

VANET 随机部署环境下基于改进型共享最近邻密度峰聚类的快速分簇算法

陈靖宇, 徐志林

(广东工业大学 计算机学院, 广州 510006)

摘要: 针对车辆高速移动场景下, 网络拓扑变化过大导致网络分簇结果不稳定的问题, 提出一种基于改进型共享最近邻密度峰聚类的快速成簇算法 SNNCA (shared nearest neighbor clustering algorithm); 通过综合考虑节点的链路生存周期和移动相似性, 提出一种全新的节点连接稳定程度评估指标, 并将该评估指标应用于节点共享最近邻的计算过程, 以组织网络节点为划分合理的多跳簇结构; 为适应网络环境的动态变化, 提出一种簇维护策略, 其中每个层级的簇成员承担着维护下一层级簇成员的任务, 该策略能够对簇成员进行批量分离或合并, 从而实现了算法的分布式快速收敛; 根据随机部署场景中进行的仿真实验结果显示, 相比其他较新算法, SNNCA 算法降低了 74% 的簇数量, 并且簇成员的平均存活时间增加了近 1 倍, 表现出更好的网络稳定性和健壮性。

关键词: 车载自组织网络; 快速分簇算法; 共享最近邻; 密度峰聚类; 随机部署场景; 多跳簇结构

Fast Clustering Algorithm Using Improved Shared-Nearest-Neighbor-based Density Peaks Clustering in Random Deployment Environment of VANET

CHEN Jingyu, XU Zhilin

(School of Computing, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: For the issue of unstable network clustering results due to excessive network topology changes in high-speed vehicle movement scenarios, a fast clustering algorithm for shared nearest neighbor clustering algorithm (SNNCA) based on the improved shared-nearest-neighbor-based density peak clustering is proposed. By comprehensively considering the node's link survival period and movement similarity, a novel node connection stability evaluation metric is proposed. The metric is utilized in the shared nearest neighbor calculation process of the node to organize the network nodes into reasonable multi-hop cluster structure. To adapt to the dynamic changes in the network environment, a cluster maintenance strategy is introduced, where each level of cluster members takes on the task of maintaining the next level of cluster members, and this strategy can perform the batch separation or merging of cluster members, and realize the distributed and rapid convergence of the algorithm. The simulation results of the random deployment scenario show that compared to other newer algorithms, the SNNCA algorithm reduces 74% of cluster numbers, and the average survival time of cluster members has nearly doubled, demonstrating better stability and robustness of the network.

Keywords: vehicular ad hoc network (VANET); fast clustering algorithm; shared nearest neighbor; density peak clustering; random deployment scenario; multi-hop cluster

0 引言

随着车辆数量的快速增长, 城市交通拥堵和交通安全的问题严峻, 智能交通系统 (ITS, intelligent transportation system) 在这样的背景下产生^[1]。ITS 提供的服务包括城市交通智能调度^[2], 辅助驾驶^[3], 交通预警等。ITS 的应用能够有效缓解交通拥堵、减少交通事故发生, 提高交通安全性^[4]。车载自组织网络 (VANET, vehicular ad hoc network) 作为 ITS 的核心部分, 在实现有效的路侧通信和数据传输过程中发挥着重要作用^[5-6]。车载自组织网络

是一种分布式的无线网络系统, 该网络系统由网络节点间的动态链接形成, 不需要依赖既存的网络架构。VANET 中的节点移动速率较大导致网络拓扑变化较大^[7], 采用分簇机制能够有效实现网络的快速部署及动态重建, 提高网络的可扩展性^[8]。分簇是指对 VANET 中的网络节点进行分组, 每个组称为一个簇。每个簇至少拥有一个簇首 (CH, cluster head), 簇中的其余节点为簇成员 (CM, cluster member)。簇首能够协调簇成员进行簇内及簇间通信。根据 CH 与 CM 间的距离, 分簇算法可以分为单跳分簇和多跳分

收稿日期: 2023-03-06; 修回日期: 2023-04-10。

作者简介: 陈靖宇 (1973-), 男, 高级实验师。

通讯作者: 徐志林 (1993-), 男, 硕士。

引用格式: 陈靖宇, 徐志林. VANET 随机部署环境下基于改进型共享最近邻密度峰聚类的快速分簇算法[J]. 计算机测量与控制, 2023, 31(9): 174-182.

簇。单跳分簇中的 CM 节点处在 CH 的一跳通信范围, CM 与 CH 能够直接通信, 如文献 [9-10]。在多跳分簇中, 由于 CM 节点不一定处在 CH 的一跳通信范围, 因此 CM 不一定直接与 CH 进行通信, 而是通过其他簇成员进行转发来实现与簇首的通信, 如文献 [11-12]。VANET 分簇算法一般包括簇首选取、成簇形成和簇维护三个过程。目前各学者在 VANET 分簇算法上已经做了大量的工作, 主要围绕分簇算法的三个基本过程进行研究, 目的是提高成簇效率及簇的稳定性^[13]。

文献 [14] 以节点的移动速度、相对位置为依据, 提出了一种移动性指标计算方法。该算法规定网络通信的最大跳数为 D 跳, 节点以移动性指标和节点连通度作为簇首选择的依据, 从其 D 跳邻居中选择符合条件的节点作为备用簇首, 并维护一个长度为 D 的备用簇首列表。节点与原簇首通信中断时优先使用备用簇首, 从而降低成簇开销。该方法通过备用簇首的策略在一定程度上降低了簇重构的频率, 但由于备用簇首的选择是在成簇阶段完成的, 并未考虑节点运动过程中网络拓扑变化导致的备用簇首失效问题, 因此该方法仍存在簇重构频率较大的问题。文献 [15] 通过判断节点所处区域的节点密度大小, 使处于密度中心的节点向邻居节点发出入簇邀请, 从而将处于密度较高区域的节点分为同一个簇。此外该方法以节点的相对位置、速度和加速度为依据提出了一个移动性指标, 用于簇首选择。当簇首离开簇时, 簇内成员将根据当前节点的移动性指标推选出新的簇首。该方法能够降低簇重构的频率, 改善在高速移动场景下节点状态变化频率较大的问题。但由于该方法仅根据节点的空间位置进行簇的划分, 未考虑节点间通信的稳定性, 因此节点状态变化频率较大的问题仍然存在。文献 [16] 提出了一种双簇首策略, 该方法在推选出主簇首后, 由主簇首在其簇成员中选择一个合适的节点作为副簇首。当簇成员与主簇首的通信中断时, 由副簇首暂时承担主簇首的功能。该方法以节点间的相对位置、相对速度、平均链路寿命、平均信号质量为依据提出一种合适性指标, 将合适性指标值的较大者作为簇首。文献 [17] 以节点与邻居节点间的平均距离、平均相对速度、平均相对加速度为依据, 提出一种节点分数指标, 将分数较高者推选为簇首。该指标考虑了节点与邻居节点间的移动相关性, 能够选择移动较稳定的节点作为簇首, 提高了簇首的稳定性。该方法在完成簇首推选后, 会周期性地在其簇成员中推选一个备用簇首; 当原簇首离开簇时, 备用簇首将成为新的簇首。文献 [18] 以节点间的通信质量、节点连接寿命、一跳邻居数量为依据, 提出一种节点等级指标, 将节点通信范围内等级最高的节点推选为簇首, 簇成员中等级最高的节点作为备用簇首。文献 [15-18] 均采用了由簇首选择备用簇首这一策略来降低簇重构发生的频率, 但并未考虑簇首和备用簇首同时离开簇的情况, 以及簇首在选择备用簇首前就离开簇的情况。此外在文献 [15-18] 的分簇方法中, 簇首只能在本节点的通信范围内进行选择,

