

说明文档

一. 算法详细介绍

1.1. 数学定义

对一向量 $\mathbf{c} = [c_x, c_y]^T$ ，可定义其在全局坐标系 xOy 中的方向角 $\angle \mathbf{c} \in [0, 2\pi)$ 为

$$\angle \mathbf{c} = \begin{cases} \arctan \frac{c_y}{c_x}, & c_x > 0 \wedge c_y \geq 0 \\ \arctan \frac{c_y}{c_x} + 2\pi, & c_x > 0 \wedge c_y < 0 \\ \arctan \frac{c_y}{c_x} + \pi, & c_x < 0 \\ \frac{\pi}{2}, & c_x = 0 \wedge c_y > 0 \\ \frac{3\pi}{2}, & c_x = 0 \wedge c_y < 0 \end{cases} \quad (1)$$

对两个方向角 $a \in [0, 2\pi)$ ， $b \in [0, 2\pi)$ ，它们的角度差为

$$\text{dif}(a, b) = \begin{cases} 2\pi - |a - b|, & |a - b| > \pi \\ |a - b|, & |a - b| \leq \pi \end{cases} \quad (2)$$

1.2. 运动学方程

围捕机器人和目标的运动学方程为：

$$\begin{aligned} p_x[k+1] &= p_x[k] + v[k] \cos \theta[k] \cdot T_s \\ p_y[k+1] &= p_y[k] + v[k] \sin \theta[k] \cdot T_s \\ \theta[k+1] &= \theta[k] + \omega[k] \cdot T_s \end{aligned} \quad (3)$$

其中 $p_x[k]$ ， $p_y[k]$ 分别表示该个体第 k 步时在 xOy 全局坐标系中的 x 坐标和 y 坐标， $\theta[k] \in [0, 2\pi)$ 表示在 xOy 全局坐标系中从 x 轴正方向顺时针旋转到该个体第 k 步时车头方向的角度， $v[k]$ 表示第 k 步时该个体的控

制输入线速度， $\omega[k]$ 表示第 k 步时该个体的控制输入角速度， T_s 为仿真的采样时间。

1.3. 拓扑结构

本文所描述的围捕-逃脱仿真模型为根据距离的时变拓扑结构，目标只有在围捕机器人的观察范围内时才可被获取到相对位置信息，而围捕机器人之间可以全局通信。以围捕机器人数量是5，目标数量是1的情况为例，有下图：

