

程式說明文件

一、專案介紹 – Description

本專案使用 UWB 進行無人機定位。其中定位分為靜態定位以及動態定位，其中流程可以分成兩個部分：接收距離量測數據、求解定位座標近似解。

二、運行環境需求 – Requirement and .env Setting

本專案在 UWB 靜態定位以及 UWB 動態定位(不包含卡爾曼濾波器)，使用 python 語言。

模組使用：

- numpy
- re
- math
- matplotlib.pyplot
- serial

在實時定位的部分，使用 arduino。

函式庫使用：

- MPU6050
- MatrixMath
- positioning

三、架構圖和目錄結構 – System architecture

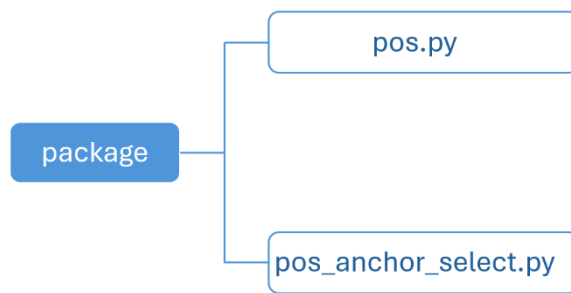
1. python 靜態定位

A. 主要程式

`pos_stationary.py`

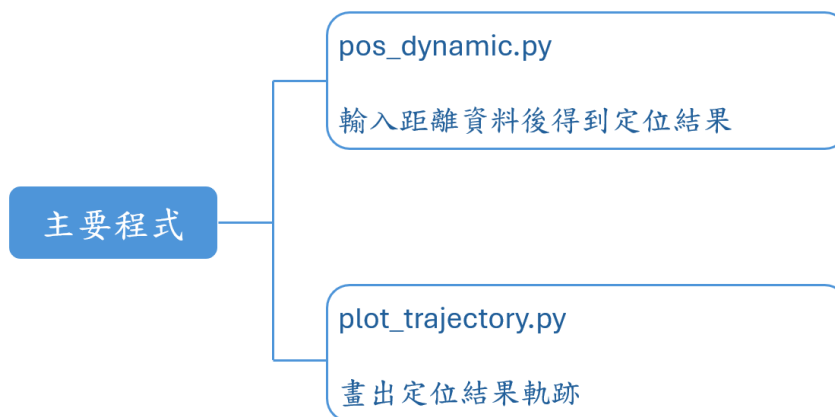
輸入距離資料後得到靜態定位結果

B. package

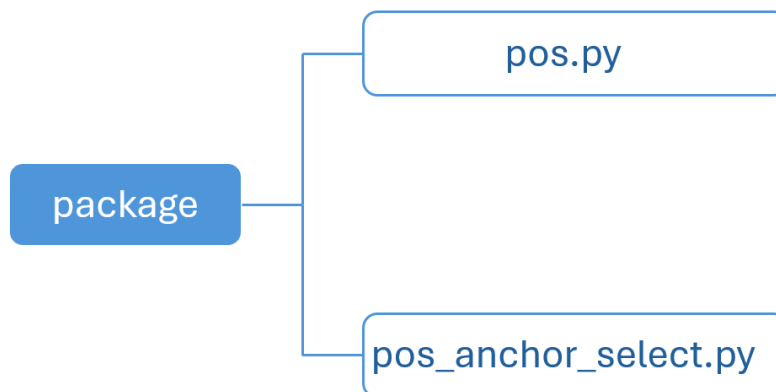


2. python 動態定位

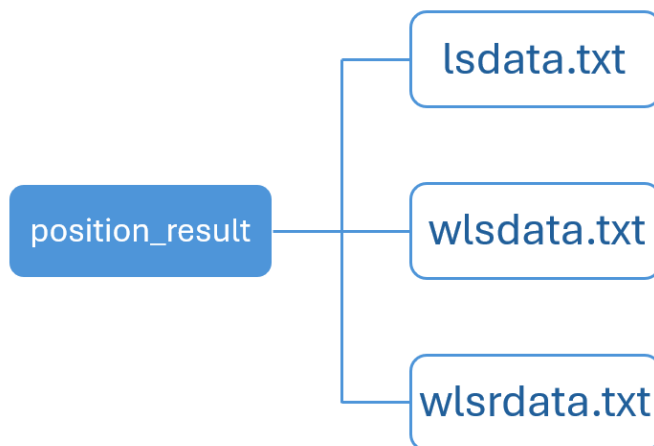
A. 主要程式



B. package



C. 產生的資料

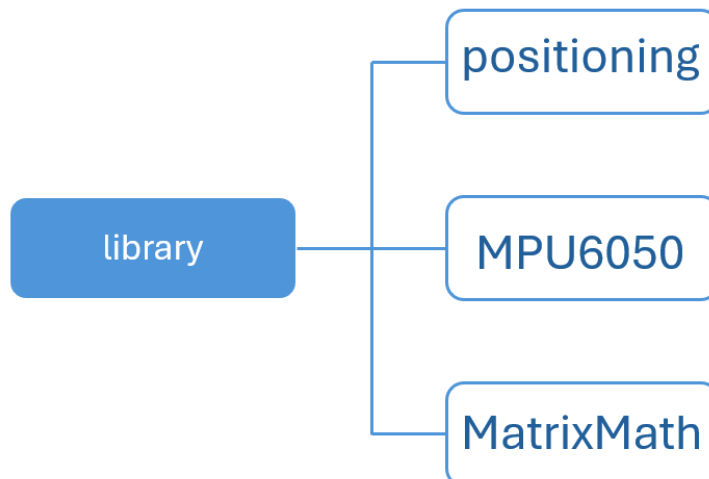


3. 實時定位(僅有卡爾曼濾波器部分，需搭配距離量測程式、求定位近似解程式、傳送 UWB 定位結果至卡爾曼濾波器程式)

