Miracle dataset

The Miracle collaboration
Per domande su questa nota contattare: domenico.riccardi@pi.infn.it

12 agosto 2022

Sommario

Questa nota descrive brevemente la costruzione del dataset dell'esperimento Miracle a partire dai tre log file prodotti dall'environment software sviluppato per il volo.

1 Introduzione

Durante il volo del 01/07/2022 sono stati raccolti una grande quantità di dati con diversi rate di acquisizione. Indipendentemente, i vari sistemi hanno scritto lo stato dei detectors e sensori su diversi log file, richiedendo quindi di definire una procedura di merger per poter arrivare alla produzione di un dataset unificato. La diversità nel rate di acquisizione e scrittura ha portato anche a prendere una serie di decisioni per l'esecuzione di un matching ragionevole dei log file attraverso l'utilizzo del `timestamp` dell'*evento*[1].

Il dataset unificato è stato prodotto utilizzando Python/Pandas e successivemente salvato come ROOT file per poter eseguire l'analisi dati nell'ambiente ROOT.

2 I log files

Tre differenti log files sono stati prodotti durante il volo:

1. `20220701_data_xlr8.log`
2. `20220701_data_IMU.log`
3. `20220701_data_datalogger.log`

dai diversi sistemi (xlr8, IMU e datalogger), come indicato esplicitamente nel nome del file.

2.1 Dati dell'xlr8

L'xlr8 scrive una riga ogni secondo, raccogliendo le informazioni come mostrato in figura 2. Le diverse variabili sono definite come segue:

- `timestamp`. Identificativo univoco relativo all'istante temporale dell'acquisizione (in secondi).
- `reset_time`. Tempo trascorso a partire dello start/reset del sistema di acquisizione (in secondi).

[1] Definiamo *evento* la singola acquisizione dello stato dei detectors e sensori ad un dato tempo, corrispondente ad una singola linea all'interno del log file.

- `trigger`. AND di tutti gli ingressi della xlr8.
- `IN0, ..., IN7`. Conteggi di singola mattonella. Solo 3 ingressi della xlr8 erano collegati ad altrettante mattonelle, `IN1, IN2, IN3`, che corrispondono alle uniche colonne non nulle mostrate in figura 2. In particolare, facendo riferimento alla figura 1 che mostra la disposizione delle mattonelle, abbiamo:

  `mattonella 0 connessa a IN1`
  `mattonella 2 connessa a IN2`
  `mattonella 7 connessa a IN3`

- `AND0, AND1, AND2`. Conteggi dell'AND tra due mattonelle. Le connessioni esplicite sono:

  `AND0 = mattonella 0 + mattonella 2` (setup flusso orizzontale)
  `AND1 = mattonella 0 + mattonella 7` (setup flusso verticale)
  `AND3 = AND delle 3 mattonelle`

- `threshold1, threshold2`. Rispettivamente, rappresentano le tensioni di sogna in mV per i primi 4 ingressi (`IN0, ..., IN3`) e i restanti (`IN4, ..., IN7`). Al fine dell'analisi, solo la variabile `threshold1` deve essere considerata, essendo tutte le mattonelle connesse agli ingressi gestiti da questo trigger.
- `vbias`. Tensione di bias in V.

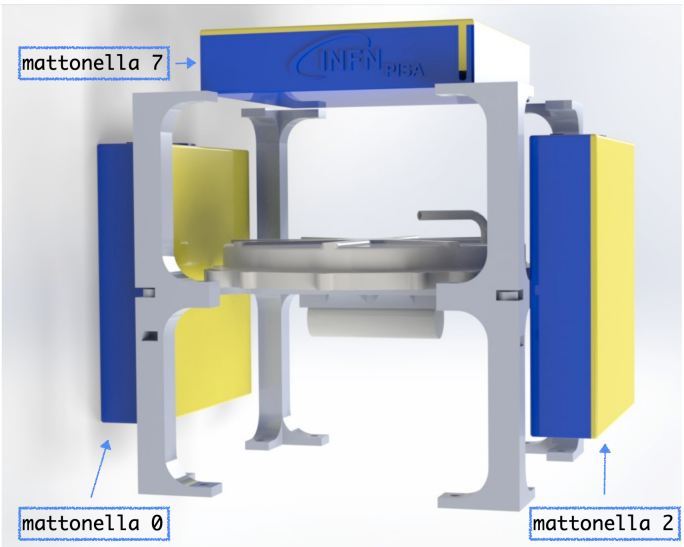

Figura 1: Disposizione delle mattonelle costituenti il telescopio. Il tubo a neutroni e la micromegas non erano presenti al momento del lancio.

