

竞赛无人机基本自动飞行支持函数与导航控制函数解析

- 基本自动飞行支持函数

`void basic_auto_flight_support(void)`

```
23 //基本飞行保障必备子函数一定高、定点
24 void basic_auto_flight_support(void)
25 {
26     OpticalFlow_Control_Pure(0); //SLAM定点控制
27     Flight.yaw_ctrl_mode=ROTATE; //偏航控制为手动模式
28     Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
29     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL); //高度控制
30 }
31
```

根据前面几讲的介绍，要想实现无人机的自动飞行，单依靠姿态自稳、高度控制远远不够，必须在水平位置—速度控制的基础上，才能保证无人机全程可控。因此设计了一个基本自动飞行支持函数，

第一部分 `OpticalFlow_Control_Pure()` 为 SLAM 定点控制，该函数在最后给定了水平俯仰、横滚方向上的角度期望；第二部分 `yaw_ctrl_mode=ROTATE`，`yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW]` 表示偏航控制为手动模式，偏航控制期望来源于遥控器偏航杆位给定；第三部分 `Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL)` 为高度手动控制。

以上三部分共同保障了无人机自动飞行的基础，在 `basic_auto_flight_support` 运行的基础上，可以通过改变三维的位置期望来实现无人机三维位置控制，因此专门设计了一个导航控制函数用于处理三维期望数据。

- 导航控制函数

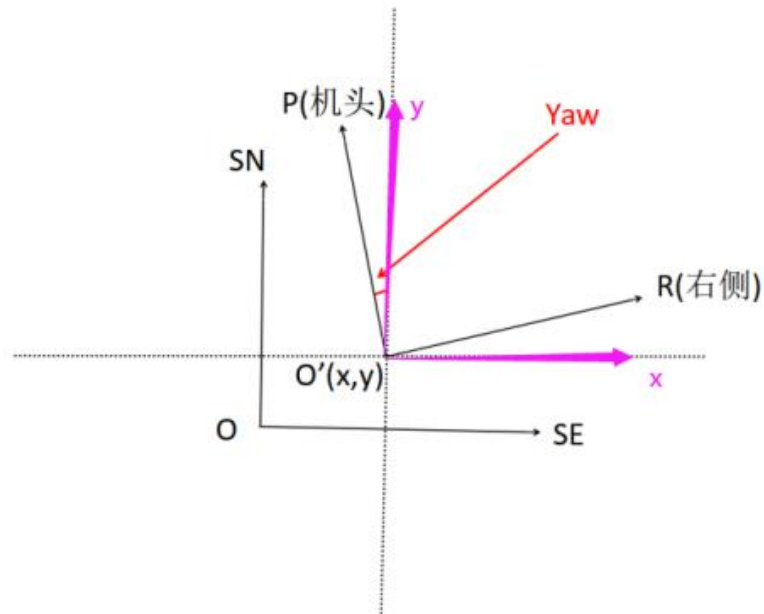
`void Horizontal_Navigation(float x,float y,float z,uint8_t nav_mode,uint8_t frame_id)`

前三组参数表示下 `xyz` 三个方向的位置期望，单位为 `cm`，`xy` 实际表征意义与 `nav_mode`、`frame_id` 相关，下面列出二者的可选枚举值：

