Users Manual

TRAGALDABAS documentation

Hardware & Software

Instituto Galego de Altas Enerxías

August 5, 2020

IGFAE
Instituto Galego de Física de Altas Enerxías

USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

XUNTA DE GALICIA

Contents

Chapter 1

Summary

Chapter 2

Introduction

2.1 The Cosmic Rays

At present, cosmic rays with large energies cannot be detected directly, so that we must measure the products of the atmospheric cascades of particles initiated by the incident astroparticle. In general, this cascades of secondary particles are generated by the inelastic nuclear collision between cosmic rays –those astroparticles– and the atmospheric particles. Those secondary particles continue interacting and generating other and other secondary particles until a maximum is reached, and then the shower atenuates as far as more and more particles fall below the threshold for further particle production.

2.2 The Detector

Since 2014, in the LabCAF laboratory of the Faculty of Physics of the USC, a TRASGO type detector has been installed and taking data: TRAGALDABAS (TRAsGo for the AnaLysis of the nuclear matter Decay, the Atmosphere, the earth B-field And the Solar activity), with the intention of making a joint analysis of the data taken simultaneously with TRISTAN (TRasgo para InveSTigaciones ANtárticas), separated by a distance around 1.3×10^7 m.

This TRAGALDABAS detector is made of four planes of avalanche RCPs, but at the moment only three of them are instrumented yet. Those planes of $1.2{\times}1.5$ m^2 are placed in a range of 1.8 m high and they are made up of 120 cells each one, placed in a $30{\times}40$ array. Therefore, this device has an active area of $1.2{\times}1.5$ m^2 and covers a vertical solid angle of ~ 5 sr offering a time resolution of ~ 300 ps and track arriving angle resolution better than $3°$.

2.3 The Data Flow

The detector is taking data with coincidence trigger between planes, at a rate about 7 million of registered events per day. This analog data of coincidences is converted to digital data and it is stored, along with humidity, pressure and temperature data.

For monitoring and alerting if data it is out of expected ranges, we use a software called Nagios. It is a software that provides great versatility to consult any parameter of interest in the system. The alerts generated are received by the corresponding managers (among other means) by email, when these parameters exceed the margins defined by the network administrator.

To format the numerical data and visualize it, we use Grafana. It is a platform without ani dependency and allows creating dashboards and graphs from multiple sources.

Both applications are multi-platform open-sources, licensed under the terms of the GNU General Public License and they are accessible from the computer called Trucha

2.3.1 A PC Called Trucha

It's name cames from the trout, that is a fish. In the LabCAF, the PCs (Pe-Ce-s in spanish, fishes in english) tower computers take names of fishes.

Actually, in this PC are stored the Nagios' warnings and alerts, and defined their ranges of activation. It looks like the directory tree of the Figure 2.3.1.

To keep the code clean, readable, and manageable, each of the scripts in `/etc/nagios/scripts/` whose name begins with `sensor` parses the data from a single detector plane. Scripts that their name start with `check` call the classes defined in the previous ones for each of the functional detector planes.

Scripts in `/etc/nagios/objects/` are the configurations of the variables used for calling the later mentioned python scripts and where the limits of the alerts for Nagios are defined.

```
/etc/nagios/
├── ...
├── ...
├── scripts/
│   ├── check_allAmbient.py
│   ├── check_allPower.py
│   ├── check_all.py
│   ├── sensorHighVoltage_tragaldabas.py
│   ├── sensorHumidity_tragaldabas.py
│   ├── sensorPressure_tragaldabas.py
│   └── sensorTemperature_tragaldabas.py
└── objects/
    ├── commands.cfg
    ├── contacts.cfg
    ├── hosts.cfg
    ├── printer.cfg
    ├── services.cfg
    └── checkcommands.cfg
```

Figure 2.1: Directory tree of Trucha, where the programs for flow control are stored.

Chapter 3

Kalman Filter

The Kalman filter method is intended for finding the optimum estimation $\mathbf{r}$ of the unknown vector $\mathbf{r}^t$, which describes the SAETA[1], according to the measurements $\mathbf{m}_k$, $k = 1...n$ of the vector $\mathbf{r}^t$.