Il timestamp è noto con 6 cifre decimali. A fine della costruzione del dataset unificato, questo viene approssimato ad una cifra decimale con la funzione `round` di Python[2], vedi fig.2.

2.2 Dati dell'IMU

L'IMU acquisisce ogni $\sim 0.02\,\mathrm{s}$ ed anche in questo caso si è optato per una approssimazione del timestap ad una sola cifra decimale. Ciò comporta che più righe del IMU dataset condividano lo

	timestamp	reset_time	trigger	IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7	AND0	AND1	AND2	threshold1	threshold2	vbias
0	55060.7	2	0	0	1	4	2	0	0	0	0	0	0	0	-151.0	-19.0	31.4
1	55061.8	3	0	0	2	4	6	0	0	0	0	0	1	0	-146.0	-14.0	31.4
2	55062.7	4	0	0	3	6	9	0	0	0	0	0	1	0	-156.0	-19.0	31.2
3	55063.8	5	0	0	4	6	10	0	0	0	0	0	1	0	-141.0	-29.0	31.5
4	55064.7	6	0	0	4	6	12	0	0	0	0	0	1	0	-141.0	-29.0	31.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13098	68191.5	12800	0	0	763282	744089	747298	0	0	0	0	29690	63027	10677	-156.0	-24.0	31.8
13099	68192.5	12801	0	0	763284	744089	747301	0	0	0	0	29690	63027	10677	-146.0	-24.0	31.6
13100	68193.5	12802	0	0	763287	744089	747303	0	0	0	0	29690	63027	10677	-151.0	-19.0	31.6
13101	68194.5	12803	0	0	763287	744089	747306	0	0	0	0	29690	63027	10677	-146.0	-14.0	31.8
13102	68195.5	12804	0	0	763287	744090	747306	0	0	0	0	29690	63027	10677	-166.0	-24.0	31.8

Figura 2: Alcune righe provenineti dal log file `20220701_data_xlr8.log`. Come visibile dalla colonna `timestamp`, la scrittura avviene circa ogni secondo.

	timestamp	heading	tiltCompensatedHeading	kalmanX	kalmanY
0	55060.0	50.45	50.36	-0.00	-0.13
1	55060.1	50.51	50.63	0.08	-0.11
2	55060.1	50.59	50.72	0.11	-0.09
3	55060.1	50.63	50.77	0.14	-0.08
4	55060.2	50.69	50.83	0.17	-0.10
...	...	...	...	...	...
299589	68296.5	357.92	358.58	0.20	-0.03
299590	68296.6	358.09	358.75	0.20	-0.03
299591	68296.6	358.09	358.70	0.20	-0.04
299592	68296.7	358.11	358.72	0.20	-0.03
299593	68296.7	358.41	359.07	0.20	-0.02

Figura 3: Alcune righe del `20220701_data_IMU.log`. Il `timestamp` viene arrotondato ad una cifra decimale e questa approssimazione provoca che a più righe corrisponda lo stesso timestamp.

stesso timestamp, come visibile per diversi eventi in figura 3. Volendo fare un esempio esplicito, consideriamo le righe 1,2,3 del dataset in figura 3. I timestamp prima dell'arrotondamento sono:

```
1 -> 55060.05240
2 -> 55060.09142
3 -> 55060.13020
```

e questo produce 3 righe del dataset che alla fine sono identificate dallo stesso timestamp, ovvero `55060.1`. Le altre colonne sono definite come:

- `heading`
- `tiltCompensatedHeading`
- `kalmanX`
- `kalmanY`

2.3 Dati dal datalogger

Il datalogger scrive ogni $\sim 2\,\mathrm{s}$ e il log file relativo, contiene ben 18 diverse informazioni/variabili. Per il momento, si sono considerate solo le seguenti 9 variabili[3]:

[2]Sintassi della funzione utilizzata: `round(number, digits)`.

