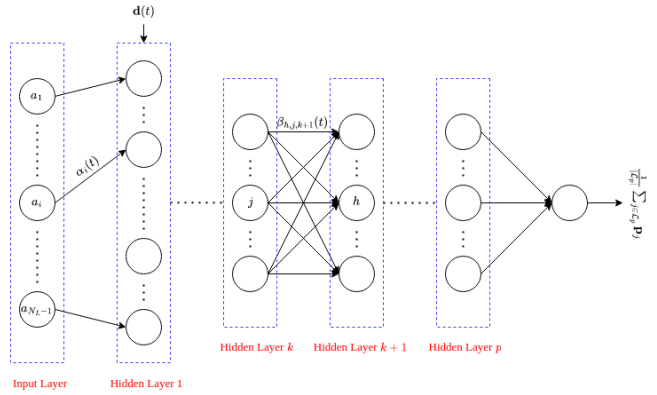


图2. 所提出的用于 MQS 连续体变形优化的神经网络结构。

满足

$$\bigwedge_{i\in\mathcal{L}_1}\bigwedge_{q\in\{x,y,z\}}\left(q_{i,d}=\alpha q_{i,0}\right), \tag{15}$$

$$\bigwedge_{k=1}^{p-1}\bigwedge_{i\in\mathcal{L}_{k+1}}\bigwedge_{q\in\{x,y,z\}}\left(q_{i,d}=\sum_{j\in\mathcal{L}_k}\beta_{i,j,k}q_j\right), \tag{16}$$

$$\bigwedge_{k=1}^{p-1}\bigwedge_{\substack{i,h\in\mathcal{L}_{k+1}\\ i\neq h}}\bigwedge_{q\in\{x,y,z\}}\left(\sum_{j\in\mathcal{L}_k}\left|\left(\beta_{i,j,k}-\beta_{h,j,k}\right)q_{j,d}\right|>2\left(\delta+\epsilon\right)\right),$$
$$\forall\beta_{i,j,k},\beta_{h,j,k}\in[\beta_{k,min},\beta_{k,max}]. \tag{17}$$

我们使用算法1求解上述优化问题。

算法1 获取深度连续体变形协调与优化的安全保证条件

获取： $\beta_{k}{}_{min}$ 和 $\beta_{k}{}_{,max}$ 对于 $k=1,\cdots,p-1$、$\Delta\alpha$，以及每个智能体的参考位置 $i\in\mathcal{V}$
获取： α_{min}
设置： $\alpha=1$
当 方程 (17) 满足**时**
    $\alpha\leftarrow\alpha-\Delta\alpha$
    使用方程 (15) 更新智能体 $i\in\mathcal{L}_1$ 的 $\mathbf{p}_i(t)$
    对于 $k\in\{1,\cdots,p-1\}$ **执行**
        对于 $j\in\mathcal{L}_k$ **执行**
            对于 $q\in\{x,y,z\}$ **执行**
                对于 $\beta_{i,j,k}\in[\beta_{k,min},\beta_{k,max}]$ **做**
                    使用方程（16）更新$\mathbf{p}_i(t)$。
                    检查约束条件（17）。
                结束循环
            结束循环
        结束循环
    结束循环
结束 while
$\alpha_{min}\leftarrow\alpha$

参数	值	单位
m	0.5	kg
g	9.81	$\mathrm{m/s^2}$
l	0.25	m
I_r	3.357×10^{-5}	$\mathrm{kgm^2}$
I_x	0.0196	$\mathrm{kgm^2}$
I_y	0.0196	$\mathrm{kgm^2}$
I_z	0.0264	$\mathrm{kgm^2}$
b	3×10^{-5}	$\mathrm{Ns^2/rad^2}$
k	1.1×10^{-6}	$\mathrm{Ns^2/弧度^2}$

