

# 无人船安全目标追踪与自动避障算法

耿以才, 黄立新, 陈凌珊, 杭鲁滨

(上海工程技术大学 汽车工程学院, 上海 201620)

**摘要:** 设计了无人船安全目标追踪算法和双回路追踪以及自动避障控制策略, 内回路是控制无人船安全目标追踪并与目标保持一定安全距离, 通过 Lyapunov 函数证明了该控制算法的渐近稳定性; 外回路是无人船在安全距离内发现障碍, 将目标点虚拟化, 运用模糊控制原理, 实现自动避障; 优化传统的 Dijkstra 算法, 设计了无人船实时最短路径算法; 仿真结果表明, 提出的综合算法能够实现无人船以合理速度通过最短距离, 顺利避障, 兼顾了时间效率。

**关键词:** 无人船; 模糊控制; 自动避障

## Safe Target-tracking Algorithm With Obstacle Avoidance for Unmanned Ship

Geng Yicai, Huang Lixin, Chen Lingshan, Hang Lubin

(College of Vehicle Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

**Abstract:** A safe target-tracking and obstacle avoidance control system for unmanned ship is proposed in this paper. Target-tracking law in the inner loop controls the ship to track the predetermined target by reducing the posture error to zero and its stability is analyzed by using Lyapunov function method after the tracking time efficiency is also taken into account. When any obstacle is detected with ultrasonic sensors, the external loop provides a modification on the target position based on the fuzzy control algorithm. Based on djijkstra algorithm, the improved shortcut algorithm of unmanned ship has been proposed. The reliability of this improved control system has been verified based on modeling and simulation of automatic collision avoidance with Matlab and VB6.0.

**Keywords:** unmanned ship; fuzzy control; automatic obstacle-avoiding

## 0 引言

自动避障技术是无人船安全稳定行驶的重要保证<sup>[1]</sup>。早期的避障算法是在建立被控制对象精确的数学模型和检测全面的环境信息基础上提出的, 控制算法复杂, 一般情况下被控对象的数学模型很难精确确定, 最终会导致避障效果不明显<sup>[2-3]</sup>。文献[4-5]设计了机器人安全追踪与自动避障算法, 但是没有具体讨论如何选择最短路径的问题。

本论文运用模糊控制理论, 通过优化传统 Dijkstra 算法讨论了最短路径<sup>[6]</sup>, 并嵌入安全追踪目标的算法, 通过 Lyapunov 函数<sup>[7]</sup>证明了该控制算法的渐行稳定性, 并进行了仿真研究。

## 1 参考坐标系的建立

假设船体与水面无摩擦, 如图1所示, 无人船从起始位置  $(x(t), Y(t))$  驶向目标点  $(x_s, y_s)$ , 定义船体的质心为转向点,  $\alpha$  为无人船的当前运动方向与质心到目标点连线的夹角,  $L$  为无人船到目标点的距离<sup>[8]</sup>。

根据图1结合几何知识, 可得:

$$V(t) = \arctan \frac{\Delta y}{\Delta x} = \frac{y(t+1) - y(t)}{x(t+1) - x(t)} \quad (2)$$

$$L = \sqrt{(x_s - x(t))^2 + (y_s - y(t))^2} \quad (3)$$

$$\alpha = \arctan \frac{(y_s)}{(x_s)} - V(t) \quad (4)$$

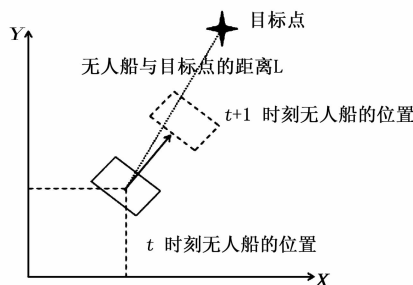


图1 XOY参考坐标系

无人驾驶船的运动学方程为:

$$\begin{cases} \dot{x} = v(t) \cos \theta \\ \dot{y} = v(t) \sin \theta \\ \dot{\theta} = \omega \end{cases}$$