[3]Una possibile estensione del set di variabili accessibili nel dataset finale, può essere richeista.

- `timestamp`. Usuale definzione (in secondi).
- `sats_in_use`. Corrisponde al numero di satelliti *visti* dal datalogger.
- `speed_over_ground`. Velocità espressa in km/h.
- `altitude`. Altitudine misura in metri.
- `board_temp`. Temperatura letta da un sensore sul datalogger (°C).
- `external_temp` Temperatura letta da un sensore nella regione interna della scatola (°C).
- `humidity, pressure` Umidità e pressione, rispettivamente in % e hPa.
- `battery_voltage` Tensione di alimentazione fornita dal raspberry (V).

Un esempio del datalogger dataset è in figura 4. In questo caso, si è scelto di approssimare il timestamp con zero cifre decimali, come mostrato nella figura precedentemente indicata.

	timestamp	sats_in_use	speed_over_ground	altitude	board_temp	external_temp	humidity	pressure	battery_voltage
0	55060.0	6	2.3	159.8	32.500	30.500	37.491	999.150	4.9
1	55062.0	7	2.1	157.0	32.500	30.625	37.384	999.990	4.9
2	55064.0	8	0.3	155.2	32.500	30.500	37.338	999.169	4.9
3	55066.0	8	0.1	153.9	32.500	30.375	37.247	999.239	5.0
4	55068.0	8	0.5	153.8	32.500	30.500	37.710	999.200	5.0
...	...	...	...	...	...	...	...	...	...
6460	68187.0	9	0.2	226.7	30.750	23.375	66.422	990.830	4.9
6461	68189.0	9	0.4	226.9	30.500	23.500	66.498	990.739	4.9
6462	68191.0	9	0.1	227.5	30.625	23.250	66.383	990.919	4.9
6463	68193.0	9	0.3	227.6	30.875	23.250	66.307	990.809	4.9
6464	68195.0	9	0.1	227.8	30.750	23.500	66.391	990.760	4.9

Figura 4: Alcune righe provenineti dal log file `20220701_data_datalogger.log`.

3 Fusione dei dataset

Una prima fusione avviene tra xlr8 dataset e l'IMU dataset attraverso il mathcing del timestamp, 5. Per effetto delle righe con lo stesso timestamp dell'IMU, anche questo dataset ha righe multiple con valori ridondati per le variabili dall'xrl8 e diversi per le variabili dall'IMU. Quindi, per poter *eliminare* questa ridondanza, si è eseguito un raggruppamento delle righe per `timestamp` assegnando alle variabili dell'IMU la media dei valori relativi ad ogni singola linea raggruppata[4]. Per esempio, si considerino le righe 0,1,2 del dataset in figura 5, che hanno lo stesso timestamp. I valori delle variabili provenienti dall'IMU dataset delle 3 righe vengono mediati:

	heading	tiltCompensatedHeading	kalmanX	kalmanY
55060.7	50.44	50.42	0.05	-0.08
	50.44	50.41	0.06	-0.09
	50.44	50.40	0.05	-0.10
mean	50.44	50.41	0.0533	-0.09

e viene costruito così un dataset con righe uniche, figura 6.

L'ultimo dataset così ottenuto viene ulteriormente modificato, arrotondando il timestamp a zero cifre decimali. Questo è l'ultimo step prima di fonderlo assieme al datalogger dataset per

[4]La media è stata calcolata con il metodo `mean()` di Python e il raggruppamento eseguito con `groupby` della libreria Pandas.