表 I
四旋翼无人机规格

V. 仿真结果

这里，我们展示了使用 PyTorch[1] 在一台运行 Ubuntu 20.04 LTS 的台式机上获得的仿真结果。该电脑配备第 11 代 Intel i7 CPU、NVIDIA GPU 和 16 GB 内存。我们研究多四旋翼无人机系统（MQS）的演化，该系统由 $N=16$ 架四旋翼无人机组成，并在时间 $t=0$时采用初始编队，如图 1a 所示。MQS 中的四旋翼无人机型号相同，并采用 [31]中建立的动力学模型。 [32] 中给出的四旋翼无人机参数列于表 I 中。

我们将 MQS 配置的期望路径设为椭圆，如图 3 所示，其长半轴半径为 $a=100$m，短半轴半径为 $b=80$m。完成椭圆路径的总行程时间记为 $T=60$s。MQS 分布在高度为 $z=10$m 的水平面上，该高度始终保持常数。

MQS 中四旋翼无人机通过定义集合 $\mathcal{V}=\mathcal{B}\bigcup\mathcal{C}\bigcup\mathcal{I}$ 来标识，其中 $\mathcal{B}=\{1,2,3\}$、 $\mathcal{C}=\{4\}$ 和 $\mathcal{I}=\{5,\cdots,16\}$。或者，如第 II 节所述，集合 $\mathcal{L}_1=\{1,\cdots,4\}$、 $\mathcal{L}_2=\{5,\cdots,7\}$ 和 $\mathcal{L}_3=\{8,\cdots,16\}$ 分别标识隐藏层 1、2 和 3 中的四旋翼无人机。我们采用第 III 节提出的方法计算 α_i 和 $\beta_{i,\ j,\ k}$ 参数。表 II 展示了通信权重的最小值和最大值，α_i (*t*) 和 $\beta_{i,\ j,\ k}$(*t*)。需要注意的是，最小值的设定确保了 MQS 中任意两架四旋翼无人机之间的智能体间避碰，同时满足第四节中提到的约束。

表 II 获取神经网络权重的上下界

α_{min}	$\beta_{1,min}$	$\beta_{2,min}$	$\beta_{1,max}$	$\beta_{2,max}$
0.5	0.2	0.35	0.6	0.65

我们希望四旋翼无人机的构型在满足表 II 所示安全条件的同时，沿着期望的椭圆轨迹运动。因此，我们按照第 III 节所述运行神经网络 6000 个训练轮次，学习率设为 0.01。采用随机梯度下降 (SGD)，并使用第 III 节中提到的损失函数。神经网络的权重和偏置用于计算