簇的规模被节点通信范围限制, 因此以上方法的成簇规模较小, 生成的簇数量较多。

针对以上问题, 提出一种基于改进的共享最近邻密度峰聚类^[19]的 VANET 分簇算法 (SNNCA, shared nearest neighbor clustering algorithm), 详细的创新点概括如下:

1) 将共享最近邻的算法思想应用于 VANET 分簇过程中, 以两节点的共同邻居数量作为节点分簇的依据。该分簇方法能够考虑节点间邻居网络的相似程度, 从而达到延长网络稳定周期的目的。

2) 针对节点间连接稳定性的问题, 设计了一种基于节点链路生存周期和移动相似性的综合评估方法, 并将该评估方法应用于节点的共享最近邻的计算, 以得到划分合理的分簇结果。

3) 提出了一种多层级的簇成员控制机制, 其中每个层级的簇成员担任着维护下一层级簇成员的任务, 从而降低了簇首的通信开销。同时该机制能够对簇成员进行批量分离或合并, 从而降低成簇开销, 提高了网络的性能表现。

1 共享最近邻密度峰聚类算法

共享最近邻密度峰聚类 (SNN-DPC, shared-nearest-neighbor-based clustering by fast search and find of density peaks) 算法是一种基于密度峰点的聚类算法, 它可以自动地将数据集划分成若干个簇。

共享最近邻的成簇基于这样的假设: 两个样本点的共同邻居总数越大, 则它们更相似。SNN-DPC 算法的基本思想是利用每个样本点的局部密度和样本点与其邻居的距离来判断该样本点是否为簇中心, 以及该点所属的簇。该算法提出了 K 近邻和共享最近邻这两个基本概念, 用于参与聚类的过程。

其中, K 近邻指样本点 x_i ($x_i \in X$) 与其他样本点相似度最大的 k 个点, 表达式如下:

$$\Gamma(i) = \{x_j \in X \mid \text{index}_{\text{sim}(x_i, x_j)} \leq k\} \quad (1)$$

式中, $\text{index}_{\text{sim}}(x_i, x_j)$ 指样本点 x_i 与其他点 x_j 的相似度在降序后的索引值。

共享最近邻指两个样本点 x_i 和 x_j 的 K 近邻集合的交集, 表达式如下:

$$\text{SNN}(i, j) = \Gamma(i) \cap \Gamma(j) \quad (2)$$

K 近邻和共享最近邻能够反映数据点之间的相似性和关联性, 同时还能够有效减少噪声和异常点对聚类结果的影响。

在 SNN-DPC 算法中, 首先需要构建每个点的共享最近邻图 (shared nearest neighbor graph, SNN), 然后根据每个点的局部密度和距离计算出各点的 SNN 密度和 SNN 距离, 接着再将各点的 SNN 密度和 SNN 距离作为坐标, 画出 SNN 密度图, 并找出其中的密度峰点。对于每个密度峰点, 选择其 SNN 密度大于一定阈值的最近邻点作为该点的邻居, 并将这些邻居点分配到该密度峰点所代表的簇中。如果邻居点已经被分配到其它簇中, 则将该密度峰点与其

它簇合并。不断迭代该过程，直到所有点都被分配到簇中。最后输出所有的簇结果。

SNN-DPC 算法适用于处理大规模、高维度、复杂结构的数据集，对于不同形状和大小的簇的识别效果较好。该算法可以自动确定簇的个数，同时对噪声点有很好的容忍性。但是，该算法中 K 值的设置对聚类结果有较大的影响，需要根据具体问题进行调整。

2 提出的 SNNCA 算法

道路环境上的车辆节点存在分布不均匀、移动速度快的特点。SNN-DPC 算法能够有效识别具有不规则形状的簇，但是在节点移动的情况下无法进行簇的划分。因此，本文提出一种基于改进的共享最近邻密度峰聚类的 VANET 分簇算法 SNNCA。

SNNCA 算法考虑了节点与其邻居节点的连接稳定性，将稳定性相似的节点划分为同一簇。本研究还提出了一种节点类型控制策略，能够快速分离部分簇节点或合并多个簇，从而降低了簇重构的成本。在本方法中，簇成员能够根据邻居节点情况自主维护簇信息，进行簇成员的合并或分离操作。簇首的作用被削弱，对簇的划分结果影响较小，因此本方法中簇首的选择采用随机选择的方式。

2.1 网络模型

2.1.1 网络假设

本文对车载自组织网作了以下假设：(1) 网络节点基于 WAVE 协议栈进行数据报文交换，WAVE 在 MAC 层使用 IEEE P1609 标准，在 PHY 层使用 IEEE 802.11p 标准^[20]；(2) 车辆配有 GPS 接收器，车辆可通过 GPS 实时获取其当前位置、速度、加速度等信息；(3) 车辆的时钟同步。

2.1.2 节点类型

SNNCA 算法规定了三种节点类型，每种节点类型的具体介绍如下。

普通节点 (ON, orphan node)：节点的初始类型；网络中的节点从 ON 类型转为其他类型后，不能再转为 ON 类型。

簇成员 (CM, cluster member)：节点加入簇后的状态。本成簇方法形成的簇结构为树形结构，簇层级从 0 开始记起，处于第 0 层的簇成员只有一个，称为簇首 (CH, cluster head)。