其中:  $(x, y, \theta)$  是无人驾驶船的位姿, 无人船与目标点的距离为  $L$ ,  $L_d$  为距离目标点的安全距离。记无人驾驶船与目标点的距离误差为  $L_{\text{误}} = L - L_d$ , 则无人驾驶船的位姿误差极坐标方程为

$$\begin{cases} L_{\text{误}} = -v(t) \cos \alpha \\ \dot{\alpha} = \omega + \frac{v(t)}{L_{\text{误}} + L_d} \sin \alpha \end{cases} \quad (5)$$

很多情况下, 由于工作于未知的环境中, 无人驾驶船只能通过传感器<sup>[9]</sup>探测障碍物, 所以无人船的目标追踪问题应该考虑两个方面的问题, 问题的核心是要设计合适的控制策略指导无人驾驶船能够以平稳且高效的速度追踪到目标并与障碍物保

收稿日期:2014-10-24; 修回日期:2014-12-04。

基金项目:科技成果转化项目(E4186000203)。

作者简介:耿以才(1990-),男,江苏徐州人,硕士研究生,主要从事汽车发动机控制及后处理系统方向的研究。

黄立新(1964-),女,上海宝山人,副教授,特高级实验师,硕士研究生导师,主要从事高速齿轮切削方向的研究。

持一定的距离,其实也就是使得极坐标参数 ( $L_{\text{误}}, \alpha$ ) 逐渐减小到 (0, 0), 因此本文利用模糊控制<sup>[10]</sup>的理念和双回路的控制实施方案来解决无人船追踪目标和自动避障问题,如图 2 所示。其中是无人船新的目标点,它可以通过旋转矩阵得到,即

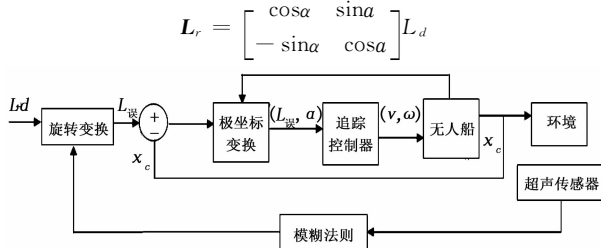


图 2 无人船安全追踪目标及自动避障框图

## 2 追踪目标算法

考虑距离障碍的安全距离和无人驾驶船的运动速度的情况下,控制策略设计如下。

Lyapunov 函数选择为:

$$v(t) = \frac{1}{2}L_{\text{误}}^2 + \frac{1}{2}\alpha^2 \quad (6)$$

对该函数求导并根据无人船运动学方程,得:

$$\dot{v}(t) = L_{\text{误}}(v\cos\alpha) + \alpha(\omega + \frac{v}{L_{\text{误}} + L_d}\sin\alpha) = v_1 + v_2 \quad (7)$$

取线速度  $v$  为:

$$v = u \cdot S(L_{\text{误}}) \cdot \cos\alpha \quad (8)$$

其中:  $u$  是无人驾驶船能够达到的最大线速度,定义  $S$  形函数<sup>[11]</sup>为:

$$y = \begin{cases} -1, L_{\text{误}} < -k \\ (\frac{L_{\text{误}} + k}{k})^2 - 1, -k < L_{\text{误}} \leq 0 \\ 1 - (\frac{L_{\text{误}} - k}{k})^2, 0 < L_{\text{误}} \leq k \\ 1, L_{\text{误}} > k \end{cases} \quad (9)$$

$k$  是一个可以设计的常数,将线速度  $v$  代入式 (7) 的第一项,得:

$$v_1 = -u \cdot L_{\text{误}} \cdot S(L_{\text{误}}) \cdot \cos^2\alpha \quad (10)$$