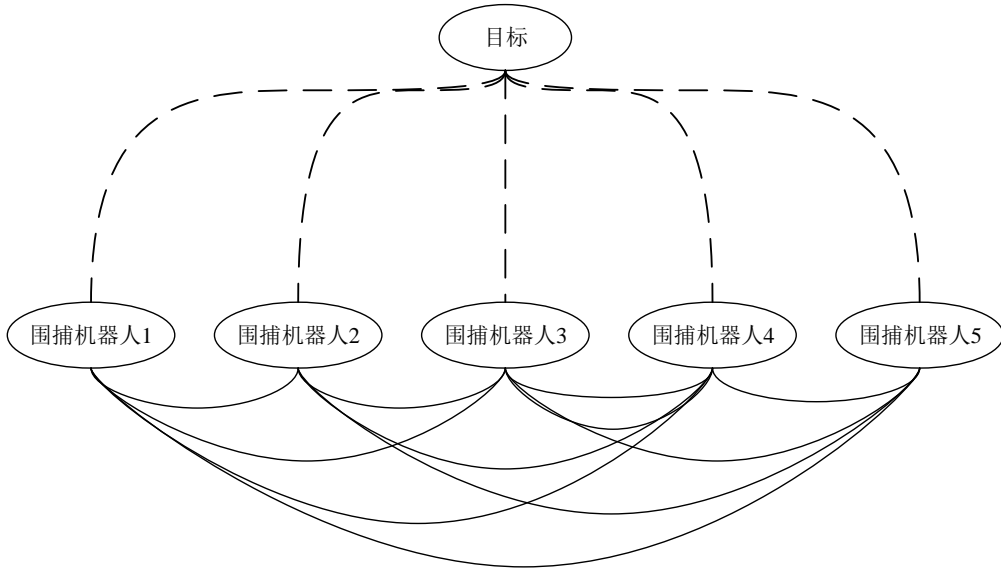


图 1: 拓扑结构图

图1中的实线表示，在任意条件下都可以通信，虚线表示当虚线连接的另一节点和本节点距离小于本节点的感知距离时，本节点即可获得另一端节点的相对位置信息。

1.4. 围捕策略

在不考虑避障的情况下，机器人 \mathcal{R}_i 的速度设计为：

$$\boldsymbol{\nu}_i[k] = \boldsymbol{\nu}_{i1}[k] + \boldsymbol{\nu}_{i2}[k]. \quad (4)$$

1.4.1. 前进速度

围捕机器人的前进速度为：

$$\boldsymbol{\nu}_{i1}[k] = \begin{cases} \frac{\alpha(d_{ti}[k] - r_h)}{d_{ti}^{\tau_1}[k]}(\mathbf{p}_0[k] - \mathbf{p}_i[k]), & d_{ti}[k] \leq r_w \\ \alpha\varsigma(\mathbf{p}_j[k] - \mathbf{p}_i[k]) \cdot d_{ij}^{\tau_2}[k], & d_{ti}[k] > r_w \end{cases} \quad (5)$$

其中 \mathbf{p}_j 是离 \mathcal{R}_i 最近的能观察到目标的围捕机器人 \mathcal{R}_j 的坐标， α ， ς ， τ_1 ， τ_2 是系数， $d_{ti}[k] = \|\mathbf{p}_i[k] - \mathbf{p}_0[k]\|$ ， $d_{ij}[k] = \|\mathbf{p}_j[k] - \mathbf{p}_i[k]\|$ ， r_w 是围捕机器人的观察范围。

1.4.2. 排斥速度

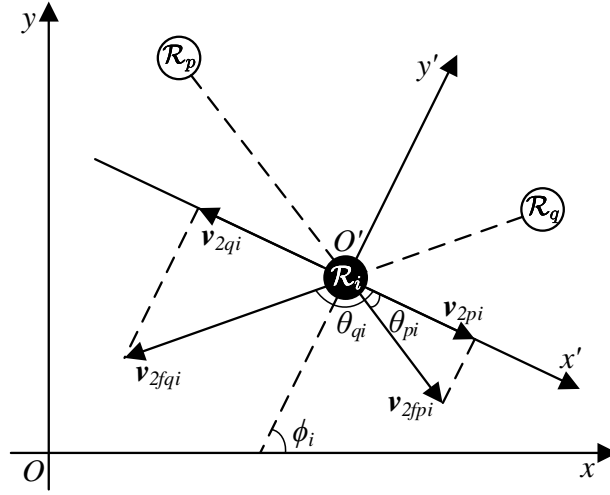


图 2: $x'Oy'$ 坐标系

以目标所在方向为 y' 轴， y' 轴顺时针旋转 $\pi/2$ 为 x' 轴建立 $x'Oy'$ 坐标系。

机器人的排斥速度为

$$\boldsymbol{\nu}_{i2}[k] = \begin{cases} \boldsymbol{\nu}_{2pi}[k] + \boldsymbol{\nu}_{2qi}[k], & d_{pi}[k] < r_w, d_{qi}[k] < r_w \\ \boldsymbol{\nu}_{2pi}[k], & d_{pi}[k] < r_w, d_{qi}[k] \geq r_w \\ \boldsymbol{\nu}_{2qi}[k], & d_{pi}[k] \geq r_w, d_{qi}[k] < r_w \\ 0, & d_{pi}[k] \geq r_w, d_{qi}[k] \geq r_w \end{cases} \quad (6)$$