预备成簇节点 (RN, ready node)：从簇中离开后，节点类型转为 RN。RN 节点保留了上一次成簇的簇层级关系，便于节点快速恢复簇结构。

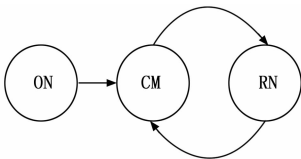


图 1 节点类型转移图示

2.1.3 控制报文

SNNCA 算法定义了以下四种控制报文用于其成簇过程和簇维护过程。

1) HELLO 报文：

网络中的节点通过周期性地广播 HELLO 报文来向其邻居节点更新自身信息。HELLO 报文的信息如表 1 所示。

表 1 HELLO 报文信息

报文类型	节点类型	本节点 IP
位置信息	速度信息	所属簇 ID
本节点簇层级	父级节点 IP	K 近邻节点 IP

2) JOIN 报文：

ON、RN 类型节点或者簇首节点通过向 CM 节点发出 JOIN 报文，来请求加入该 CM 节点所属的簇。JOIN 报文的信息如表 2 所示。

表 2 JOIN 报文信息

报文类型	本节点 IP	目的节点 IP
所属簇 ID	备选所属簇 ID	K 近邻节点 IP

3) APPROVE 报文：

CM 节点接收到入簇请求，若允许新节点加入簇，则通过 APPROVE 报文向新节点发出响应。APPROVE 报文的信息如表 3 所示。

表 3 APPROVE 报文信息

报文类型	本节点 IP	目的节点 IP
所属簇 ID	本节点簇层级	

4) CHANGE_STAT 报文：

当节点的簇信息发生改变时，需要通过 CHANGE_STAT 报文来通知子节点对自身的簇信息做相应的更新。CHANGE_STAT 报文的信息如表 4 所示。

表 4 CHANGE_STAT 报文信息

报文类型	节点类型	本节点 IP
目的节点 IP	所属簇 ID	本节点簇层级

2.2 连接稳定程度评估

车辆的位置坐标、速度以及加速度可以通过 GPS 获得。设车辆节点一跳的传输范围为 r ，节点 i 的速度为 v_i ，加速度为 a_i ，位置为 (x_i, y_i) ，速度方向与标准坐标系的夹角为 θ_i ，加速度方向与标准坐标系的夹角为 φ_i 。

2.2.1 链路生存周期

链路生存周期 $T(i, j)$ 指两节点 i, j 在彼此的通信范围内所能保持通信的最长时间。根据车辆节点的瞬时速度、加速度、位置，以及节点通信范围，可以预测节点间通信链路的生存周期。

对于车辆节点 i, j ，节点经过时间 Δt 后的位置为：

$$\begin{cases} x' = x + v \times \cos\theta \times \Delta t + \frac{a \times \cos\varphi \times \Delta t^2}{2} \\ y' = y + v \times \sin\theta \times \Delta t + \frac{a \times \sin\varphi \times \Delta t^2}{2} \end{cases} \quad (3)$$

当节点 i, j 间的距离不超过通信范围 r 时, 节点间的通信可靠; 两节点的距离为 r 时所经历的最大时间为节点间的链路生存周期, 可表示为:

$$r^2 = (d_n - d_0)^2 \quad (4)$$

其中: d_0 和 d_n 分别为节点 i, j 在 t_0 时刻和 t_n 时刻下的欧氏距离。将式 (4) 变形后, 得到方程:

$$d_n^2 - 2d_0d_n + (d_0^2 - r^2) = 0 \quad (5)$$

以 d_n 作为未知数, 求解方程 (5), 可以得到 d_n 的值。接着, 将 d_n 的值代入以下方程, 可以解得 t_n :

$$d_n^2 = [A \times t_n^2 + B \times t_n + (x_j - x_i)]^2 + [C \times t_n^2 + D \times t_n + (y_j - y_i)]^2 \quad (6)$$

其中:

$$A = \frac{a_j \times \cos\varphi_j - a_i \times \cos\varphi_i}{2}$$

$$B = v_j \times \cos\theta_j - v_i \times \cos\theta_i$$

$$C = \frac{a_j \times \sin\varphi_j - a_i \times \sin\varphi_i}{2}$$

$$D = v_j \times \sin\theta_j - v_i \times \sin\theta_i$$

t_n 值最大的解即为节点间的链路生存周期。链路生存周期可以反映两个节点之间通信连接的质量和可靠性。链路生存周期越长, 说明节点之间的通信连接越稳定, 可以更可靠地传输数据和信息。因此, 链路生存周期对于 VANET 网络中节点之间通信连接的评估和优化具有重要意义。

2.2.2 移动相似性

节点移动相似性是指在 VANET 网络中, 两个节点在一段时间内的移动行为具有相似性或一致性的程度, 可以通过节点的速度、加速度对节点移动相似性进行度量。对于节点 x_i 和节点 x_j , 两个节点的移动相似性可表示为:

$$S(i, j) = \frac{S_v(i, j) + S_a(i, j)}{2} \quad (7)$$

节点的移动相似性由速度相似性和加速度相似性加权求和得出。式 (7) 中, $S_v(i, j)$ 为节点间的速度相似性, $S_a(i, j)$ 为节点间的加速度相似性, 分别表示为:

$$S_v(i, j) = \frac{\cos(|\theta_i - \theta_j|) + \min\left(\left|\frac{v_i}{v_j}\right|, \left|\frac{v_j}{v_i}\right|\right)}{2} \quad (8)$$

$$S_a(i, j) = \frac{\cos(|\varphi_i - \varphi_j|) + \min\left(\left|\frac{a_i}{a_j}\right|, \left|\frac{a_j}{a_i}\right|\right)}{2} \quad (9)$$

其中: $\cos(|\theta_i - \theta_j|)$ 和 $\cos(|\varphi_i - \varphi_j|)$ 分别为节点间的速度余弦相似度和加速度余弦相似度, 余弦相似性越大, 则两节点的运动方向越一致, 即它们的移动轨迹越相似。 $\min\left(\left|\frac{v_i}{v_j}\right|\right)$ 和 $\min\left(\left|\frac{a_i}{a_j}\right|, \left|\frac{a_j}{a_i}\right|\right)$ 分别为节点的速度比值和加速度比值, 比值的范围为 $[0, 1]$, 比值越大, 则两节点的速度或加速度的值相差越小, 即两节点的相对

位置变化较小。

将移动相似性较高的节点划分为同一簇, 可以使得网络中节点的移动更加协同, 提高网络性能。同时, 可以减少因节点频繁移动导致的链路断开的情况, 从而提高网络的稳定性和可靠性。