[1]https://pytorch.org/

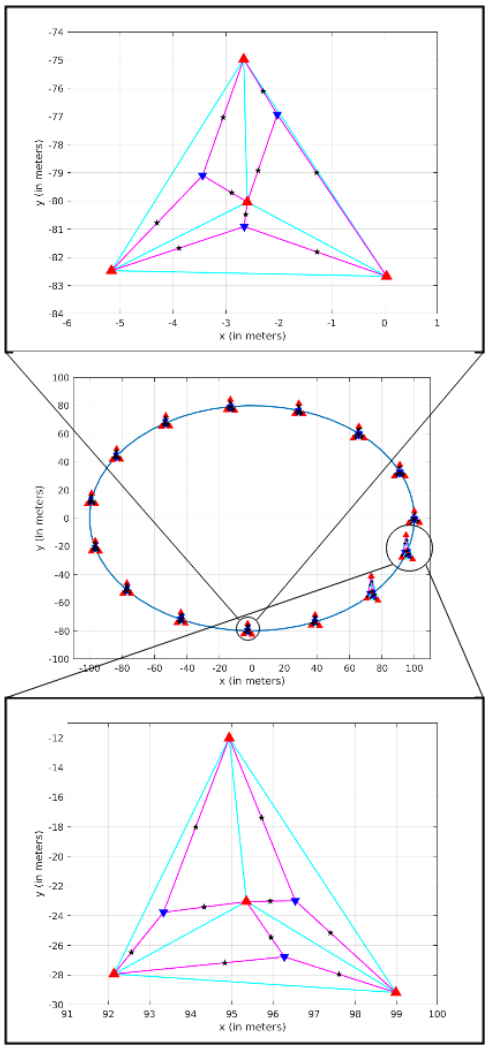


图 3. MQS 在不同时刻的期望构型。

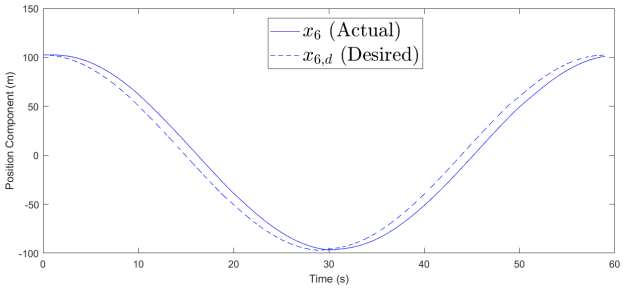


图 4. 四旋翼无人机 6 的实际轨迹和期望轨迹的 x 分量。

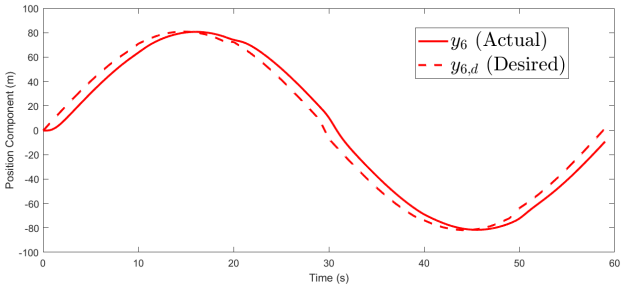


图 5. 四旋翼无人机 6 的实际轨迹和期望轨迹的 y 分量。

MQS 编队中每架四旋翼无人机的标称轨迹。采用输入输出反馈线性化控制方法 [31]，图 4、5、6 和 7 分别绘出了四旋翼无人机 $6 \in \mathcal{L}_2$ 的 x-位置分量、y-位置分量、推力大小和旋翼角速度。

VI. 结论

本文提出并介绍了一种基于神经网络的技术，用于在三维运动空间中跟踪已知目标轨迹时，对 N 架四旋翼无人机团队进行安全的连续变形协调与优化。假设跟踪期望轨迹时不存在障碍物，且 MQS 中没有故障，我们提出了一种新算法：以常数参考构型作为神经网络的输入，并根据待跟踪的期望轨迹进行优化，从而获得与齐次变换中的矩阵直接相关的权重。我们的工作还提供了安全保证，确保智能体间避碰。未来的工作方向包括提高算法在存在环境障碍物时的鲁棒性，并开展进一步的仿真和有障碍物环境下的飞行实验，并将流体飞行导航 [25], [29], [33] 纳入避障算法。另一个方向是研究 MQS 编队通过已知障碍物（例如不同形状和尺寸的窗户）时的变形。

参考文献

[1] Y. Cao、W. Yu、W. Ren 和 G. Chen，《分布式多智能体协调研究的最新进展综述》，*IEEE*控制系统技术汇刊，第9卷，438，2012 1，427，第–页。[2] S.-L. Du、X.-M. Sun、M. Cao 和 W. Wang，《在多智能体监视网络中通过协作中继追捕逃逸者》，*Automatica*，第83卷，第155–161页，2017年。[3] L. C. B. Da Silva、R. M. Bernardo、H. A. De Oliveira 和 P. F. Rosa，《具有动态优先级的持续监视多无人机基于智能体的协调》，载于*2017*年国际军事技术会议（*ICMT*）。IEEE，2017年，第765–771页。[4] S.-h. Ha 和 S.-d. Chi，《面向群集与编队飞行的自主无人机多智能体设计》，《先进导航技术期刊》，第21卷，第6期，第521–528页，2017年。[5] H. AlTair、T. Taha、J. Dias 和 M. Al-Qutayri，《多目标多智能体搜救任务的决策制定》，载于国际电气与计算技术及应用会议（*ICECTA*）。IEEE，2017年，第1–4页。[6] J. Witczuk、S. Pagacz、A. Zmarz 和 M. Cypel，《探索无人飞行器和热成像技术用于森林有蹄类动物调查的可行性——初步结果》，《国际遥感杂志》，第39卷，第15–16期，第5504–5521页，2018年。[7] D. C. Tsouros、S. Bibi 和 P. G. Sarigiannidis，《基于无人机的精准农业应用综述》，《*Information*》，第10卷，第11期，第349页，2019年。[8] H. Rastgoftar 和 E. M. Atkins，《协作空中举升与操控（CALM）》，《航空航天科学与技术》，第82卷，118，2018 105，第–页。