其中 \mathbf{u} 是图2中的 $x'Oy'$ 坐标系 x' 轴正方向的单位向量,

$$\nu_{2pi}[k] = \|\nu_{2fpi}[k]\| \cos \theta_{pi}[k] \cdot \mathbf{u} \quad (7)$$

$$\nu_{2qi}[k] = \|\nu_{2fqi}[k]\| \cos \theta_{qi}[k] \cdot \mathbf{u}$$

$$\begin{aligned} \nu_{2fpi}[k] &= \beta \frac{\mathbf{p}_i[k] - \mathbf{p}_p[k]}{\|\mathbf{p}_i[k] - \mathbf{p}_p[k]\|^{\tau_3}} \\ \nu_{2fqi}[k] &= \beta \frac{\mathbf{p}_i[k] - \mathbf{p}_q[k]}{\|\mathbf{p}_i[k] - \mathbf{p}_q[k]\|^{\tau_3}} \end{aligned} \quad (8)$$

1.5. 避障策略

机器人总的速度设计为:

$$\mathbf{v}[k] = f(\nu_i[k], \mathbf{v}[k-1], \bar{S}_i[k], \bar{B}[k]) \quad (9)$$

其中包含避障的算法。具体如下:

1.5.1. 不规则障碍物

如图3, 将个体 \mathcal{R} 与最近的不规则障碍物的两条切线构成的角的角平分线表示为参考线 L , 那么个体的速度 $\mathbf{v}[k]$ 的方向按逆时针方向距离 L 的角度即为 x , 个体与不规则障碍物切线和 L 的夹角为 θ 。

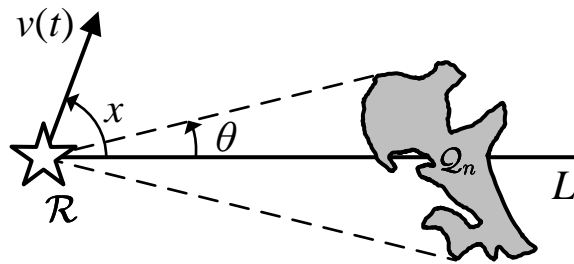


图 3: 个体遭遇不规则障碍物的情况

1.5.2. 矩形边界

如图4, 将个体 \mathcal{R} 与矩形边界上最近点的连线表示为参考线 L , 那么个体的速度 $\mathbf{v}[k]$ 的方向按逆时针方向距离 L 的角度即为 x , 个体的安全角度为 $\theta = \frac{\pi}{2}$ 。

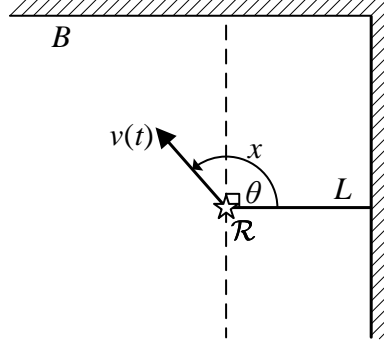


图 4: 个体遭遇矩形边界的情况

1.5.3. 避障算法

以上2节中展示了个体遇到各种障碍的情况及相应的安全角度 θ 。

假定 t 时刻个体 \mathcal{R} 的观察范围为 $\Phi[k] = \{z \mid z \in \mathbb{R}^2, \|z - p[k]\| \leq r_a\}$, 其中 r_a 为个体的避障距离。 Q_i 表示第 i 个障碍物, $S_i[k] \subseteq \mathbb{R}^2$ 是 t 时刻障碍物的形状中所包含的所有点, B 表示边界上所有点的集合。