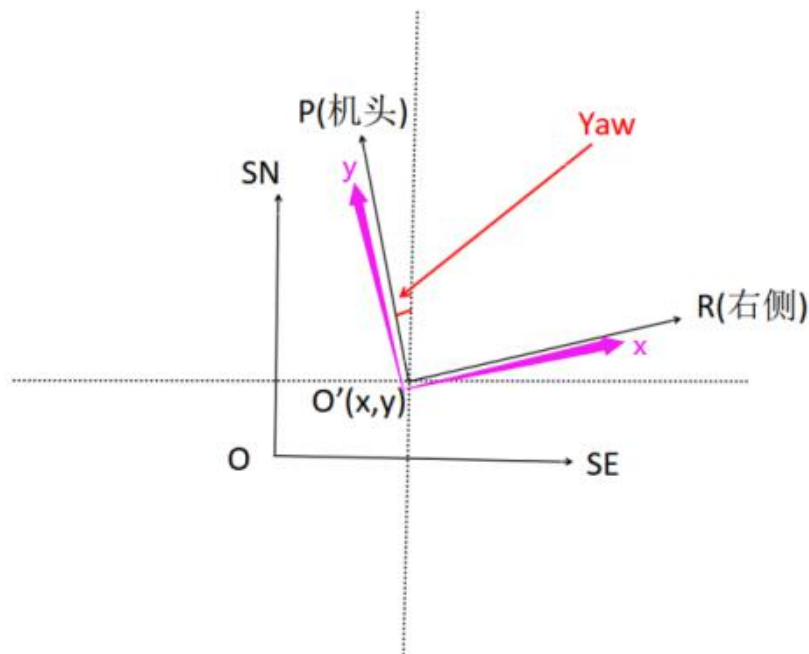
```
341 typedef enum
342 {
343     BODY_FRAME=0, //机体坐标系
344     MAP_FRAME,    //导航坐标系
345 } Navigation_Frame;
346
347 typedef enum
348 {
349     RELATIVE_MODE=0, //相对模式
350     GLOBAL_MODE,    //全局模式
351 } Navigation_Mode;
```

① 当 `nav_mode` 为 `RELATIVE_MODE`，`frame_id` 为 `MAP_FRAME` 时，`xy` 表示在当前位置的基础上，向虚拟正东(西)、虚拟正北(南)移动一定距离，`z` 表示以当前高度为基准向上或者向下移动一定距离。