根据此式分析可得,当  $L_{\text{误}} \neq 0$  时,  $L_{\text{误}}$  与  $S(L_{\text{误}})$  总是保持同号,所以对于所有的  $L_{\text{误}}$ ,有  $(L_{\text{误}} \cdot S(L_{\text{误}})) \geq 0$  成立,根据常识,线速度  $v$  大于 0,  $\cos^2\alpha \geq 0$ ,可推出  $v_1 \leq 0$ 。

将线速度  $v$  代入式 (7),所以该式的第二项

$$v_2 = \alpha \cdot \omega + \frac{u \cdot S(L_{\text{误}}) \cdot \cos\alpha}{L_{\text{误}} + L_d} \sin\alpha \quad (11)$$

将无人驾驶船的转动速度设计为:

$$\omega = -c\alpha - \omega + \frac{u \cdot S(L_{\text{误}}) \cdot \cos\alpha}{L_{\text{误}} + L_d} \sin\alpha \quad (12)$$

其中:  $c$  是常数,大于 0,根据式 (12),有:

$$|\omega| \leq c\pi + 0.5v_{\text{max}}/L_d = |\omega|_{\text{max}} \quad (13)$$

其中是允许无人驾驶船最大转动速度,常数  $c$  满足

$$0 < c \leq |\omega|_{\text{max}} - 0.5v_{\text{max}}/L_d$$

将式(12)代入式(11)可得

$$v_2 = -c\alpha^2 \leq 0$$

根据以上分析, Lyapunov 候选函数的一阶导数即式:

$$\dot{v}(t) = v_1 + v_2 \leq 0$$

只有  $L_{\text{误}} = 0$  且  $\alpha = 0$ ,  $\dot{v}(t)$  才会是 0,因此  $\dot{v}(t)$  是小于 0 的,这也就意味着误差变量 ( $L_{\text{误}}, \alpha$ ) 能够逐渐趋近于 (0, 0),可以安全的追踪并到达目标。

该控制算法中,  $S(L_{\text{误}})$  的参数  $k$  决定无人船检测到障碍开始减速的距离。当无人船在安全距离  $k$  内没有检测到障碍物时,  $S(L_{\text{误}}) = 1$ ,此时无人船以上文求出的优化速度迅速向目标移动,在安全距离  $k$  内发现障碍物时,  $S(L_{\text{误}}) < 1$ ,无人船逐渐减速,避免撞到障碍。当追踪目标为静止时,将无人船速度以及加速度代入式(9)可以计算得到所允许的最小减速距离。

在可以得到任意时刻目标点位置的情况下,式(8)与式(12)同样适用于无人船追踪动态目标。在无人船接近目标点时,此时可以认为是趋近于 0 的,即无人船沿直线向目标运动。具体情况如下图 3 所示。

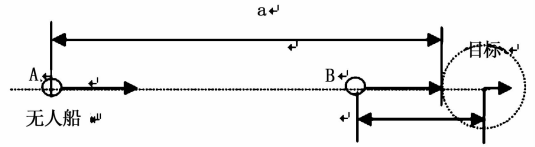


图 3 无人船逼近运动目标情况

根据  $S(L_{\text{误}})$ , 为避免碰撞,无人船与障碍物或者目标的距离为  $a$  时,无人船开始减速,到达  $B$  点时,无人船与目标运动速度相同,并保持安全距离,根据上文式 (9),可得:

$$v_{\text{max}}(1 - \frac{|AB|^2}{a^2}) = v_T \quad (14)$$

其中:  $v_T$  是目标运动速度,  $a$  是设定的减速距离,因此

$$|AB| = a \sqrt{1 - v_T / v_{\text{max}}}$$

所以,

$$\bar{L}_d = L_d + \sqrt{1 - v_T / v_{\text{max}}} \quad (15)$$

从式 (15) 可看出无人船跟踪目标的实际安全距离比初始值增大,当然若要求无人船以安全距离初始值追踪目标,只需要

$$\bar{L}_d = L_d - \sqrt{1 - v_T / v_{\text{max}}} \quad (16)$$

## 3 实时最短路径

传统的 Dijkstra 算法主要有效率较低、无法对这些路径进行动态选择缺点、无法准确估计船只到达目标点的时间及只能按照预先设计的路径行走等缺点。针对这种情况,结合无人船行驶的环境,提出了一个动态网络环境下实时路径模型,在此基础上构造了 Dijkstra 双桶算法<sup>[12]</sup>。

