## 龙鱼航线规划算法接口说明文档

* 通过NativeHelper类调用各函数接口获取轨迹算法

1. **航点任务轨迹轨迹算法**

//飞机当前位置

double[] drone = new double[]{22.59651, 113.9972969, 0};//经纬高

//返航点位置

double[] homePoint = new double[]{22.59651, 113.9972969, 100.0};//经纬高

//上升盘旋点

double[] upHomePoint = new double[]{22.59651, 113.99434723115584, 100.0, 120.0};//经、纬、高、盘旋半径

//下降盘旋点

double[] downHomePoint = new double[]{22.59651, 114.00024656884415, 100, 120.0};//经、纬、高、盘旋半径

/\*\*

\* waypointParams：航点参数每16个值为一组，以下是以两个航点为例子；

参数说明：航点定义根据接口协议有16个变量，分别为：

航点定义根据接口协议有16个变量，分别为：

变量 0：当前航点标识（目前等于航点在当前任务中的序号）

变量 1：当前航点类型，其中：0–普通航点/飞越;1-兴趣点Orbit;4–起飞航 点;5–按时间盘旋航点;6-按圈数盘旋航点;7–降落航点

变量 2：航点坐标，纬度

变量 3：航点坐标，经度

变量 4：航点坐标，高度

变量 5：航点飞行速度，单位米/秒

变量 6：盘旋时间或盘旋圈数，只针对航点类型为盘旋有用

变量 7：盘旋半径，单位：米

变量 8：盘旋方向：0-顺时针;1-逆时针盘旋

变量 9：兴趣点起始角度 1-360度

变量10：兴趣点水平角度 1-360度

变量11：相机动作类型: 0-无，1-拍照，2-定时拍照，3-定距拍照，4-录像

变量12：相机动作参数，定时(单位s)和定距（单位m）的参数

变量13：相机动作参数，云台俯仰角（-120 -- 0）

变量14-15：未定义

\*/

double[] waypointParams = new double[]{1.0, 0.0, 22.59794923247847, 113.9946704742452, 100.0, 17.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,2.0, 0.0, 22.593907884795755, 113.99646218984662, 100.0, 17.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

PathPlanningResult result = NativeHelper.getWaypointMissionPath(drone, homePoint, upHomePoint, downHomePoint, waypointParams);

int errorCode = result.getErrorCode();//是否规划任务成功，0-成功，1-失败

double flyLength = result.getFlyLength();//航线总距离

double flyTime = result.getFlyTime();//预计飞行总时间

double pictNum = result.getPictNum();//预计拍照数量

double optCourseAngle = result.getOptCourseAngle();//自动规划主航线角 度时使用的主航线角度

List<AutelCoordinate3D> latLngList = result.getLatLngList();//整条航线所有 点的纬经高

List<DirectionLatLng> directionLatLngList = result.getDirectionLatLngList();//航线中箭头的纬经度

List<DistanceModel> distanceModelList = result.getDistanceModelList();// 航线中两个航点的距离的显示位置的纬度、经度、距离

List<AutelCoordinate3D> plusList = result.getPlusList();//两个航点间加号的 纬度、经度

**二、矩形或者多边形任务算法库接口**

//飞机当前位置

double[] drone = new double[]{22.59651, 113.9972969, 0};//纬经高

//返航点位置

double[] homePoint = new double[]{22.59651, 113.9972969, 100.0};//纬经高

//上升盘旋点

double[] upHomePoint = new double[]{22.59651, 113.99434723115584, 100.0, 120.0};//纬、经、高、盘旋半径

//下降盘旋点

double[] downHomePoint = new double[]{22.59651, 114.00024656884415, 100, 120.0};//纬、经、高、盘旋半径

//途经点1 （上升盘旋点到任务之间添加）

double[] startAvoid = new double[]{22.595300191562032, 113.98885025388489, 100, 1};//纬、经、高、是否有效（0- 无效，1-有效）