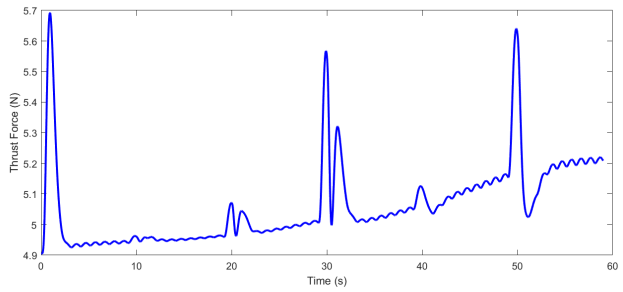


图 6. 四旋翼无人机 6 的推力。

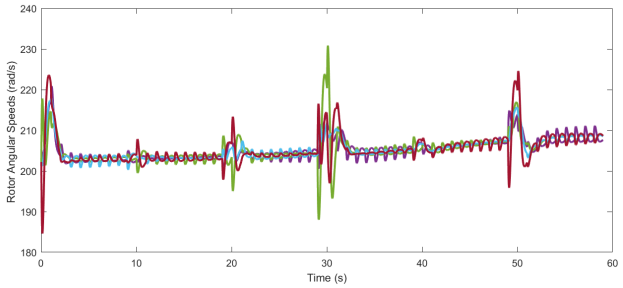


图 7. 四旋翼无人机 6 的旋翼角速度。

[9] F. Rossomando、C. Rosales、J. Gimenez、L. Salinas、C. Soria、M. Sarcinelli-Filho 和 R. Carelli，《基于运动学控制器和神经滑模动态补偿的多架四旋翼无人机空中载荷运输》，智能与机器人系统期刊，第100卷，第2期，第519–530页，2020年。[10] B. Argrow、D. Lawrence 和 E. Rasmussen，《用于危险环境中的传感器部署、遥测和可视化的无人机系统》，载于第*43*届*AIAA*航空航天科学会议暨展览会，2005年，第1237页。[11] M. A. Lewis 和 K.-H. Tan，《利用虚拟结构实现移动机器人的高精度编队控制》，《自主机器人》，第4卷，第4期，第387–403页，1997年。[12] W. Ren 和 R. Beard，《基于虚拟结构并带有编队反馈的航天器编队控制》，载于*AIAA*制导、导航与控制会议暨展览会，2002年，第4963页。[13] W. Ren 和 R. W. Beard，《交互拓扑动态变化下多智能体系统的一致性寻求》，*IEEE*自动控制汇刊，第50卷，第5期，第655–661页，2005年。[14] L. Ding、Q.-L. Han 和 G. Guo，《分布式多智能体系统中基于网络的领导者跟随一致性》，*Automatica*，第49卷，第7期，第2281–2286页，2013年。[15] S. Rao 和 D. Ghose，《用于无人飞行器无领导者一致性的基于滑模控制的自动驾驶仪》，*IEEE*控制系统技术汇刊，第22卷，第5期，第1964–1972页，2013年。[16] W. Ren，《网络化欧拉–拉格朗日系统的分布式无领导者一致性算法》，《国际控制杂志》，第82卷，第11期，第2137–2149页，2009年。[17] Q. Song、J. Cao 和 W. Yu，《通过钉扎控制实现非线性多智能体系统的二阶领导者跟随一致性》，《系统与控制快报》，第59卷，第9期，第553–562页，2010年。[18] L. Xu、H. Fan、Z. Dong 和 W. Wang，《切换拓扑下领导者跟随多智能体系统的鲁棒一致性控制》，载于*2016*年控制论、机器人与控制国际会议（*CRC*）。IEEE，2016年，第27–31页。[19] G. Ferrari-Trecate、M. Egerstedt、A. Buffa 和 M. Ji，《拉普拉斯羊群：一种基于领导者的包容控制混合式走停策略》，载于混合系统：计算与控制国际研讨会。施普林格，2006年，第212–226页。[20] Z. Li、W. Ren、X. Liu 和 M. Fu，《存在多个领导者时具有一般线性动力学的多智能体系统的分布式围捕控制》，《国际鲁棒与非线性控制杂志》，第23卷，第5期，第534–547页，2013年。