### 3.1 实时路径数学模型

基于传统的 Dijkstra 算法以及考虑河道宽度、天气、船只的情况下,提出一个动态网络环境实时路径评估模型,建立数学模型。

起始点与目标点之间为一个交通网络。  $D = (A, B, C)$  是有向图,其中  $A$  由  $m$  个节点组成的点集,  $B$  由  $n$  个节点组成的边集,  $C$  是路权集,假设

1)  $D$  只是一个简单图,不存在环形或者多重边行;

2) 路权具有可加性,  $A_i, A_k, A_j$ , 有  $l(A_i, A_j) = l(A_i, A_k) + l(A_k, A_j)$

起始点与目标点间的最短路径  $Q$  为:

$$\min L = \sum_{(A_i, A_j) \in Q} l(A_i, A_j)$$

对于任意一条边  $B$ , 都有一个与之对应的权值。

3.2 改进后的实时最短路径算法

为减少需要搜索的节点数量, 按权值大小, Dijkstra 双桶算法将待扫描的算法分别装在两个桶内, 权值大的放在高级别的桶内, 利用该算法选择下一个被扫描的节点。每次扫描都只扫描低级别桶内的节点, 如果低级别桶的数量为  $R$ , 则第  $i$  个高级别桶中的权值在  $[iR, (i+1)R]$  范围并被扫描的节点  $P$  定为指针, 而且指向最小的一个非空的高级别桶内。当低级别桶中节点被完全扫描, 则  $P+1$ , 如果所以高级别桶为空, 算法停止。

扫描过程的输入低级别桶内的节点和有向图  $D$ , Dijkstra 双桶算法不断调用扫描过程对节点进行扫描, 直到所有节点都被扫描结束, 算法终止。 $a$  表示源节点,  $e$  是边  $B$  的最大权值, 节点  $p$  的父节点为  $P(e)$ , 每一个节点  $P$ , 低级别桶数目为  $M$ , 高级别桶数目为  $N$ ,  $S(P)$  取下列值中一个: `unreached`, `labelled`, `scanned`。

扫描算法为:

```
1) for each child e of A do:
   if d (e) > d (A) + e (A, e)
     d (e) = d (A) + e (A, e)
     P (e) = A
     S (e) = labeled
   end if
2) S (A) = scanned
该算法主要过程为:
1) d(a) = 0, S(a) = labeled
2) for a ∈ A/{a} do: d(p) = ∞, S(p) unreached
3) N = [(E + 1)/M]
4) for L = 0 to N do /高级别桶扫描
5) for A = 0 to M do / 扫描所有低级别桶
   while? Empty (Low [M]) do
     Scan (Head (Low [M]))
   for all {w|((Head(Low[M]), w) ∈ C) ∧ (S(w)=labelled))} do
     if d(w) ≥ (E + 1)M
       then Insert (High[d(w)/M], w)
     else Insert (Low[d(w) - EM], w)
     end for
   end while
```

该有向图共有  $m$  个节点, 根据标准的 Dijkstra 算法, 该改进后的 Dijkstra 算法的执行时间为  $O(m^2)$ 。

4 仿真

4.1 寻障模式模糊控制仿真模拟

根据前文所述, 整个避障行为分为安全追踪模式和自动避障模式两种。两种模式仿真实情况类似, 本文只给出安全追踪模式的仿真。在安全追踪模式下, 该算法的输入量为是无人船与目标点的距离  $L$  以及船运动方向与船的质心与目标点连线所形成的夹角, 输出量为无人船的转向角  $\theta$ 。在避障模式下, 输入量为无人船正前方与障碍物之间的距离  $df$  以及船体两侧传感器与障碍物的距离  $ds$ , 左侧为正, 右侧为负, 输出量仍是无人船的转向角  $\theta$ 。