//途经点2

double[] endAvoid = new double[]{22.592050563109837, 113.99623427307421, 100, 1};//纬、经、高、是否有效（0- 无效，1-有效）

//长度为8（两个点），如果没有可以全设为0，只用于矩形和多边形， 矩形/多边形与上升下降盘旋点之间的点的纬度、经度、高度、是否 使用该航点(0-使用，1-不使用)

double[] avoidPoints = Arrays.copyOf(startAvoid, startAvoid.length + endAvoid.length);

//将b数组添加到已经含有a数组的c数组中去

System.arraycopy(endAvoid, 0, avoidPoints, startAvoid.length, endAvoid.length);

//矩形或多边形顶点坐标(经、纬、高)

double[] vertexs = new double[]{22.603459238667625, 113.99525530891242, 100.0

, 22.603459238667625, 113.9972294147372, 100.0

, 22.601993332010267, 113.9972294147372, 100.0

, 22.601993332010267, 113.99525530891242, 100.0};

//航线高度

float height = 100f;

//航线速度

float speed = 17.0f;

//旁向重叠率

double sideRate = 0.8f;

//主航线重叠率

float courseRate = 0.7f;

//主航线角度 0:自动，1：用户自定义航向角度

int userDefineAngle = 0;

//当userDefineAngle为1时有效

int courseAngle = 30;

//飞机转弯半径，默认要设置120

int turningRadius = 120;

//旁向扫描宽度

double sideScanWidth = 140.56235f;//2\*height\*tan(HFOV/2)需要 自行计算得出

//航向扫描宽度

double courseScanWidth = 78.98377f;//2\*height\*tan(VFOV/2)

PathPlanningResult result = NativeHelper.getMappingMissionPath(drone, homePoint, upHomePoint, downHomePoint,

vertexs, avoidPoints, height, speed, sideRate, courseRate

, userDefineAngle, courseAngle, turningRadius

, sideScanWidth, courseScanWidth);

double area = result.getArea();//矩形，多边形的面积

double flyLength = result.getFlyLength();//航线总距离

double flyTime = result.getFlyTime();//预计飞行总时间

double pictNum = result.getPictNum();//预计拍照数量

double optCourseAngle = result.getOptCourseAngle();//自动规划 主航线角度时使用的主航线角度

List<AutelCoordinate3D> whiteLatLngList = result.getWhiteLatLngList();//矩形/多边形区域内转折点的纬经 高

List<AutelCoordinate3D> latLngList = result.getLatLngList();//整条 航线所有点的纬经高

List<DirectionLatLng> directionLatLngList = result.getDirectionLatLngList();//航线中箭头的纬经度

List<DistanceModel> distanceModelList = r esult.getDistanceModelList();//航线中两个航点的距离的 显示位置的纬度、经度、距离

List<AutelCoordinate3D> plusList = result.getPlusList();//两个航点 间加号的纬度、经度

**三、将航线任务保存到文件中**

//先创建文件保存目录

File myDir = new File(FileUtils.getMissionFilePath());

if (!myDir.exists()) {

myDir.mkdirs();

}

double missionType = 1;//任务类型，1-航点任务，6-矩形/多边形任 务

//长度/高度单位均为米

//长度为3，飞机的纬度、经度、起飞高度

double[] droneLocation = new double[]{22.59638835580453, 113.99613850526757, 40.0};

//长度为3，返航点的纬度、经度、返航高度

double[] homeLocation = new double[]{22.59638835580453, 113.99613850526757, 50.0};

//长度为4，上升盘旋点的纬度、经度、高度、盘旋半径

double[] launchLocation = new double[]{22.59638835580453, 113.99318883642341, 100.0, 120.0};

//长度为4，下降盘旋点的纬度、经度、高度、盘旋半径

double[] landingLocation = new double[]{22.59291695879857, 113.99787910849454, 100.0, 120.0};