The Kalman filter starts with a certain initial approximation $\mathbf{r} = \mathbf{r}_0$ and refines the vector $\mathbf{r}$, consecutively adding one measurement ater the ohter. The optimum value is attained after the addition of the last measurement.

Like it is seen in table 3.1, the upper plane T1 has first index and its height is 1.8 m, while lower plane T4 has the latest and it is at ground. So that, since we are starting Kalman filter by the lowest plane, $\mathbf{r}_0 \equiv \mathbf{r}_4$, and k indices go from $k = 4$ to $k = 1$

Name	Height / mm	Index
T1	1800	0
T2	900	1
T3	600	2
T4	0	3

Table 3.1: The four planes of TRAGALDABAS.

The vector $\mathbf{r}^t$ can change from one measurement to the next

$$\mathbf{r}^t = F_k \mathbf{r}^t_{k+1} + \boldsymbol{\nu}_k \tag{3.1}$$

where F_k is a linear operator, $\boldsymbol{\nu}_k$ is a process noise between $(k-1)$-th and k-th measurements

The measurement $\mathbf{m}_k$ linearly depends won $\mathbf{r}^t_k$:

$$\mathbf{m}_k = H_k \mathbf{r}^t_k + \boldsymbol{\eta}_k, \tag{3.2}$$

where $\boldsymbol{\eta}_k$ is an error of the k-th measurement.

It is assumed that measurement errors $\boldsymbol{\eta}_i$ and the process noise $\boldsymbol{\nu}_i$ are uncorrelated, inbiased ($\langle \boldsymbol{\eta}_i \rangle = \langle \boldsymbol{\nu}_i \rangle = \mathbf{0}$) and those covariance matrices V_k, Q_k are

[1]Te particle is defined completely by a set of parameters $\mathbf{r}_k = (x_k, x'_k, y_k, y'_k, t_k, 1/v)^T$. We call it **SAETA** (SmAllest sET of pArameters).

knownw:

$$\langle \boldsymbol{\eta}_i \cdot \boldsymbol{\eta}_i^T \rangle \equiv V_i,$$
$$\langle \boldsymbol{\nu}_j \cdot \boldsymbol{\nu}_j^T \rangle \equiv Q_j. \tag{3.3}$$

The Kalman filter starts with an initial hypothetical vector $\mathbf{r_0}$, then for each measurement $\mathbf{m}_k$ a vector $\mathbf{r}_k$ is calculated, wwhich is the optimum estimation of the vector $\mathbf{r}^t$ according to the first k measurements.

The conventional Kalman filter algorithm consists of four stages:

1. Initialization step: Choose an appropieate value of the vector $\mathbf{r}_0$. We use an hypothetical normal one

$$\mathbf{r}_0 = (x_0, 0, y_0, 0, t_0, sc)^T, \tag{3.4}$$

where x_0, y_0, t_0 are the coordinates of hit in the T4 plane, $sc = 1/c$ the slowness (inverse of light celerity), and $x' = y' = 0$ the projections on x and y axes respectively:

$$x' = \frac{cx}{cz} = \frac{\sin\theta\cos\phi}{\cos\theta},$$
$$y' = \frac{cy}{cz} = \frac{\sin\theta\sin\phi}{\cos\theta}, \tag{3.5}$$

in spherical coordinates.

Its covariance matrix is set to $C_0 = I \cdot \text{inf}^2$, where inf denotes a large positive number. We used:

$$C_0 = \begin{pmatrix} \text{SIGX}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \text{VSLP} & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{SIGY}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{VSLP} & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{SIGT}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{VSLN} \end{pmatrix}, \tag{3.6}$$

where VSLP $= 0.1^2$ and VSLN $= 0.01^2$ are the variances for slope and slowness respectively, and

$$\text{SIGX} = \frac{1}{\sqrt{12}}\text{WCX},$$
$$\text{SIGY} = \frac{1}{\sqrt{12}}\text{WCY},$$
$$\text{SIGT} = 300 \text{ ps}, \tag{3.7}$$

the variances for cell and time dimensions, with WCX $= 125$ mm and WCY $= 120$ mm the width of cells on x and y axes respectively.