2.2.3 连接波动程度

连接波动程度考虑了链路生存周期和移动相似性对节点间通信的影响, 能够对节点间通信链路的稳定程度进行评估:

$$F(i, j) = 1 - \frac{1}{1 + e^{-T(i, j)}} + \frac{1 - S(i, j)}{2} \quad (10)$$

连接波动程度 $F(i, j)$ 将链路生存周期和节点移动相似性的贡献加权求和, 它的值的范围为 $[0, 1]$ 。其中, $1 - \frac{1}{1 + e^{-T(i, j)}}$ 这一项表示链路生存周期的贡献, 通过 sigmoid 函数将 $T(i, j)$ 的值映射到 $[0.5, 1]$ 的区间内, 然后用 1 减去这个值, 得到一个从 0 到 0.5 的值, 表示链路生存周期对连接波动程度的负向影响。 $\frac{1 - S(i, j)}{2}$ 这一项表示节点移动相似性的贡献, 这一项将 $S(i, j)$ 的值从 $[0, 1]$ 映射到 $[0, 0.5]$ 的区间内, 表示移动相似性对连接波动程度的负向影响。

连接波动程度越低, 说明节点间的通信越稳定。反之, 连接波动程度越高, 说明节点间出现连接异常的可能性较大, 需要提前做好应对措施, 保障网络的正常运行。

2.3 基于共享最近邻的簇划分

基于共享最近邻的簇划分过程如图 2 所示。首先设置车辆的最近邻数为 K , 以车辆间的连接波动程度作为相似度函数, 计算车辆节点 x_i 和 x_j 的 K 近邻。接着, 基于节点 x_i 和 x_j 的 K 近邻计算两节点的共享最近邻 $SNN(i, j)$ 。

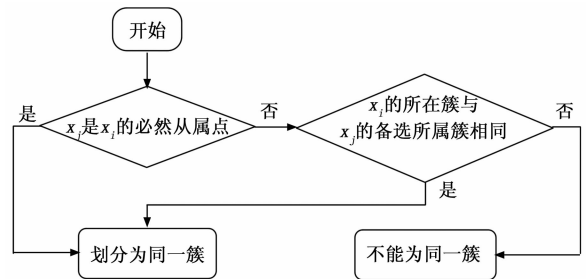


图 2 簇的划分过程

若共享最近邻的节点个数不小于 $\frac{k}{2}$, 即:

$$|SNN(i, j)| \geq \frac{k}{2} \quad (11)$$

则说明 x_i 和 x_j 拥有较多连接稳定的共同邻居, 两节点的网络情况相似, 因此可直接划分为同一簇。此时称节点 x_j 为节点 x_i 的必然从属点。

若共享最近邻的节点个数不超过 $\frac{k}{2}$, 即:

$$0 < |SNN(i, j)| < \frac{k}{2} \quad (12)$$

则说明 x_i 和 x_j 拥有的连接稳定的共同邻居数量较少，需要对 x_i 和 x_j 的关系做进一步判断，才能确定 x_j 是否从属于 x_i 所在的簇。此时称节点 x_j 为节点 x_i 的可能从属点。

为实现节点的分布式成簇，本文所提出的 SNNCA 算法未采用 SNN-DPC 算法中对可能从属点的再分配步骤，转而使用备选所属簇作为簇的划分依据。节点的备选所属簇是指在该节点的 K 近邻点中，出现次数最多的一个簇。假设簇的总数为 m ，则节点的备选所属簇可表示为：

$$C_{\max} = \arg \max_{C_m} \left\{ \sum_{x_j \in \Gamma(i)} [x_j \in C_m] \right\} \quad (13)$$

其中： $[x_j \in C_m]$ 是指如果点 x_j 属于簇 C_m ，则为 1，否则为 0。 \max_{C_m} 表示在簇 C_0 到 C_m 中取最大结果值的操作，即从所有可能的簇中找出节点出现次数最多的一个簇。节点的备选所属簇是本节点在一定范围内节点密度最高的簇，此时本节点属于该簇的可能性最大。因此，当节点 x_i 所在的簇与 x_j 的备选所属簇相同时，可以将这两个节点划分为同一簇。

2.4 SNNCA 算法流程

2.4.1 成簇过程

SNNCA 算法的成簇过程如图 3 所示，算法根据连接波动程度来计算节点的 K 近邻，以节点间的共享最近邻数量和备选所属簇作为簇的划分依据。具体的成簇步骤如下。

第一步：节点间周期性地互相发送 HELLO 报文，节点通过 HELLO 报文来获取自身一跳邻居节点的信息。初始化时，节点类型为 ON。

第二步：过滤邻居节点，更新邻居表。根据 HELLO 报文中提供的邻居节点信息，进行连接波动程度 F 的计算，将 F 值小于设定阈值的邻居节点加入邻居表。同时，当本节点与邻居节点互相为父级节点时，会将该邻居记录移除，以避免形成环路，造成簇首缺失的问题。

第三步：以节点间的连接波动程度作为节点间相似度的依据，连接波动程度越小，相似度越大。计算本节点的 K 近邻节点，将 K 近邻信息置于“HELLO”报文中广播。

第四步：ON 节点向其一跳邻居节点中的 CM 节点发出 JOIN 报文。若经过一定时间周期 ON 节点类型未改变，本节点将生成一个簇 ID，成为新的簇，节点类型转为 CM，并立即广播 HELLO 报文。

第五步：CM 接收处理 JOIN 报文，为满足条件的 JOIN 报文响应允许入簇的 APPROVE 报文。若发出 JOIN 报文的节点与本 CM 节点属于同一个簇，则将该报文丢弃，以避免簇内节点形成环路。根据 JOIN 报文中的 K 近邻信息判断请求节点是否满足必然从属点的条件，若满足则允许入簇。若仅满足可能从属点的条件，则判断 JOIN 报文中的备选所属簇是否与本节点的所属簇相同，若相同则允许入簇。