	timestamp	reset_time	trigger	IN0	IN1	IN2	IN3	IN4	IN5	IN6	...	AND0	AND1	AND2	threshold1	threshold2	vbias	heading	tiltCompensatedHeading	kalmanX	kalmanY
0	55060.7	2	0	0	1	4	2	0	0	0	...	0	0	0	-151.0	-19.0	31.4	50.44	50.42	0.05	-0.08
1	55060.7	2	0	0	1	4	2	0	0	0	...	0	0	0	-151.0	-19.0	31.4	50.44	50.41	0.06	-0.09
2	55060.7	2	0	0	1	4	2	0	0	0	...	0	0	0	-151.0	-19.0	31.4	50.44	50.40	0.05	-0.10
3	55061.8	3	0	0	2	4	6	0	0	0	...	0	1	0	-146.0	-14.0	31.4	50.66	50.56	0.05	-0.15
4	55061.8	3	0	0	2	4	6	0	0	0	...	0	1	0	-146.0	-14.0	31.4	50.68	50.60	0.05	-0.16
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
29658	68193.5	12802	0	0	763287	744089	747303	0	0	0	...	29690	63027	10677	-151.0	-19.0	31.6	325.74	330.10	-16.02	4.57
29659	68194.5	12803	0	0	763287	744089	747306	0	0	0	...	29690	63027	10677	-146.0	-14.0	31.8	325.83	330.14	-16.03	4.55
29660	68194.5	12803	0	0	763287	744089	747306	0	0	0	...	29690	63027	10677	-146.0	-14.0	31.8	325.76	330.07	-16.03	4.55
29661	68195.5	12804	0	0	763287	744090	747306	0	0	0	...	29690	63027	10677	-166.0	-24.0	31.8	325.88	330.26	-16.02	4.49
29662	68195.5	12804	0	0	763287	744090	747306	0	0	0	...	29690	63027	10677	-166.0	-24.0	31.8	325.89	330.28	-16.03	4.50

Figura 5: Alcune righe del dataset ottenuto dalla fusione del xlr8 dataset e dell'IMU dataset. Notare le righe con lo stesso timestamp per effetto di come viene costruito l'IMU dataset.

	timestamp	reset_time	trigger	IN0	IN1	IN2	IN3	IN4	IN5	IN6	...	AND0	AND1	AND2	threshold1	threshold2	vbias	heading	tiltCompensatedHeading	kalmanX	kalmanY
0	55060.7	2	0	0	1	4	2	0	0	0	...	0	0	0	-151.0	-19.0	31.4	50.440000	50.410000	0.053333	-0.090000
1	55061.8	3	0	0	2	4	6	0	0	0	...	0	1	0	-146.0	-14.0	31.4	50.706667	50.620000	0.050000	-0.153333
2	55062.7	4	0	0	3	6	9	0	0	0	...	0	1	0	-156.0	-19.0	31.2	50.605000	51.305000	0.830000	-0.130000
3	55063.8	5	0	0	4	6	10	0	0	0	...	0	1	0	-141.0	-29.0	31.5	50.435000	50.460000	0.145000	-0.070000
4	55064.7	6	0	0	4	6	12	0	0	0	...	0	1	0	-141.0	-29.0	31.5	50.373333	50.313333	0.003333	-0.100000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
13097	68191.5	12800	0	0	763282	744089	747298	0	0	0	...	29690	63027	10677	-156.0	-24.0	31.8	325.875000	330.275000	-16.050000	4.520000
13098	68192.5	12801	0	0	763284	744089	747301	0	0	0	...	29690	63027	10677	-146.0	-24.0	31.6	325.680000	329.925000	-16.020000	4.560000
13099	68193.5	12802	0	0	763287	744089	747303	0	0	0	...	29690	63027	10677	-151.0	-19.0	31.6	325.720000	330.103333	-16.013333	4.570000
13100	68194.5	12803	0	0	763287	744089	747306	0	0	0	...	29690	63027	10677	-146.0	-14.0	31.8	325.795000	330.105000	-16.030000	4.550000
13101	68195.5	12804	0	0	763287	744090	747306	0	0	0	...	29690	63027	10677	-166.0	-24.0	31.8	325.885000	330.270000	-16.025000	4.495000

Figura 6: Alcune righe del dataset finale ottenuto unendo l'xlr8 dataset e l'IMU dataset.

ottenere il dataset finale. Il dataset finale, lo chiameremo **miracle dataset**, ha quindi *eventi* caratterizzati da timestamp che differiscono di 2 secondi e con tutte e 29 variabili di ciascuno dei singoli datset, figura 7.