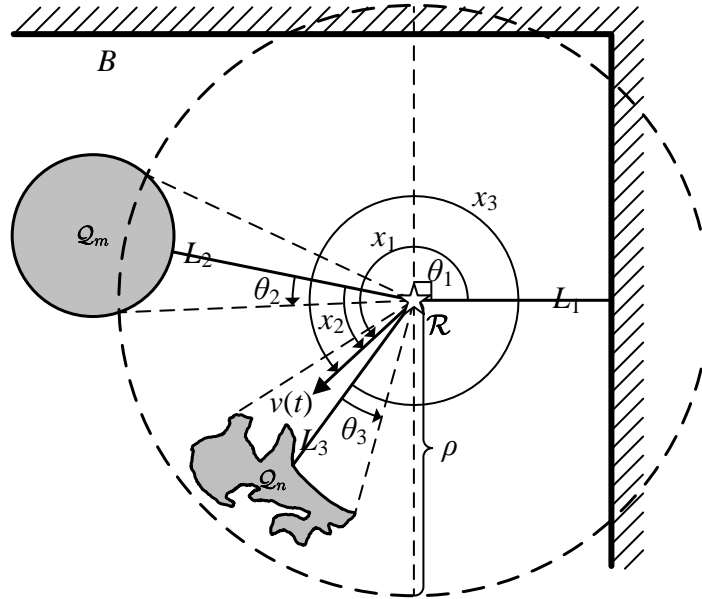


图 5: 个体遭遇多障碍的情况

以图5为例, 令 $\bar{S}_i[k] = \Phi[k] \cap S_i[k]$, $\bar{B}[k] = \Phi[k] \cap B$, 则总的速度设计为:

$$v[k] = f(\nu_i[k], v[k-1], \bar{S}_i[k], \bar{B}[k]) \quad (10)$$

其中, $\nu_i[k] \in \mathbb{R}^2$ 表示不在避障模式下的个体的速度。

由图5可知, 个体 \mathcal{R} 遭遇了障碍物 Q_m 的可见部分 $\bar{S}_m[k]$ 、 Q_n 的可见部分 $\bar{S}_n[k]$ 和边界 B 的可见部分 $\bar{B}[k]$ 。根据前2节的设定可得 $\bar{B}[k]$ 对应的参考线为 L_1 、安全角度 θ_1 、角度 x_1 、表面与个体最近点距离 d_1 , $\bar{S}_m[k]$ 对应的参考线 L_2 、安全角度 θ_2 、角度 x_2 、表面与个体最近点距离 d_2 , $\bar{S}_n[k]$ 对应的参考线 L_3 、安全角度 θ_3 、角度 x_3 、表面与个体最近点距离 d_3 。

令 $\phi_j = \kappa\theta_j$, 则对个体 \mathcal{R} , 若将 \mathbf{L}_j 视为从个体出发到障碍物的向量(模为1), 其某一个危险角度区间可表示为 $(\underline{\varphi}_j, \bar{\varphi}_j)$, 其中 $\underline{\varphi}_j = \angle \mathbf{L}_j - \phi_j$, $\bar{\varphi}_j = \angle \mathbf{L}_j + \phi_j$ 。在 \mathcal{R} 的避障距离内只有一个障碍物(包括边界)时, 对边界为 $\kappa = 1.2$, 对固定、移动圆形障碍物和不规则障碍物为 $\kappa = 2.0$, 对移动不规则障碍物为 $\kappa = 2.8$; 在 \mathcal{R} 的避障距离内有多于一个障碍物(包括边界)时, 对边界为 $\kappa = 1.2$, 对固定、移动圆形障碍物和不规则障碍物为 $\kappa = 1.6$, 对移动不规则障碍物为 $\kappa = 2.0$ 。

将在 $\bar{B}[k]$ 危险角度范围内的方向角 θ 表示为集合 $\Theta_1 = \{\theta | \theta \in (\underline{\varphi}_1, \bar{\varphi}_1)\}$, 在 $\bar{S}_m[k]$ 危险角度范围内的速度矢量 \mathbf{v} 表示为集合 $\Theta_2 = \{\theta | \theta \in (\underline{\varphi}_2, \bar{\varphi}_2)\}$, 在 $\bar{S}_n[k]$ 危险角度范围内的速度矢量 \mathbf{v} 表示为集合 $\Theta_3 = \{\theta | \theta \in (\underline{\varphi}_3, \bar{\varphi}_3)\}$ 。