第六步：ON 接收到 APPROVE 报文后，节点类型转为 CM，并立即发出 HELLO 报文更新自身状态。

在网络初始化阶段，多个节点可能同时启动成簇过程，从而在短时间内形成多个簇。为了解决这个问题，本研究

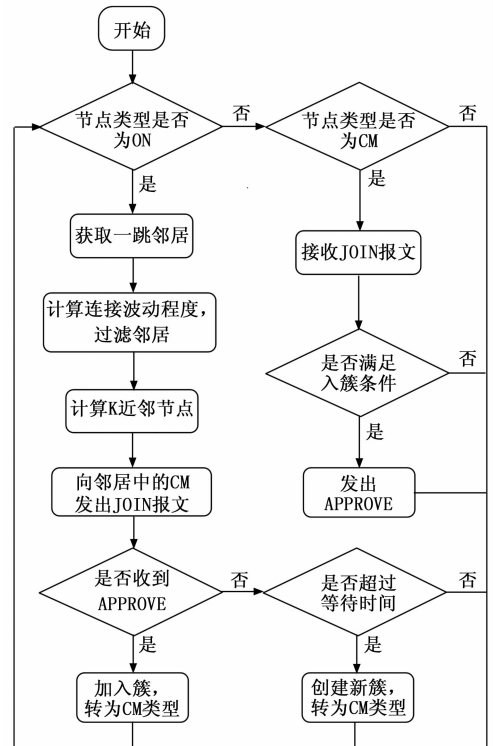


图 3 成簇过程流程图

提出了一种簇维护方法，该方法能够识别运动规律相似的多个簇，并将它们合并成一个簇，从而得到稳定且合理的簇结构。

2.4.2 簇成员离开

车载自组织网络由于其拓扑变化频繁，容易导致车辆节点之间链路的中断和重连。通过执行簇维护过程，可以降低簇重构的频率，提高簇的稳定性。簇维护过程涉及到簇成员的加入和离开两种情况。在 SNNCA 算法中，簇成员的父级节点和孩子节点均为其一跳邻居，因此节点可以根据其邻居表信息确定其父级节点和孩子节点的信息。

簇中的非簇首节点会周期性地检查其邻居表中是否存在其父级节点。若不存在，则说明本节点与父级节点的连接已断开，需要执行簇成员分离策略，从而将当前节点及其子节点从簇中分离，避免因节点的簇信息不一致而导致成簇结果不稳定。簇成员离开的处理过程如图 4 所示，具体步骤如下。

第一步：判断当前邻居表中是否存在与本节点所属簇相同且簇层级高于本节点的邻居节点。若存在满足条件的节点，则本节点向这些邻居节点发出 JOIN 报文，等待响应。若不存在满足条件的节点，则将本节点类型转为 RN，节点层级重置为 0，并立即发出 CHANGE_STAT 报文，通知其子节点更新簇信息。

第二步：节点接收到 CHANGE_STAT 报文，判断源节点 IP 是否与其父节点 IP 相同，若相同，则更新节点类型为 RN，自身簇层级设置为源节点簇层级加 1，并立即发出 CHANGE_STAT 报文给其下一级簇成员。

第三步: 下一级簇成员接收到 CHANGE_STAT 报文, 重复第二步。

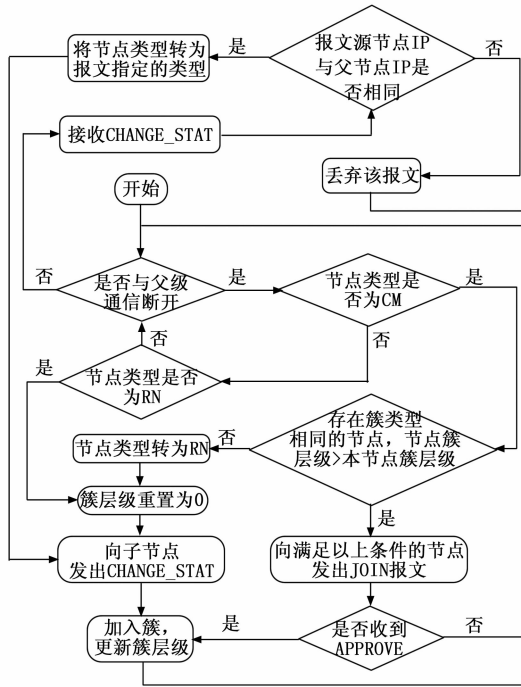


图 4 簇成员离开流程图

需要注意的是, RN 类型的节点不能主动发出或处理 APPROVE 报文, 需要等待其父级节点发出状态改变通知, 使其重新转为 CM 类型。若 RN 节点与其父节点通信断开, 则将本节点的层级重置为 0, 并立即向其孩子节点发出 CHANGE_STAT 报文。

SNNCA 的簇成员分离策略可以将簇中的一组节点分离, 并保留这组节点的簇层级关系。这种方法便于节点簇信息的快速更新, 降低了成簇开销。

2.4.3 簇成员加入

簇成员加入的处理过程如图 5 所示。当节点的类型为 RN 或 CM, 且簇层级为 0 时, 该节点会周期性地发出入簇请求, 从而实现单个节点加入簇, 或簇的合并, 具体步骤如下。

第一步: 若节点类型为 RN, 节点将周期性地向其一跳邻居中的 CM 节点发出 JOIN 报文, 并等待响应。若节点类型为 CM, 则当节点的备选所属簇与节点当前所属簇不同时, 才向其邻居中的一跳 CM 节点发出 JOIN 报文。

第二步: 若节点得到 APPROVE 报文响应, 则更新本节点的簇信息。若节点接收到多个 APPROVE 报文, 则优先选择与本节点的备选所属簇相同, 且源节点与本节点连接波动程度最小的报文。若 RN 节点未得到 APPROVE 报文响应, 则该节点在经过一段时间后将自动转为簇首, 生成新的簇。

第三步: 当节点的簇信息发生变化时, 会向连接稳定度没有超过设定阈值的子节点发送 CHANGE_STAT 报

文。该报文将通知子节点更新它们的簇信息, 以便与其它节点进行同步。

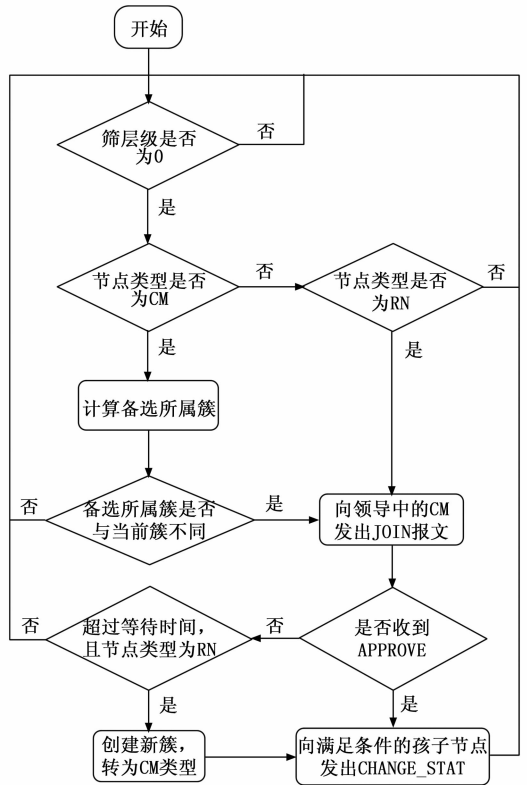


图 5 簇成员加入流程图

在处理簇成员加入的过程中, 步骤三可以帮助节点在执行簇合并操作之前, 提前将与其连接不稳定的子节点分离, 从而降低簇重建的频率。如果本节点与子节点的连接稳定度低于设定阈值, 说明该子节点与本节点通信的可靠性可能较低, 因此不必再维护它们之间的关系。