[21] X. Wang、S. Li 和 P. Shi，《分布式有限时间围捕双积分器多智能体系统的控制，”*IEEE*《控制论汇刊》，第 44 卷，第 9 期，第 1518–1528 页，2013 年。[22] G. Wen 、 Y. Zhao 、 Z. Duan 、 W. Yu 和 G. Chen，“高阶多领导者多智能体系统的围捕：一种动态输出方法，”*IEEE*《自动控制汇刊》，第 61 卷，第 4 期，第 1135–1140 页，2015 年。[23] H. Rastgoftar 和 S. Jayasuriya，“多智能体系统作为连续体的演化，”《动态系统、测量与控制期刊》，第 136 卷，第 4 期，页码 041014，2014 年。[24] H. Rastgoftar，《多智能体系统的连续体变形》。施普林格，2016 年。[25] H. Emadi、H. Uppaluru 和 H. Rastgoftar，“一种用于容错多四旋翼协调的基于物理的安全恢复方法，”*arXiv* 预印本 *arXiv:2110.07777*，2021 年。[26] H. Emadi、H. Uppaluru、H. Ashrafiuon 和 H. Rastgoftar，“利用协同定位实现多四旋翼无人机系统的无碰撞连续变形协调，”载于 *2022*年欧洲控制会议 *(ECC)*。IEEE，2022 年，第 1349–1354 页。[27] H. Emadi、H. Uppaluru 和 H. Rastgoftar，“一种用于容错多四旋翼协调的基于物理的安全恢复方法，”载于 *2022*年美国控制会议 *(ACC)*。IEEE，2022 年，第 2527–2532 页。[28] M. Romano、P. Kuevor、D. Lukacs、O. Marshall、M. Stevens、H. Rastgoftar、J. Cutler 和 E. Atkins，“五架四旋翼飞行器团队的连续体变形实验评估，”载于 *2019*年美国控制会议 *(ACC)*。IEEE，2019 年，第 2023–2029 页。[29] H. Uppaluru、H. Emadi 和 H. Rastgoftar，“利用协同定位实现具有韧性的多无人航空系统协调，”《航空航天科学与技术》，第 131 卷，页码 107960，2022 年。 [在线]。网址：https: // www.sciencedirect.com/science/article/pii/S1270963822006344 [30] H. Rastgoftar 和 I. V. Kolmanovsky，“基于安全仿射变换的大规模多四旋翼系统制导，”*IEEE*《网络系统控制汇刊》，第 8 卷，653，2021 2，640 no. pp. – 。[31] H. Rastgoftar，“大规模多四旋翼系统 (MQS) 的实时部署，”*arXiv* 预印本 *arXiv:2201.10509*，2022 年。[32] E. Gopalakrishnan，“四旋翼飞行力学模型与控制算法，”捷克技术大学，第 69 卷，2017 年。[33] M. Romano、H. Uppaluru、H. Rastgoftar 和 E. Atkins，“通过流体流动导航函数实现能够应对飞行器突发故障的四旋翼编队飞行，”*arXiv* 预印本 *arXiv:2203.01807*，2022 年。