Pos_control标志量

初始化:

reset pos sp = true, reset_alt_sp = true, 定高定点中,将当前位置设置为期望位

置,即不通过位置计算期望速度

_pos_hold_engaged = flase _alt_hold_engaged = flase, 杆在死区中才置为true

_run_alt_control = true,

control_vel_enabled_prev = flase

从手动或定高,切到定点,根据姿态计算 期望速度,保证姿态连续

bool reset_int_z = true; 定高定点模式起飞后将z积分量置为0.1

bool reset_int_xy = true;

定点起飞,或者手动定高切到定点,将xx 积分量置为0

bool reset_int_z_manual = false; 从手动切到定高定点的时候调用,将z积分

量置为0.5

bool reset_yaw_sp = true; 解锁的时候置为true,随后将当前航向设置为期望航向

通用: >>>>>>>>

飞机解锁,重置标志量 reset_alt_sp = true;

reset_int_z = true; reset int xy = true;

reset_yaw_sp = true;

非定点定高模式:

非定高模式:

定点或定高模式:

_run_pos_control = true; _run_alt_control = true;

进入Control_manual:

定点或定高调用reset_alt_sp(): 定点模式调用reset_pos_sp():

if (<u>reset_pos_sp</u>) { pos sp(0) = pos(0);_pos_sp(1) = _pos(1);

·>>>> 定点模式进入,判断是否需要保持当前位置: 当roll/pitch杆处于死区中的时候:

当前速度小于某个阈值:

当前速度依旧很大: pos hold er

当roll/pitch杆不在死区中:

if (<mark>!_pos_hold_engaged</mark>) { 此处不保持位置,则根据杆量设定速度

_pos_sp(0) = _pos(0); _pos_sp(1) = _pos(1); _run_pos_control = false; _vel_sp(0) = req_vel_sp_scaled(0); _vel_sp(1) = req_vel_sp_scaled(1);

.....

·>>>>> 定高模式进入,判断是否需要保持当前高度:

如果thrust杆处于死区中: if (!_alt_hold_engaged) {

当前Z方向速度小于某个值: alt hold engaged = true; 当前Z方向的速度值很大: _alt_hold_engaged = false;

如果thrust杆不在死区:

_pos_sp(2) = _pos(2); if (!_alt_hold_engaged) { 此处不保持高度,根据杆量设置速度

_run_alt_control = false;

_vel_sp(2) = req_vel_sp_scaled(2); <<<<<<<<<<<<<<<<<<

·>>>>> 定高定点模式, 降落后

reset_pos_sp = true; reset_alt_sp = true;

reset int z = true; reset int xy = true

..... if (_run_pos_control) { 根据位置计算水平速度:

_vel_sp(0) = (_pos_sp(0) - _pos(0)) * _params.pos_p(0); _vel_sp(1) = (_pos_sp(1) - _pos(1)) * _params.pos_p(1); if (_run_alt_control) {

_vel_sp(2) = (_pos_sp(2) - _pos(2)) * _params.pos_p(2);

非定点模式: _reset_pos_sp = true;

根据高度计算垂直速度:

非定点定高模式:

定高模式: _vel_sp_prev(0) = _vel(0); _vel_sp_prev(1) = _vel(1); vel sp(0) = 0.0f;

手动模式,进入判定:

reset_alt_sp = true; reset_pos_sp = true;

reset_int_z = true; reset_int_xy = true;

control_vel_enabled_prev = false;

手动、定高、定点模式:

if (reset_yaw_sp) { 将期望航向置为当前航向 reset yaw sp = false; _att_sp.yaw_body = _yaw;

解锁后:

定高定点:

<<<<<<