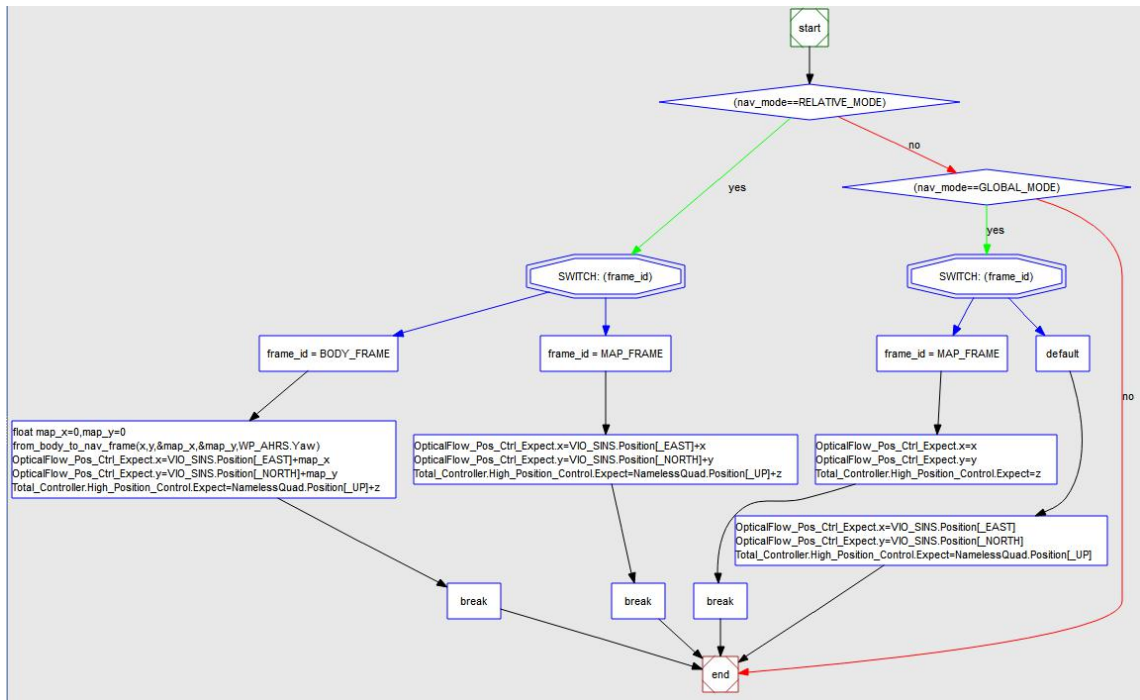
无名创新



② 当 nav_mode 为 RELATIVE_MODE, frame_id 为 BODY_FRAME 时, xy 表示在**当前位置的基础上**, 向无人机的右侧(R)、机头(P)移动一定距离, z 表示以**当前高度为基准**向上或者向下移动一定距离。



③ 当 nav_mode 为 GLOBAL_MODE, frame_id 为 MAP_FRAME 时, xy 表示以**初始 O 为原点**, 虚拟正东(西)、虚拟正北(南)方向的**全局期望位置**, z 表示**相对地面**的高度期望值。



上述函数用法学习可以参考 Subtask_Demo.c 提供的 flight_subtask_5、flight_subtask_6、flight_subtask_7 来练习掌握用法。

```

206 //本demo适用于激光雷达SLAM定位条件下，普通光流（LC307、LC302）定位条件下无效
207 //机体坐标系下相对位移
208 //右前上分别对应XYZ正方向
209 void flight_subtask_5(void)
210 {
211     static uint8_t n=4;
212     if(flight_subtask_cnt[n]==0)
213     {
214         basic_auto_flight_support();//基本飞行支持软件①
215         flight_subtask_cnt[n]=1;
216         execute_time_ms[n]=10000/flight_subtask_delta;//子任务执行时间
217         //向前移动100cm
218         Horizontal_Navigation(0,100,0,RELATIVE_MODE,BODY_FRAME);②
219     }
220     else if(flight_subtask_cnt[n]==1)
221     {
222         basic_auto_flight_support();//基本飞行支持软件
223         if(execute_time_ms[n]>0) execute_time_ms[n]--;
224         if(execute_time_ms[n]==0)
225         {
226             flight_subtask_cnt[n]=2;
227             execute_time_ms[n]=10000/flight_subtask_delta;//子任务执行时间
228             //向右移动100cm
229             Horizontal_Navigation(100,0,0,RELATIVE_MODE,BODY_FRAME);
230         }
231     }
232     else if(flight_subtask_cnt[n]==2)
233     {
234         basic_auto_flight_support();//基本飞行支持软件
235         if(execute_time_ms[n]>0) execute_time_ms[n]--;
236         if(execute_time_ms[n]==0)
237         {
238             flight_subtask_cnt[n]=3;
239             execute_time_ms[n]=10000/flight_subtask_delta;//子任务执行时间

```

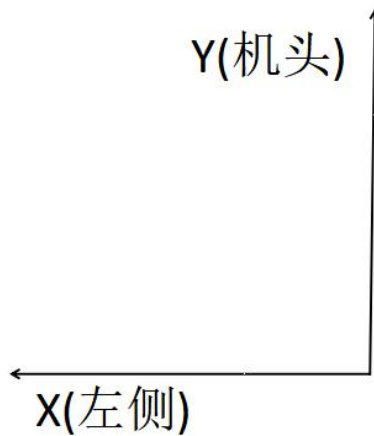
用户需要对无人机进行三维位置控制时，基本自动飞行支持函数与导航控制函数是配合使用的，二者缺一不可。新手在二次开发时往往只对飞行航点任务进行编写，却遗漏掉了自动飞行支持函数。