4 Da Pandas a ROOT

L'ultimo step è la produzione del root file a partire dal dataset pandas appena descritto. Per fare questo si è utilizzato il metodo `recreate` del pacchetto Uproot

```
miracle_data = uproot.recreate("miracle_data.root")
miracle_data["events"] = miracle_dataset
```

Il root file così creato, contiene al suo interno un TTree dal nome `events` con tutte le variabili del miracle dataset.

	timestamp	reset_time	trigger	IN0	IN1	IN2	IN3	IN4	IN5	IN6	...	kalmanX	kalmanY	sats_in_use	speed_over_ground	altitude	board_temp	external_temp	humidity
0	55062.0	3	0	0	2	4	6	0	0	0	...	0.050000	-0.153333	7	2.1	157.0	32.500	30.625	37.384
1	55064.0	5	0	0	4	6	10	0	0	0	...	0.145000	-0.070000	8	0.3	155.2	32.500	30.500	37.338
2	55066.0	7	0	0	6	8	14	0	0	0	...	0.050000	-0.080000	8	0.1	153.9	32.500	30.375	37.247
3	55068.0	9	0	0	9	12	19	0	0	0	...	-0.130000	-0.090000	8	0.5	153.8	32.500	30.500	37.710
4	55070.0	11	0	0	11	14	21	0	0	0	...	0.015000	-0.120000	8	0.4	153.4	32.500	30.625	37.790
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
6444	68156.0	12766	0	0	763258	744061	747235	0	0	0	...	-16.040000	4.555000	9	0.1	227.5	30.250	23.375	66.361
6445	68158.0	12767	0	0	763258	744061	747236	0	0	0	...	-16.045000	4.585000	8	0.0	227.1	30.375	23.250	66.376
6446	68158.0	12767	0	0	763258	744062	747239	0	0	0	...	-16.056667	4.533333	8	0.0	227.1	30.375	23.250	66.376
6447	68160.0	12768	0	0	763259	744062	747242	0	0	0	...	-16.000000	4.566667	9	0.7	227.3	30.500	23.250	66.475
6448	68160.0	12769	0	0	763260	744063	747244	0	0	0	...	-16.030000	4.575000	9	0.7	227.3	30.500	23.250	66.475

Figura 7: Una porzione del miracle dataset ottenuto dalla fusione delle informazioni contenute in tutti e 3 i log files.

5 Osservazioni ulteriori

Poteva essere evitata la fusione intermedia di xlr8 con IMU arrotondando il timestamp direttamente a zero cifre decimali? Si, questo poteva essere fatto, ma introducendo delle approssimazioni maggiori. Infatti, approssimare in questo modo i timestamp dell'IMU avrebbe significato mediare i valori delle relative variabili su almeno 20 righe del dataset (invece che 3). Con la prima approssimazione ad una cifra, invece, ci garantiamo un matching dei due dataset calcolando i valori medi in un intorno molto piccolo del timestamp dettato dall'xrl8.

6 Conclusioni

In questa nota si è discussa la procedura che ha portato alla relizzazione del miracle dataset. Il root file finale prodotto `miracle_data.root`, rappresenta il dataset principale su cui si svilupperà la successiva analisi dati.

Esso contiene un totale di 6449 eventi, ciascuno dei quali possiede 29 variabili (in figura 8 l'elenco con il relativo tipo).

```
name                 | typename
---------------------+------------------------
index                | int64_t
timestamp            | double
reset_time           | int64_t
trigger              | int64_t
IN0                  | int64_t
IN1                  | int64_t
IN2                  | int64_t
IN3                  | int64_t
IN4                  | int64_t
IN5                  | int64_t
IN6                  | int64_t
IN7                  | int64_t
AND0                 | int64_t
AND1                 | int64_t
AND2                 | int64_t
threshold1           | double
threshold2           | double
vbias                | double
heading              | double
tiltCompensatedHe... | double
kalmanX              | double
kalmanY              | double
sats_in_use          | int64_t
speed_over_ground    | double
altitude             | double
board_temp           | double
external_temp        | double
humidity             | double
pressure             | double
battery_voltage      | double
```

Figura 8: Elenco delle variabili del TTree `events` con il relativo tipo.