

PPG Pixhawk 2 への実装: Ardu Plane ver.3.7.1.

山本 海斗 2021.07.04.更新

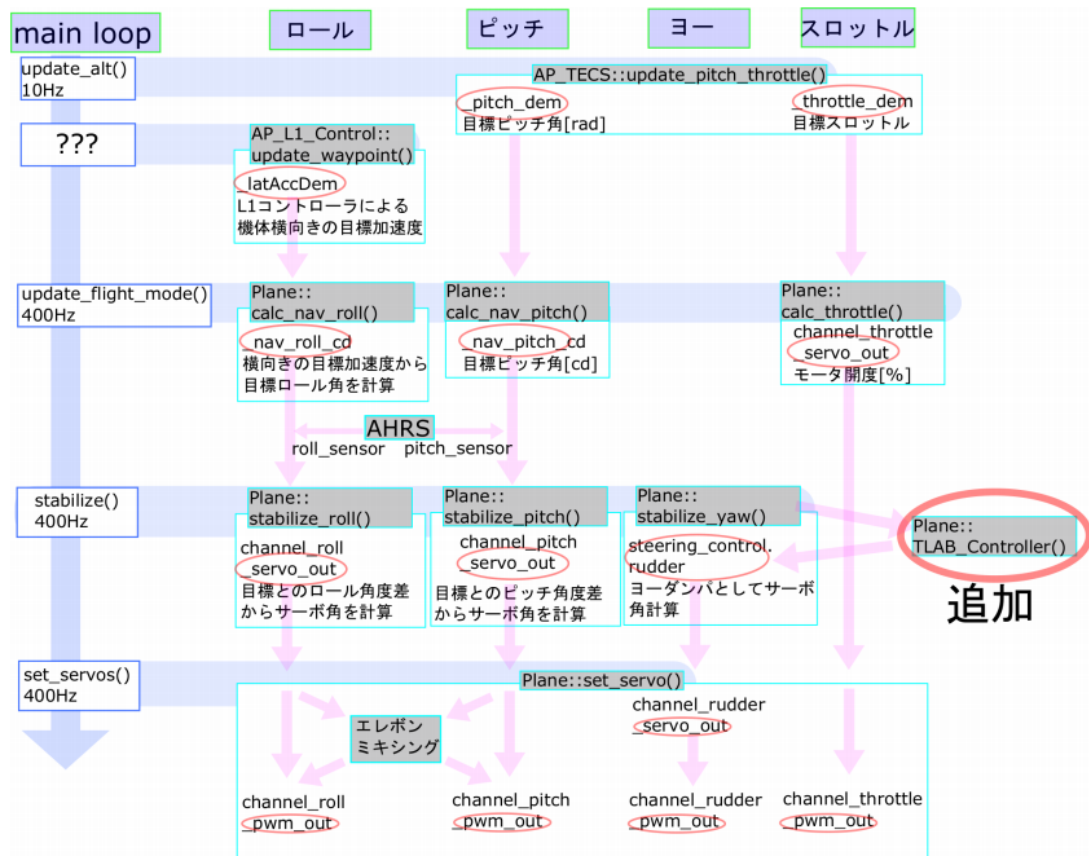


図1. 制御フロー (岩瀬, 2018 [1])

<”Attitude.cpp” 内の注目する関数>

■ `void Plane::stabilize_yaw(float speed_scaler)`

- PPG のコントロールバー角度を計算する関数。
- デフォルトのプログラムはコメントアウトしており、関数” `calc_nav_yaw_coordinated(float speed_scaler)`”の実行のみを行う。したがって、実質的には次に示す関数が本関数を代替している。

■ `void Plane::calc_nav_yaw_coordinated(float speed_scaler)`

- 関数” `stabilize_yaw(float speed_scaler)`”の代替で、PPG のコントロールバー角度 [cdeg.]を計算する関数。
- 引数: “float speed_scaler” (デフォルトで使用。本実装では使用しない。)

・”steering_control.rudder”の値を”Combine_Mode_flag”および”TLAB_Control_flag”に応じ
て切替えて計算・更新する。

・参照する関数:

・“TLAB_Line_Trace_Controller()”: 直線追従コントローラ

・“TLAB_Circle_Trace_Controller()”: 円追従コントローラ(右旋回)

・“TLAB_Combine_Controller()”: 円+直線追従コントローラ(組み合わせ経路)

	Combine_Mode_flag	TLAB_Control_flag
TLAB_Line_Trace_Controller()	0	0
TLAB_Circle_Trace_Controller()	0	1 (左旋回), 2 (右旋回)
TLAB_Combine_Controller()	1	#

■ void Plane::init_TLAB_Controller(void)

・起動時に一度だけ実行される関数。

・各パラメータ・変数の初期化を行う。

・”g.~”で記述されたものは、Mission Planner 上で設定可能なグローバル変数である。

■ int32_t Plane::TLAB_Circle_Trace_Controller(void)

・円経路追従コントローラ。コントロールバー角度 [cdeg.]の計算値を返す。

・目標円経路の中心点座標は、構造体”Location”で定義した変数”Target_Circle_Center”に
MP で設定可能なパラメータ(” g.TPARAM_center_lat”, “g.TPARAM_center_lng”)を代入する
ことで指定。

<パラメータの追加方法 [2]>

ここでいう、「パラメータ」とは Mission Planner 上で変更可能なグローバル変数(メイン関数
で使用可能な変数)のことを指す。

1. ”Parameters.h”内の”enum”クラスに新しいパラメータ名を追加。

2. ”Parameters.h”内の”enum”クラスの下に、先程のパラメータをデータ型と共に宣言する。

・設定可能な型は、AP_Int8, AP_Int16, AP_Float, AP_Int32, AP_Vector3 の 5 種類。

・変数名は、先程設定した変数名から接頭文字”k_param_”を除いたものを用いる。

3. ”Parameters.cpp”内に他に倣って、先程の変数を追加する。

・Mission Planner 上で使用するパラメータ名は、16 文字以内に設定すること(大文字+”_”
で記述するのが慣習)。

・”@~”で始まるコメントアウト文は、Mission Planner 上でユーザに表示される。

4. ”config.h”内で新しいパラメータのデフォルト値を指定する(デフォルト値を3.で直接指定し
ていれば必要なし)。

<データログ方法 [3]>

ログデータを追加するために編集が必要なのは、“defines.h”, “Plane.h”, “Log.cpp”の3つのファイル。詳細は、文献[3]を参考にする。

<関数および変数の追加方法>

1. 関数および変数の宣言: “Attitude.cpp”のヘッダファイル”Plane.h”内で宣言する。
2. 関数の定義(記述): “Attitude.cpp”内で定義。
3. 変数の初期化: 関数“init_TLA B_Controller()”内に記述。

<Eclipse でコードを読み書きする際の便利機能>

- ・「F2」: 関数・変数・構造体・クラス等の宣言のプレビューができる。
- ・「F3」: 関数・変数・構造体・クラス等の宣言場所へ飛べる。
- ・「Ctrl + F」: 文字探し(ファイル内をサーチ)
- ・「Ctrl + H」: 文字探し(より広範囲をサーチ)

＜経路追従プログラムの実装＞

・ポイント

- ・MP で変更可能なパラメータの設定
 - ・フィードバックゲイン F
 - ・目標経路の中心点慣性座標
 - ・対気速度 V_a の推定値
 - ・機体旋回速度とコントロールバー角度の比例定数 k
- ・慣性座標系からセレ・フレネ座標系への変換
- ・目標経路生成のプログラム: MATLAB の "F_PATH_PPG*.m"
 - ・MP から経路の切替を可能にする(直線・円・リサージュ)
- ・緯度経度 [deg.]を xy 座標系 [m]に変換: MATLAB の "LatLng2XY.m"
 - ・Ardupilot の標準関数 "location_diff()" を利用する ("location.cpp")

参考文献

- [1] 岩瀬惟真, 有理多項式ファジィ制御による無人航空機の経路追従安定化制御, 電気通信大学 修士論文, 2018
- [2] ArduPilot Dev Team, Adding a parameter to Copter, 2021
<https://ardupilot.org/dev/docs/code-overview-adding-a-new-parameter.html>
- [3] 青木颯太郎, ミーティング資料, 2021.03.29.