● 其它控制函数

void OpticalFlow_Vel_Control(Vector2f target)

void OpticalFlow_X_Vel_Control(float target_x)//机头左侧为 X+

void OpticalFlow_Y_Vel_Control(float target_y)//机头前侧为 Y+



上面两个为光流速度控制函数，在仅需要速度控制的情况下，比如定巡航、自主循迹、自动绕杆、靠近塔杆等条件下使用。相关用法参考飞控开发者模式中的相关 demo 历程。

```
1501 else if(flight_subtask_cnt[n]==35)//首先控制偏航运动，使得飞机头部对准塔杆
1502 {
1503     float expect_yaw_gyro=Total_Controller.Yaw_Angle_Control.Kp*target_yaw_err;//期望偏航角速度
1504     Flight.yaw_ctrl_mode=ROTATE;
1505     Flight.yaw_outer_control_output=constrain_float(expect_yaw_gyro,-10,10);//以10deg/s的角速度顺时针转动10000ms
1506     if (ABS(target_yaw_err)<=5.0f) //只有当偏航角度比较小时，才靠近杆
1507     {
1508         float dis_err=min_dis_cm-50;
1509         dis_err=constrain_float(dis_err,-20,20);
1510         OpticalFlow_Y_Vel_Control(Total_Controller.Optical_Position_Control.Kp*dis_err);
1511         OpticalFlow_X_Vel_Control(0);
1512         fix_flag=1;
1513     }
1514     else//否则原地悬停
1515     {
1516         OpticalFlow_Control_Pure(fix_flag);
1517         fix_flag=0;
1518     }
1519     Flight_Alt_Hold_Control(ALTHOLD_MANUAL_CTRL,NUL,NUL);//高度控制
1520
1521     if(barcode_flag==1)//只要识别到了二维码，提前结束对准塔杆、靠近任务
1522     {
1523         flight_subtask_cnt[n]=36;
1524         flight_global_cnt[n]=0;
1525     }
1526 }
1527 }
```

```
OpticalFlow_Control.c Subtask_Demo.c WP_DataType.h
759 PID_Control_SDK_Err_LFF($Total_Controller.SDK_Roll_Position_Control,Opv_Top_View_Target.trust_flag);
760 OpticalFlow_Pos_Ctrl_Output.x=-Total_Controller.SDK_Roll_Position_Control.Control_OutPut;
761
762 static float target_speed;//5 3 2 1
763 if (ABS(Opv_Top_View_Target.sdk_angle)<=10) target_speed=8;//15
764 else if (ABS(Opv_Top_View_Target.sdk_angle)<=20) target_speed=5;//10
765 else if (ABS(Opv_Top_View_Target.sdk_angle)<=50) target_speed=3;//5
766 else target_speed=1;//5 3 2 1
767
768 OpticalFlow_Pos_Ctrl_Output.y=target_speed;
769
770 else//丢失目标
771 {
772     miss_flag=1;
773 }
774
775 if(miss_flag==1)//目标丢失
776 {
777     if(miss_cnt==1)//初始丢失跟踪目标后，锁定当前位置后，进行普通光流控制
778     {
779         miss_cnt=2;
780         OpticalFlow_Pos_Ctrl_Output.x=0;
781         OpticalFlow_Pos_Ctrl_Output.y=0;
782         OpticalFlow_Control_Pure(1);
783     }
784     else if(miss_cnt==2)//丢失跟踪目标后，进行普通光流控制
785     {
786         OpticalFlow_Control_Pure(0);
787         Flight.yaw_ctrl_mode=ROTATE;
788         Flight.yaw_outer_control_output =RC_Data.rc_rpyt[RC_YAW];
789     }
790 }
791
792 else//目标未丢失
793 {
794     OpticalFlow_Vel_Control(OpticalFlow_Pos_Ctrl_Output);//速度控制周期20ms
795     Flight.yaw_ctrl_mode=ROTATE;
796     Flight.yaw_outer_control_output =-3.0f*Opv_Top_View_Target.sdk_angle;
797 }
798
799 }
800
801 ...
```