3 仿真分析

3.1 仿真场景和参数设置

本文采用 Matlab R2020a 软件对所提分簇算法的性能进行评估, 仿真区域为 $1\ 000 \times 1\ 000\text{ m}$ 的平面区域, 节点的初始位置在区域内随机产生, 节点移动模型为 Random Waypoint^[21]。节点数量 100 个, 移动速率为 $10 \sim 50\text{ m/s}$, 节点传输范围为 $100 \sim 300\text{ m}$ 。实验模拟时间为 100 s 。仿真时, SNNCA 算法的 K 近邻数设为 19, 连接波动程度阈值为 0.7。算法的评价指标如下。

1) 簇数量 C_n : 实验结束时网络中存在的簇数量。若存在未加入簇的节点, 则将该节点视为一个独立的簇。在分簇网络中, 簇内通信的开销和时延会优于簇间通信, 因此减少簇的数量能够降低通信开销, 提升网络性能。

2) 簇成员平均最长寿命 C_l : 网络节点在簇中的最长存在时间的平均值。簇成员寿命越高, 说明节点进行簇重建的频率越小, 簇结构越稳定。

3.2 仿真结果及分析

将 SNNCA 算法与文献 [18] 所提出的分簇算法 AC 在

不同实验场景下进行仿真比较，实验中每组数据由 50 次仿真结果的平均值得到。

实验考察节点的传输范围与移动速度对算法的影响，采用以下两种实验场景。场景一：节点移动速率为 30 m/s，节点传输范围为 100~300 m；场景二：节点传输范围为 200 m，节点移动速率为 10~50 m/s。

在实验中，随着节点传输范围的增大，网络中的簇数量逐渐减少。如图 6 所示，当传输范围超过 200 m 时，SNNCA 的簇数量减少趋势变缓，并稳定在 1 个。与 AC 算法相比，SNNCA 算法生成的簇数量明显较少。在传输范围偏小的情况下，两种算法生成的簇数量都较大。AC 算法属于单跳分簇算法，当传输范围较小时，节点一跳邻居的数量减少，因此簇的规模较小，簇数量较多。SNNCA 算法以两节点的共享最近邻作为分类依据，节点传输范围的减小意味着共享最近邻数较少，用于成簇决策的邻居节点信息较少，导致算法倾向于将节点分为不同类，从而增加簇数量。

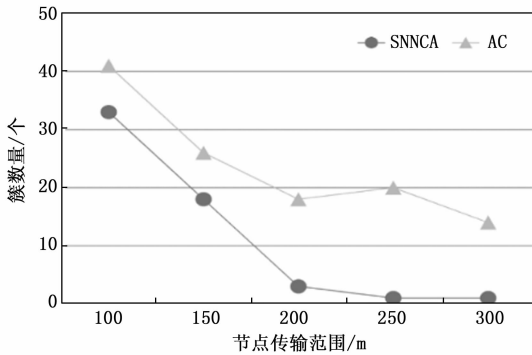


图 6 C_n 随节点传输范围的变化

如图 7 所示，SNNCA 算法和 AC 算法下的簇成员平均最长寿命随节点传输范围的增大逐渐上升。在 AC 算法中，节点的等级指标是根据其一跳邻居的信息计算得到的，提高传输范围有利于节点获取更多邻居信息，从而得到符合节点所处网络情况的等级值。较准确的等级值有助于算法选择合适的簇首，以降低簇重构的频率，提高簇成员寿命。此外传输范围的提高也能够增大簇首与簇成员间的连接时间。在 SNNCA 算法中，传输范围的增大有利于提高节点 K 近邻信息的完整性和正确性，从而提高算法分类的准确性，降低簇重建频率，使簇成员寿命增加。然而当传输范围超过一定值时，节点已经获取到足够的 K 近邻信息用于分类，因此分类结果趋于稳定，簇成员寿命稳定。

网络中的簇数量随节点移动速率的变化如图 8 所示。AC 算法在节点移动速率较小及较大的情况下生成的簇数量较多。SNNCA 算法中的簇数量随节点移动速率的增加，变化稳定，且簇数量较少。当移动速率较小时，节点在一段时间内的移动范围较小，网络中的节点位置分散，此时 AC 算法中得到的簇数量较多。当节点的移动速率较大时，节点的移动规律差异变大，此时 AC 算法难以推选出一致的簇

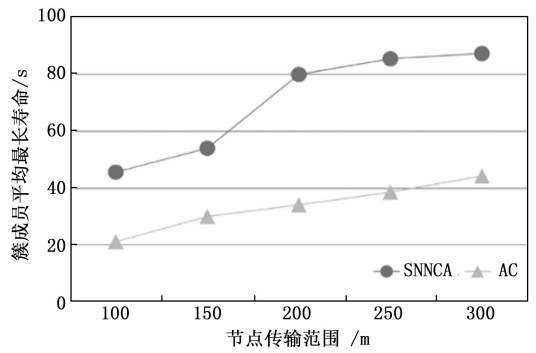


图 7 C_s 随节点传输范围的变化

首节点，因此簇的数量增加。SNNCA 算法拥有簇合并策略，在节点移动过程中，簇首节点会根据周围网络情况组织簇成员加入其他簇，因此网络中的簇数量稳定在较小范围。

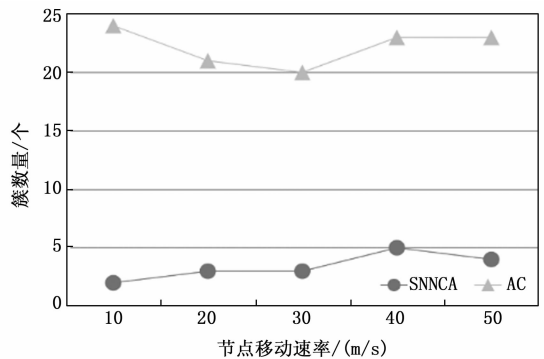


图 8 C_n 随节点移动速率的变化

如图 9 所示，两种算法中的簇成员平均最长寿命都随着节点移动速率的增加逐渐减少。AC 算法为单跳分簇算法，当节点移动速率增大时，簇成员离开簇首通信范围的时间缩短，因此簇成员寿命减少。SNNCA 算法在移动速率小于 30 m/s 时，簇成员寿命变化稳定；在移动速率大于 30 m/s 时，簇成员寿命下降幅度增大。在 SNNCA 算法中，当簇成员与其父节点的连接中断时，簇成员将启动合并策略，优先加入簇类型相同，且层级不小于其父级的合适节点。这种簇维护方式使簇结构能根据节点的运动情况进行主动优化，使节点在簇中的停留时间增加，从而提高了簇

