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

山本 海斗　2021.07.04.更新

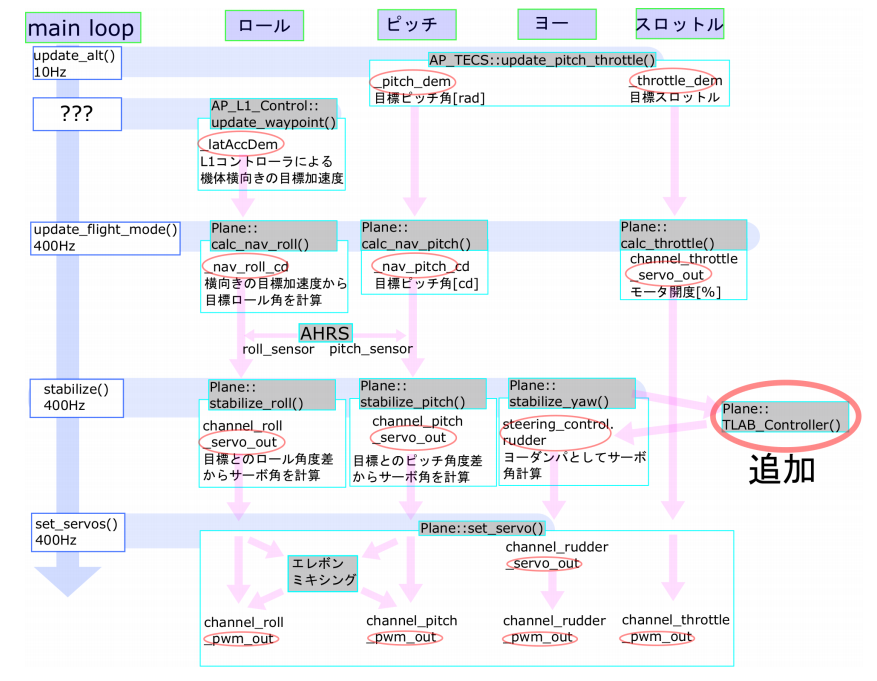


図１．制御フロー（岩瀬, 2018 [1]）

＜”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 Plannrer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。

・「Ctr + F」：　文字探し（ファイル内をサーチ）

・「Ctr + 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Va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： MALAB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.