对于个体 \mathcal{R}_i , 设计其基于航向的避障速度为

$$\mathbf{v}_i[k] = \begin{cases} \mathbf{v}_{ia}[k], & \angle \nu_i[k] \in \Theta_j, j = 1, 2, 3. \\ \nu_i[k], & (\angle \nu_i[k] \notin \Theta_1) \wedge (\angle \nu_i[k] \notin \Theta_2) \wedge (\angle \nu_i[k] \notin \Theta_3) \end{cases} \quad (11)$$

其中, $\mathbf{v}_{ia}[k]$ 设计如下:

$$\mathbf{v}_{ia}[k] = ||\boldsymbol{\nu}_i[k]|| * \mathbf{u}_a \quad (12)$$

其中 $\mathbf{u}_a = [\cos \varphi_a, \sin \varphi_a]$, φ_a 为

$$\varphi_a = \underset{\varphi_j}{\operatorname{argmin}}(\operatorname{dif}(\varphi_j, \angle \boldsymbol{\nu}_i[k])) \quad (13)$$

其中 φ_j 为 $\underline{\varphi}_j$ 或 $\overline{\varphi}_j$ 。

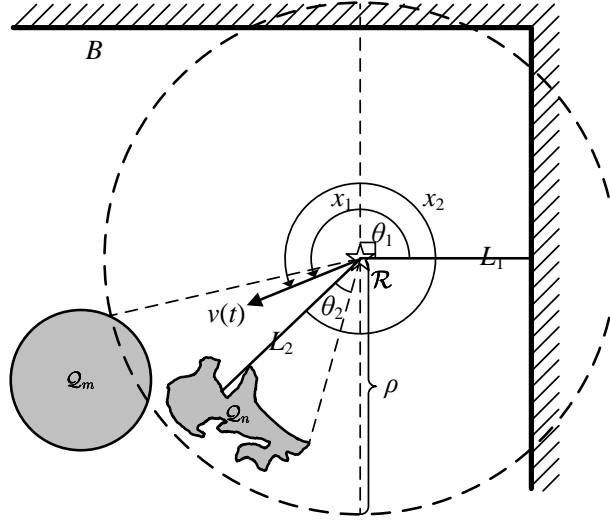


图 6: 多角度危险角度有交叠的情况

图6中, 个体 \mathcal{R} 遭遇了障碍物 Q_m 的可见部分 $\bar{S}_m[k] = \Phi[k] \cap S_m[k]$ 、 Q_n 的可见部分 $\bar{S}_n[k] = \Phi[k] \cap S_n[k] = S_n[k]$ 和边界 B 的可见部分 $\bar{B}[k]$ 。

由于 $\bar{S}_m[k]$ 和 $\bar{S}_n[k]$ 的危险角度范围有交叠, 故将这两个障碍合并计算。

那么 $\bar{B}[k]$ 对应的参考线为 L_1 、安全角度 θ_1 、角度 x_1 、表面与个体最近点距离 d_1 , $\bar{S}_m[k]$ 和 $\bar{S}_n[k]$ 共同对应的参考线为 L_2 、安全角度 θ_2 、角度 x_2 、表面与个体最近点距离 d_2 。