A. 主要程式



B. library



四、 程式的簡要介紹 – Introduction to the Program

程式分為靜態定位以及動態定位，在不同定位情況下須分別使用對應的程式，在 python 靜態定位以及 python 動態定位中，僅有使用 UWB 進行定位，另外實時定位程式中，僅包含卡爾曼濾波器部分，以下按照不同定位情況分別說明。

1. python 靜態定位

A. 主要程式

`pos_stationary.py`

用途說明:

本程式將蒐集的距離資料分別經由最小平方法(Least Squares, LS)、加權最小平方法(Weighted Least Squares, WLS)、加權最小平方法以 R 修正(Weighted Least Squares corrected by R, WLS-R)解算定位座標，並分別顯示定位方均根誤差(Root Mean Squares Error, RMSE)。

所需檔案:

`"anchor_tag\\anchor_toa.txt"`:

輸入所有 anchor 的座標

每行由左至右分別是 x y z 座標值，由第一行開始按照 anchor 編號依序填入

`"anchor_tag\\tag_toa.txt"`

輸入要測量的測試點真實座標

每行由左至右分別是 x y z 座標值，由第一行開始按照測試點編號依序填入

`"collect_data\\uwb_datai.txt"`:

靜態中的 anchor 距離量測資料，*i* 表示第 *i* 個測試點的資料，其中距離資料在 `uwb_datai.txt` 中，須以 "`Ak:dm`" 表示，其中 *k* 表示第 *k* 個 anchor，*d* 表示其測量距離。

生成檔案:

本程式無生成資料，定位結果 RMSE 於終端機印出。

B. package 中的程式

`pos.py`

用途說明:

定義 LS、WLS、WLS-R 定位近似解方法函式。可於此程式中，改變不同 anchor 測距的誤差變異數。

`pos_anchor_select.py`

用途說明:

計算 anchor 以及測試點數量，並進行將過大測量誤差的 anchor 刪除的動作。

2. python 動態定位

A. 主要程式

`pos_dynamic.py`

用途說明:

將蒐集的移動中距離測距資料經由 LS、WLS、WLS-R 方法分別計算定位近似解，並生成定位結果文字檔。

所需檔案:

`"anchor_tag\\anchor_toa.txt"`:

輸入所有 anchor 的座標

每行由左至右分別是 x y z 座標值，由第一行開始按照 anchor 編號依序填入。

`"path_data\\uwb_datai.txt"`:

靜態中的 anchor 距離量測資料，*i* 表示第 *i* 個測試點的資料，其中距離資料在 `uwb_datai.txt` 中，須以 `"Ak:dm"` 表示，其中 *k* 表示第 *k* 個 anchor，*d* 表示其測量距離。

生成檔案:

`"position_result\\lsdata.txt"`:

經由 LS 方法得到的定位結果

`"position_result\\wlsdata.txt"`:

經由 WLS 方法得到的定位結果

`"position_result\\wlsrcdata.txt"`:

經由 WLS-R 方法得到的定位結果

`plot_trajectory.py`

用途說明:

在 `pos_dynamic.py` 執行後，將

`"position_result\\lsdata.txt"`、

`"position_result\\wlsdata.txt"`、`"position_result\\wlsrcda`

ta.txt”中的定位結果畫成定位軌跡。

所需檔案:

“position_result\\lsdata.txt”:

經由 LS 方法得到的定位結果

“position_result\\wlsdata.txt”:

經由 WLS 方法得到的定位結果

“position_result\\wlsrdata.txt”:

經由 WLS-R 方法得到的定位結果

B. package 中的程式

`pos.py`

用途說明:

定義 LS、WLS、WLS-R 定位近似解方法函式。可

於此程式中，改變不同 anchor 測距的誤差變異數。

`pos_anchor_select.py`

用途說明:

計算 anchor 以及測試點數量，並進行將過大測量誤差的 anchor 刪除的動作

3. 實時定位(僅有卡爾曼濾波器部分，需搭配距離量測程式、求定位近似解程式、傳送 UWB 定位結果至卡爾曼濾波器程式)

僅包含卡爾曼濾波器的 arduino 程式，需搭配額外 UWB 實時測距程式與計算定位座標程式以及傳送 UWB 定位座標至卡爾曼濾波器的程式，才可以完整運作 UWB 無人機實時定位。另外在使用時須先將 library 內的使用到的函式庫資料夾匯入 arduino 的 libraries 內，程式才可以成功編譯。

A. 主要程式

`realtime_pos_kalman_filter.ino`

用途說明:

讀取 MPU6050 加速度計資料，並在接收 UWB 定位結果後，執行卡爾曼濾波器運算。

B. library 中的函式庫

`positioning`

用途說明:

提供 LS、WLS、WLS-R 的定位近似解方法函式。

MPU6050

用途說明:

提供 MPU6050 加速度計與陀螺儀模組相關操作函式。

MatrixMath

用途說明:

提供相關矩陣運算函式。

五、 注意事項 – Warning

- 靜態定位以及動態定位需分別使用對應的程式。
- python 的靜態與動態定位需先有距離量測資料才可以進行定位。