2. Prediction step: Propagate the vector from $(k+1)$-th to k-th plane by propagation matrix

$$F_k = \begin{pmatrix} 1 & dz_k & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & dz_k & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & ks_k dz_k \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.8}$$

where dz_k is the distance between k-th and the plane $(k+1)$-th below

$$dz_k = z_k - z_{k+1}, \tag{3.9}$$

and ks_k is

$$ks_k = \sqrt{1 + x_k'^2 + y_k'^2}, \tag{3.10}$$

so that,

$$\begin{aligned} \tilde{\mathbf{r}}_k &= F_k \mathbf{r}_{k-1}, \\ \tilde{C}_k &= F_k C_{k-1} F_k^T. \end{aligned} \tag{3.11}$$

3. PROCESS NOISE: In contrast to the prediction step, describing deterministic changes of the vector $\mathbf{r}^t$ in time, the process noise describes probabilistic deviations of the vector $\mathbf{r}^t$.

$$\begin{aligned} \hat{\mathbf{r}}_k &= \tilde{\mathbf{r}}_k, \\ \hat{C}_k &= \tilde{C}_k + Q_k. \end{aligned} \tag{3.12}$$

4. FILTRATION STEP: At this step the state vector $\hat{\mathbf{r}}_k$ is updated with the new mweasurement $\mathbf{m}_k$ to get the optimal estimate of $\mathbf{r}_k$ and its covariance matrix C_k:

$$\begin{aligned} K_k &= \hat{C}_k H_k^T (V_k + H_k \hat{C}_k H_k^T)^{-1}, \\ \mathbf{r}_k &= \hat{\mathbf{r}}_k + K_k(\mathbf{m}_k - H_k \hat{\mathbf{r}}_k), \\ C_k &= \hat{C}_k - K_k H_k \hat{C}_k, \\ \chi_k^2 &= \chi_{k+1}^2 + (\mathbf{m}_k - H_k \hat{\mathbf{r}}_k)^T (V_k + H_k \hat{C}_k H_k^T)^{-1} (\mathbf{m}_k - H_k \hat{\mathbf{r}}_k). \end{aligned} \tag{3.13}$$

Here, the k-th measurement is

$$\mathbf{m}_k = (x_k, y_k, t_k)^T, \tag{3.14}$$

and its covariance matrix

$$V_k = \begin{pmatrix} \mathrm{SIGX}^2 & 0 & 0 \\ 0 & \mathrm{SIGY}^2 & 0 \\ 0 & 0 & \mathrm{SIGT}^2 \end{pmatrix}. \tag{3.15}$$

The matrix H_k of the model is simply the identity matrix

$$H_k = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{3.16}$$

which converts every $\hat{\mathbf{r}}_k = (x_k, x_k', y_k, y_k', t_k, sc)^T$ into $\mathbf{m_r}_k = (x_k, y_k, t_k)^T$ for comparing $\delta\mathbf{m}_k = \mathbf{m}_k - \mathbf{m_r}_k$ in (3.13).

The following designations are used in Eqs. (3.11)-(3.13): $\mathbf{r}_{k+1}$, C_{k+1} are the optimum estimation, obtained at the previous step and the error covariance matrix; the matrix F_k relates the state at step $k+1$ to the state at step k; [2] $\tilde{\mathbf{r}}_k$, $\tilde{C}_k$ are predicted estimation of $\mathbf{r}_k^t$ before the process noise; $\hat{\mathbf{r}}_k$, $\hat{C}_k$ are predicted estimation of $\mathbf{r}_k^t$ after the process noise; $\mathbf{m}_k$, V_k are the k-th measurement and its covariance matrix; the matrix H_k is the model of measurement; the matrix K_k is the so-called gain matrix; the value χ_k^2 is the total χ^2-deviation of the obtained estimation $\mathbf{r}_k$ from the measurements $\mathbf{m}_1, ...\mathbf{m}_k$.

The vector $\mathbf{r}_n$ obtained after the filtration of the last measurement is the desired optimal estimation of the $\mathbf{r}_n^t$ with the covariance matrix C_n.

In track fitting applications, the state vector $\mathbf{r}_k$ is vector of the track parameters, the prediction matrix F_k dewscribes extrapolation of the track in the magnetic field from one detector to another, and the matrix of noise Q_k describes the effect of multiple scattering in the material.

[2] Remember that we start from the lowest plane to de highest.