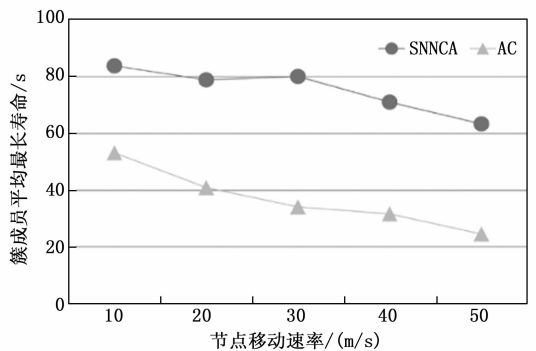


图 9 C_s 随节点移动速率的变化

成员寿命。但当节点移动速率过大时, 节点的移动规律变化较大, 节点难以保持在同一个簇中, 因此簇成员寿命减少。

3.3 不同 K 近邻数下的网络性能

在 SNNCA 算法中, 簇成员的确定需要节点的 K 近邻参与计算, 因此 K 近邻数的选取会影响成簇的结果。现对不同 K 近邻数下的 SNNCA 算法进行仿真对比。仿真环境中的参数设定为: 节点移动速率为 30 m/s, 节点传输范围为 200 m, 其他条件不变。仿真结果如表 5 所示。

表 5 不同 K 近邻数下的网络性能

K	指标	C_n	C_t
3		2	73.22
7		3	68.70
11		2	68.62
15		2	74.53
19		3	80.33
23		3	76.14
27		4	66.73
31		6	63.24
35		5	63.61
39		5	66.28

从表 5 中可以看出, 在本实验条件下, 当 K 值处于 15 至 23 的范围时, 算法得到的簇数量较少, 簇成员的寿命较长, 此时网络性能较稳定。

网络中的簇数量在 K 值小于等于 23 时稳定在 2~3 个, 而当 K 值超过 23 时, 簇数量发生小幅度增加。簇成员寿命的峰值出现在 K=19 时, 在 K 大于 23 时降低幅度增大。

SNNCA 算法中 K 近邻的设置是为了筛选出与节点连接相对稳定的邻居节点, 是判断两节点的邻居网络相似程度的依据。K 的取值过小会导致 K 近邻的数量不足以提供足够的邻居网络信息进行分类。当 K 值过大时, K 近邻成员包含了节点的大部分邻居, 失去了筛选邻居节点的作用。因此, 过小或过大的 K 值会导致算法分簇的准确性降低, 增加簇重建的发生频率, 造成网络中的簇数量增大, 簇成员寿命减少。

4 结束语

针对 VANET 网络拓扑变化频繁而导致的分簇结果不稳定的问题, 提出了适合随机部署场景的快速成簇算法 SNNCA。该算法引入了连接波动程度指标来评估节点间连接的稳定性, 并且基于共享最近邻思想进行成簇操作。SNNCA 算法考虑了节点之间网络情况的相似性, 将具有相似一跳邻居网络情况的节点分为同一簇, 提高了簇成员在簇内的存活时间。此外, 本研究还提出了一种簇维护策略, 通过周期性地检查节点间连接的情况, 对节点进行簇的分离与合并操作, 以构建层级合理、稳定的簇结构, 同时降低了簇的数量。针对节点数量多和成簇开销大的问题, 引

入了预备成簇节点类型, 使节点在离开簇后能够保留上一次的簇层级信息, 方便节点快速成簇, 从而降低了成簇开销。

经仿真测试, 与 AC 算法相比, SNNCA 算法得到的簇数量降低了 74%, 簇成员存活时间增加近 1 倍, 基于该成簇方法得到的网络稳定性强, 健壮性好。由于 SNNCA 中 K 近邻的数目由经验确定, 过大或过小的 K 近邻数会导致算法性能降低, 因此如何确定合适的 K 近邻数将是下一步的工作。

参考文献:

- [1] ZHANG H, LU X. Vehicle communication network in intelligent transportation system based on Internet of Things [J]. Computer Communications, 2020, 160: 799-806.
- [2] 颜文胜, 吕红兵. 基于 Deep Q Networks 的交通指示灯控制方法 [J]. 计算机测量与控制, 2021, 29 (6): 93-97.
- [3] CHEN C, LIU B, WAN S, et al. An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System [J]. IEEE transactions on intelligent transportation systems, 2021, 22 (3): 1840-1852.
- [4] SIROHI D, KUMAR N, RANA P S. Convolutional neural networks for 5G-enabled Intelligent Transportation System: A systematic review [J]. Computer Communications, 2020, 153: 459-498.
- [5] 陈登, 潘力. 移动边缘计算环境下基于改进 GPSR 的 VANET 路由算法 [J]. 计算机测量与控制, 2022, 30 (1): 188-195.
- [6] MAHI M J N, CHAKI S, AHMED S, et al. A review on VANET research: Perspective of recent emerging technologies [J]. IEEE Access, 2022, 10: 65760-65783.
- [7] REN M, ZHANG J, KHOUKHI L, et al. A review of clustering algorithms in VANETs [J]. Annals of Telecommunications, 2021, 76: 581-603.
- [8] MUKHTARUZZAMAN M, ATIQUZZAMAN M. Clustering in vehicular ad hoc network: Algorithms and challenges [J]. Computers & Electrical Engineering, 2020, 88: 106851.
- [9] KATIYAR A, GUPTA S K, SINGH D, et al. A dynamic single-hop clustering algorithm (DSCA) in VANET [C] // 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Kharagpur, India: IEEE, 2020: 1-6.
- [10] SENAN S, RAMASUBBAREDDY S, BALASUBA-MANIYAM S, et al. MADCR: Mobility aware dynamic clustering-based routing protocol in internet of vehicles [J]. China Communications, 2021, 18 (7): 69-85.
- [11] KUMAR D, BARANI D S. Improved multihop clustering algorithm in vanet based on dynamic mobility [C] // 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV). Tirunelveli, India: IEEE, 2021: 201-206.
- [12] KATIYAR A, SINGH D, YADAV R S. Advanced multi-hop

clustering (AMC) in vehicular ad-hoc network [J]. *Wireless networks*, 2022, 28: 45–68.