//长度为8（两个点），如果没有可以全设为0，只用于矩形和多边形， 矩形/多边形与上升下降盘旋点之间的点的纬度、经度、高度、 是否使用该航点(0-使用，1-不使用)

double[] avoidPosition = new double[]{22.598295333564423, 113.99354868480384, 100.0, 1.0,

22.598772827314363, 113.99867325644607, 100.0, 1.0};

char waypointLen = 2;//航点的个数/矩形多边形是顶点的个数

int poiPointLen = 2;//观察点的个数

//以下参数针对矩形、多边形任务,航点任务时全置为 0 就可以了

double UAVTurnRad = 120;//飞机转弯半径，默认 120 米

double UAVFlyVel = 17;//飞行速度(单位m/s)

double UserFPKIsDef = 1;//是否用户自定义主航线角度，0-自动， 1-手动

double UserFlyPathA = 0;//用户自定义主航线角度，UserFPKIsDef 为1时生效

double WidthSid = 140.56;//旁向扫描宽度,//2\*height\*tan(HFOV/2) 需要自行计算得出

double OverlapSid = 0.7;//旁向重叠率（0-1）

double WidthHead = 78.984;//航向扫描宽度,//2\*height\*tan(VFOV/2) 需要自行计算得出

double OverlapHead = 0.8;//航向重叠率（0-1）

double UAVFlyAlt = 100;//飞行高度

/\*

航点定义根据接口协议有16个变量，分别为：

变量 0：当前航点标识（目前等于航点在当前任务中的序号）

变量 1：当前航点类型，其中：0–普通航点/飞越;1-兴趣点 Orbit;4–起飞航点;5–按时间盘旋航点;6-按圈数 盘旋航点;7–降落航点

变量 2：航点坐标，纬度

变量 3：航点坐标，经度

变量 4：航点坐标，高度

变量 5：航点飞行速度，单位米/秒

变量 6：盘旋时间或盘旋圈数，只针对航点类型为盘旋有用

变量 7：盘旋半径，单位：米

变量 8：盘旋方向：0-顺时针;1-逆时针盘旋

变量 9：兴趣点起始角度 1-360度

变量10：兴趣点水平角度 1-360度

变量11：相机动作类型: 0-无，1-拍照，2-定时拍照，3-定距拍 照，4-录像

变量12：相机动作参数，定时和定距的参数

变量13：相机动作参数，云台俯仰角（-120 -- 0）

变量14-15：未定义

\*/

//航点任务

double[] waypointParamList = new double[]{1.0, 0.0, 22.597737289727164, 113.9974874391902, 100.0, 17.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -90.0, 0.0, 0.0,

2.0, 0.0, 22.59897542587946, 114.00336684129968, 100.0, 17.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -90.0, 0.0, 0.0};

/\*

航点定义根据接口协议有17个变量，分别为：

变量 0：纬度

变量 1：经度

变量 2：高度

变量 3：半径

变量 4：IP\_Type，默认 11

变量 5：关联航点个数

\*/

double[] poiParamList = new double[]{22.601550713371807, 113.99913365283817, 0.0, 120.0, 11.0, 1.0,

22.600490797193245, 113.99435713952568, 20.0, 120.0, 11.0, 0.0};

//关联航点序号列表，每个观察点最多关联五个航点，数组个数为观 察点个数\*5

int[] linkPoints = new int[]{2, 0, 0, 0, 0, 0, 0, 0, 0, 0};

//是否使用地形跟随

boolean isEnableTopographyFollow = true;

//返回0表示成功，返回非0表示失败

int res = NativeHelper.writeMissionFile(filePath, missionType,

droneLocation, homeLocation,

launchLocation, landingLocation,

avoidPosition, UAVTurnRad,

UAVFlyVel, UserFPKIsDef,

UserFlyPathA, WidthSid,

OverlapSid, WidthHead,

OverlapHead, UAVFlyAlt,

waypointLen, waypointParamList,

poiPointLen, poiParamList, linkPoints, isEnableTopographyFollow ? 1 : 0);

AutelLog.d("NativeHelper", " writeMissionFile result -> " + res);