后面部分的计算与图5相同。

二. 项目架构说明

2.1. 项目中各模块间数据交互

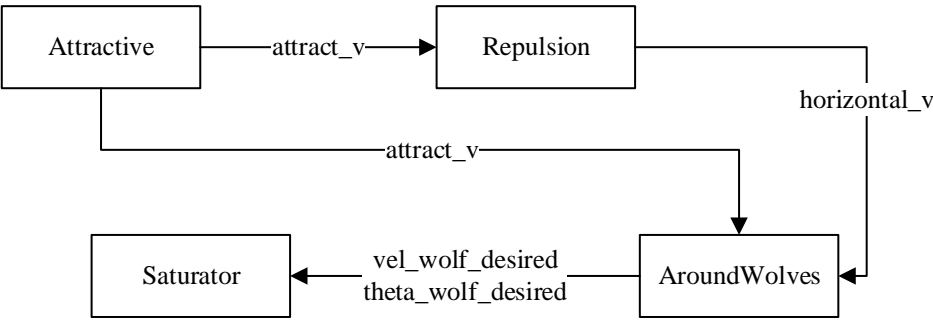


图 7: 各模块间数据交互图

表 1: 重要参数对照列表

变量	数学符号	物理意义	定义位置	单位
attract_v	ν_{i1}	前进速度	utils/robots_control.py 第586行	m/s
horizontal_v	ν_{i2}	排斥速度	utils/robots_control.py 第588行	m/s
vel_wolf_desired	ν	机器人期望速度	utils/robots_control.py 第594行	m/s
theta_wolf_desired	$\angle \nu$	机器人期望方向	utils/robots_control.py 第594行	rad