[13] JABBAR M K, TRABELSI H. A Novelty of Hyper-graph Clustering Model (HGCM) for Urban Scenario in VANET [J]. *IEEE Access*, 2022, 10: 66672–66693.

[14] AZIZIAN M, CHERKAOUI S, HAFID A S. A distributed D-hop cluster formation for VANET [C] // 2016 IEEE wireless communications and networking conference. Doha, Qatar: IEEE, 2016: 1–6.

[15] CHENG X, HUANG B. A Center-Based Secure and Stable Clustering Algorithm for VANETs on Highways [J]. *Wireless Communications and Mobile Computing*, 2019: 8415234.

[16] ALSUHLI G H, KHATTAB A, FAHMY Y A. Double-head clustering for resilient VANETs [J]. *Wireless Communications and Mobile Computing*, 2019: 2917238.

[17] BERSALI M, RACHEDI A, BOUARFA H. A New Collaborative Clustering Approach for the Internet of Vehicles (CCA-IoV) [C] // 2020 Second International Conference on Embedded & Distributed Systems (EDiS). Oran, Algeria: IEEE, 2020: 58–63.

[18] AZHDARI M S, BARATI A, BARATI H. A cluster-based routing method with authentication capability in Vehicular Ad hoc Networks (VANETs) [J]. *Journal of Parallel and Distributed Computing*, 2022, 169: 1–23.

[19] WANG L R, et al. Shared-nearest-neighbor-based clustering by fast search and find of density peaks [J]. *Information Sciences: An International Journal*, 2018, 450: 200–226.

[20] IEEE Std 1609.3-2010, IEEE Standard for Wireless Access in Vehicular Environments (WAVE) –Networking Services [S]. New York, American: IEEE, 2010: 1–144.

[21] 童超, 牛建伟, 龙翔, 等. 移动模型研究综述 [J]. *计算机科学*, 2009, 36 (10): 5–10.

（上接第 165 页）

泡体耦合动力学模型进行 6DOF 控制闭环轨迹仿真。超空泡飞行器是在尾拍振荡运动过程中实现稳定控制的。在稳定巡航以后，超空泡飞行器俯仰角振荡幅度为 $\pm 0.5^\circ$ ，在经过初始段加速和高度调整后，稳定巡航过程中深度控制误差小于 0.1 m，滚转角控制误差小于 4° 。表明该控制方法能够对飞行器滚转角和深度实现稳定控制，控制效果较好，能够满足超空泡飞行器航行要求。

5 结束语

本文给出了超空泡机动飞行试验弹局部线性化控制状态空间模型，在此基础上设计了俯仰通道和滚转通道极点配置变结构控制方法，通过超空泡飞行器六自由度泡体耦合模型进行弹道仿真，对所设计的控制方法进行了验证，能够实现超空泡飞行器稳定控制。

参考文献:

[1] 钱杏芳, 林瑞雄, 赵亚男编著. 导弹飞行力学 [M]. 北京: 北京理工大学, 2000.

[2] 余雄庆. 飞机总体设计 [Z]. 南京: 南京航空航天大学飞行器系讲义.

[3] 李新国, 方群. 有翼导弹飞行力学 [M]. 西安: 西北工业大学, 2005.

[4] HEISER W H, PRATT D T. Hypersonic Airbreathing Propulsion [R]. AIAA Education Series, 1994.

[5] DANIEL P RAYMER. Aircraft Design: A conceptual approach 2nd ed [M]. AIAA Education Series, American Institute of Aeronautics And Astronautics.

[6] 杨军, 杨晨, 段朝阳, 等. 现代导弹制导控制系统设计 [M]. 西安: 西北工业大学出版社, 2005.

[7] JOHN DZIELSKI, ANDREW KURDILA. A Benchmark control problem for supercavitating vehicles and an initial investigation of solutions [J]. *Journal of Vibration and Control*, 2003, 9 (7): 791–804.

[8] 李言俊, 张科. 自适应控制理论及应用 [M]. 西安: 西北工业大学出版社, 2005, 4.

[9] 周军, 等. 基于变结构理论的 BTT 导弹自动驾驶仪的三通独立设计 [J]. *宇航学报*, 1994, 1 (1): 42–47.

[10] 朱志刚, 等. BTT 导弹滚转通道变结构最终滑态控制系统设计 [J]. *西北工业大学学报*, 1995, 13 (2): 292–296.

[11] 王琦, 崔立君, 飞航导弹横测通道协调变结构飞行控制系统设计. *计算机测量与控制* [J]. 2007, 15 (12): 1756–1757.

[12] MATTHEW KUIPERS, PETROS IOANNOU. Robust adaptive multiple model controller design for an air-breathing hypersonic vehicle model [C] // AIAA Guidance, Navigation, and Control Conference and Exhibit. Honolulu, Hawaii, Aug, 2006.

[13] TRAVIS E GIBSON, ANURADHA M ANNASWAMY. Adaptive control of hypersonic vehicle in the presence of thrust and actuator uncertainties [C] // AIAA Guidance, Navigation, and Control Conference and Exhibit. Honolulu, Hawaii, Aug, 2006.

[14] GREGORY I M, MCMINN J D, SHAUGHNESSY J D. Hypersonic vehicle control law development using H_∞ and μ -synthesis [R]. NASA TM-4562, 1994.

[15] 严卫生. 鱼雷航行力学 [M]. 西安: 西北工业大学出版社, 2005, 1.

[16] 潘光. 鱼雷力学 [M]. 西安: 陕西师范大学出版社, 2013.

[17] 张宇文. 超空泡航行体流体力学 [M]. 北京: 国防工业出版社, 2014.

[18] 吕瑞, 等. 超空泡航行体的增益自适应变结构控制设计 [J]. *兵工学报*, 2010, 3: 303–308.

[19] 范加利, 等. 超空泡航行体的自适应模糊滑模控制研究 [J]. *计算机工程与应用*, 2011, 47 (3): 218–220.

[20] 王京华. 超空泡航行体非线性动力学建模与控制设计 [D]. 哈尔滨: 哈尔滨工业大学工学, 2010.