在安全追踪模式下, 输入量  $L$  的模糊分割 = {零, 小, 中, 大} = { $ZE, S, M, L$ }, 它的取值范围是  $[0, 10]$ ; 的取值范围为  $[-200, 200]$ , 模糊语言为 {负大, 负中, 负小, 零, 正小, 正中, 正大} = { $NB, NM, NS, ZE, PS, PM, PB$ } (此

处默认目标点在无人船的左侧时方向夹角为正, 位于右侧时为负); 同理, 将无人船的转向角  $\theta$  取值范围是  $[-30, 30]$  区间内, 模糊语言描述为  $\theta = \{右大, 右中, 右小, 零, 左小, 左中, 左大\} = \{BR, MR, SR, ZE, SL, ML, BL\}$  (默认无人船左转为正, 右转为负)。

安全追踪目标模式下, 根据输入量模糊分割, 共有条规则, 如表 1 所示。

表 1 追踪目标模式下的模糊规则

$\alpha \setminus L$	$ZE$	$S$	$M$	$L$
$PB$	$LB$	$LB$	$LB$	$LM$
$PM$	$LB$	$LB$	$LM$	$LM$
$PS$	$LB$	$LM$	$LM$	$LS$
$ZE$	$ZE$	$ZE$	$ZE$	$ZE$
$NS$	$RB$	$RB$	$RB$	$RM$
$NM$	$RB$	$RB$	$RM$	$RM$
$NB$	$RB$	$RM$	$RM$	$RS$

从图 4 中可看出, 当  $0^\circ < \alpha < 60^\circ$  时, 随着的增大,  $\theta$  也相应地增大, 但始终大于  $0^\circ$ , 这说明在无人船的行驶方向与目标点方向夹角比较小时且目标点位于无人船的左侧时, 无人船转向角也比较小, 船下一步应左转。 $60^\circ < \alpha < 110^\circ$  左右时, 随着的变化, 无人船转向角保持在最大值上。 $110^\circ < \alpha < 180^\circ$  时, 随着的增大,  $\theta$  也相应地减小。无人船转向角  $\theta$  随的变化还与无人船离目标点的远近有关, 当  $0 < L < 2$  时, 无人船离目标点比较近, 无人船的转向角变化更敏感, 随的变化幅度比较大。 $6 < L < 10$  时, 无人船离目标点比较远, 转向角随的变化幅度相对小一些。

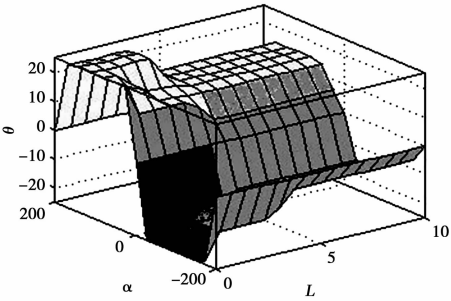


图 4 寻障模式下的输入输出关系曲面图

4.2 自动避障过程仿真模拟

运用 VB6.0 软件对无人船自动避障过程模拟, 图中箭头代表无人船, 深色物体代表障碍, 浅色物体代表目标, 无人船整个自动避障行为过程如图 5 所示。

开始时, 无人船传感器在左右前等方向上没发现障碍, 进入安全追踪模式, 向目标快速前进; 传感器检测到障碍时, 进入避障模式, 避开障碍物。

5 结论

本文将模糊控制应用于自动避障算法, 将被控过程用模糊规则描述, 经逻辑推理得到控制量的模糊量, 利用三角函数转化为精确的控制量, 优化传统的 Dijkstra 算法得出无人船到达