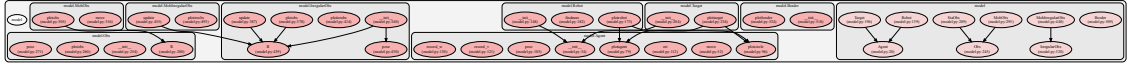


图 8: model.py文件

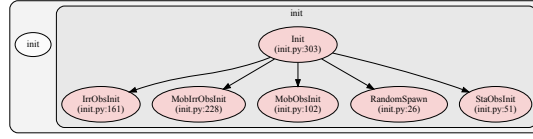


图 9: utils/init.py文件

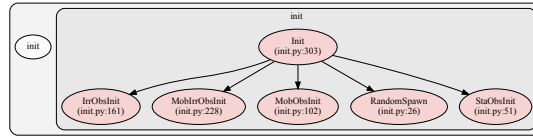


图 10: utils/init.py文件

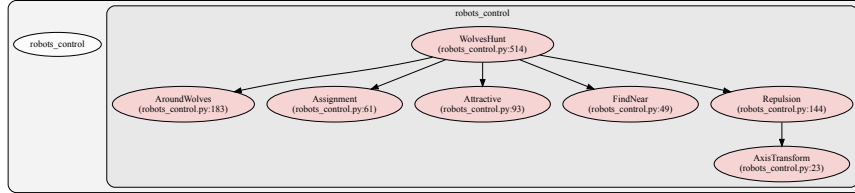


图 11: utils/robots_control.py文件

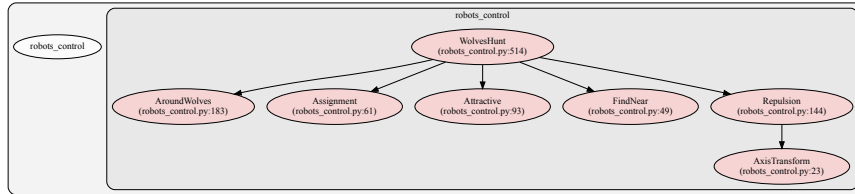


图 12: utils/robots_control.py文件

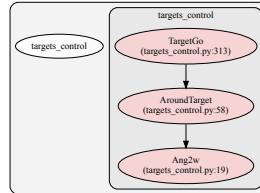


图 13: utils/targets_control.py文件

2.2. 文件结构说明

```
hunt_handover/
- docs/
  - 说明文档.pdf # 算法解释及代码说明文档
  - 文件结构.md # 用于解释文件结构的文档，其截图已包含在说明文档.pdf中
- example/
  - Figure_1.png # 仿真样例图片，README.md中调用
- output/
  - 2021_04_22.mp4 # 仿真视频样例
  - interact_matrix.txt # 仿真过程中的拓扑矩阵输出
  - wolves_ang_vel.txt # 仿真过程中围捕机器人的角速度输出
  - wolves_energy.txt # 仿真过程中围捕机器人的累计能量消耗输出
  - wolves_pos.txt # 仿真过程中围捕机器人的全局坐标输出
  - wolves_vel.txt # 仿真过程中围捕机器人的线速度输出
- utils/
  - control_input.py # 定义控制输入计算函数
  - determine.py # 定义判断围捕过程中围捕机器人是否撞上障碍物的函数
  - draw_data.py # 定义一些函数用于记录数据和将数据变化曲线图绘出
  - draw.py # 定义用于画仿真图的函数
  - init.py # 定义初始化地图和所有围捕机器人、目标、障碍物的函数
  - math_func.py # 定义一些用于数学计算的函数
  - move.py # 定义实现围捕机器人和目标移动的函数
  - read_yaml.py # 定义用于读取初始化参数存放的yaml文件的类
  - relative_pos.py # 定义用于计算一些对象间的向量差以及机器人和目标到边界距离的函数
  - robots_control.py # 定义根据算法计算出围捕机器人下一步的速度和角速度的函数
  - targets_control.py # 定义根据算法计算出目标下一步的速度和角速度的函数
  - updateobs.py # 定义实现障碍物移动和旋转的函数
- model.py # 定义围捕机器人、目标、各种障碍物及边界的类
- params.yaml # 外部初始化参数
- README.md # 快速上手文档(markdown格式)
- README.pdf # 快速上手文档(pdf格式)
- run.py # 程序运行的主函数
```

图 14: 文件结构图

三. 代码详细说明

3.1. 关键常量、函数的对照列表

表 2: 重要常量对照列表

常量	数学符号	物理意义	定义位置	单位
Ts	T_s	采样时间	params.yaml/Ts	s
radius	r_h	围捕半径	run.py/parameter['radius']	m
VISION_R	r_w	观察距离	model.py/Robot.VISION_R	m
AVOID_DIST	r_a	避障距离	model.py/Robot.AVOID_DIST	m
alpha	α	前进速度系数	run.py/parameter['alpha']	
varsigma	ς	前进速度系数	run.py/parameter['VARSIGMA']	
d1_ind	τ_1	前进速度系数	run.py/parameter['d1_ind']	
d2_ind	τ_2	前进速度系数	run.py/parameter['d2_ind']	
d3_ind	τ_3	排斥速度系数	run.py/parameter['d3_ind']	
beta	β	排斥速度系数	run.py/parameter['beta']	

表 3: 关键函数对照列表

函数	对应公式	基本作用
utils.math_func.PeriArctan	(1)	将向量转换为 $[0, 2\pi)$ 角度
utils.math_func.IncAngle	(2)	计算两个方向角间夹角
model.Agent.move	(3)	运动学方程
utils.robots_control.Attractive	(5)	计算前进速度
utils.robots_control.Repulsion	(6)	计算同伴排斥速度
utils.robots_control.WolvesHunt	(9)	围捕机器人运动的主函数
utils.robots_control.AroundWolves	(11)(12)(13)	围捕机器人的避障算法

3.2. 初始化文档说明

表 4: 初始化变量说明

变量	取值	含义
TOTSTEP	300	仿真步数(单位为step)
Ts	0.1	采样时间(单位为s/step)
WOLF_NUM	5	围捕机器人数量
TARGET_NUM	1	目标数量
S_OBS_NUM	1	固定圆形障碍物数量
M_OBS_NUM	1	移动圆形障碍物数量
IRR_OBS_NUM	5	固定不规则障碍物数量
M_IRR_OBS_NUM	5	移动不规则障碍物数量
border	$[-10, -10, 10, 10]$	边界(单位为m)
INIT_D	1.5	初始化障碍物表面间应有距离 (单位为m)