ANNEXES

Annexe A

Rappel du fonctionnement d'un photomultiplicateur.

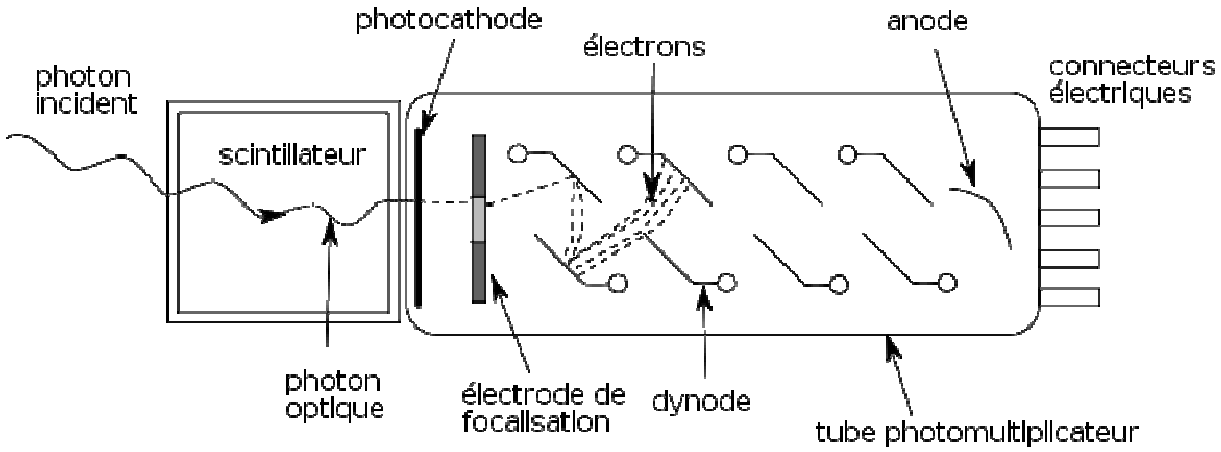

Fig. A – *Schéma du principe de fonctionnement d'un photomultiplicateur.*

Un photomultiplicateur est un dispositif permettant la détection de photons.

Lorsqu'un photon passe à travers le scintillateur, il va frapper la photocathode. Des électrons sont alors produits par effet photoélectrique.
L'électrode de focalisation permet de diriger les électrons créés vers le multiplicateur d'électrons.
Celui-ci est composé d'électrodes, également dénommées dynodes. Chacune de ces dynodes est maintenue à un potentiel plus élevé que le précédent.
Les électrons incidents sont accélérés par le champ électrique ainsi produit et vont frapper la première dynode. Ce choc va créer d'autres électrons, moins énergétiques mais plus nombreux. **C'est l'émission d'électrons secondaires.**
Tous ces électrons vont à nouveau être accélérés et vont frapper la seconde dynode, créant de nouvelles particules, et ainsi de suite jusqu'à atteindre l'anode. L'accumulation de ces électrons sur l'anode créé une brève impulsion de courant, marquant ainsi la détection d'un photon.

Les principales caractéristiques d'un photomultiplicateur sont gain, le temps de montée, le temps de transit, la dispersion du temps de transit (TTS) et le taux de comptage dans l'obscurité.
Le gain est le rapport entre le courant de l'anode et le courant photoélectrique.
Le temps de montée est le temps nécessaire pour que l'impulsion de l'anode augment de 10% à 90% de sa valeur maximale lorsque le tube est éclairé.
Le temps de transit est le temps entre le moment d'illumination et le moment de l'impulsion sur l'anode.

Le TTS est la variation du temps de transit d'une impulsion lumineuse à l'autre suivant le point d'émission sur la photocathode.
Le taux de comptage dans l'obscurité est le courant enregistré en l'absence de tout rayonnement. Il s'exprime en Hz.

Le tableau suivant donne un résumé des caractéristiques des photomultiplicateurs utilisés par ANTARES.

Gain	5.10^7
Haute tension	2000 V
Amplitude SPE	50 mV
Surface sensible	$\geq$ 500 cm²
Temps de montée	$\leq$ 5 ns
Largeur du signal	$\leq$ 12 ns
TTS	$\leq$ 3 ns
Taux de comptage dans l'obscurité	$\leq$ 10 kHz

SPE signifie photoélectron unique (« single photoelectron »).

Annexe B

Extrait du code C pour le calcul du Linear Prefit

Calcul effectué :

$$\theta = \frac{1}{\sum f_i \sum f_i c_i^2 t_i^2 - \left(\sum f_i c_i t\right)^2} \begin{bmatrix} \sum f_i \sum f_i x_i ct_i - \sum f_i ct_i \sum f_i x_i \\ \sum f_i \sum f_i y_i ct_i - \sum f_i ct_i \sum f_i y_i \\ \sum f_i \sum f_i z_i ct_i - \sum f_i ct_i \sum f_i z_i \end{bmatrix}$$

```
Float linearprefitdir(int datatothits, float datatemps[], float datax[])
{
        /* datatothits correspond au nombre total de hits de l'évènement
        Datatemps[] est un tableau contenant les temps de chaque hit
        Datax[] est un tableau contenant la coordonnée x de chaque hit. Il peut également
        contenir y et z, selon le paramètre que l'on envoit dans la fonction */

        int n;
        float f=0.1,d;
        float fraction=0.0, fraction1=0.0, fraction2=0.0, fraction3=0.0, fractionx1=0.0,
        fractionx2=0.0;

        //Calculs préliminaires
        for(n=0;n<datatothits;n++)
        {
                //Somme fi*ti^2*c^2
                fraction1=fraction1+f*datatemps[n]*datatemps[n]*0.3*0.3;
                //Somme fi*ti*c
                fraction2=fraction2+f*datatemps[n]*0.3;
                //Somme fi*xi
                fractionx1=fractionx1+f*datax[n];
                //Somme fi*ti*xi*c
                fractionx2=fractionx2+f*datatemps[n]*datax[n]*0.3;
        }
        //(Somme fi*ti*c) ^2
        fraction3=fraction2*fraction2;
        fraction=1.0/(f*datatothits*fraction1-fraction3);
        //Finalement :
        d=fraction*(f*datatothits*fractionx2-fraction2*fractionx1);

return d ;
}
```

Annexe C

Lecture des coordonnées des modules optiques

Cette partie présente un exemple du code utilisé pour lire un fichier de données.
La figure B rappelle la structure du fichier.

```
"APP_START","APP_STOP","LCM_ID","POSITION","X","Y","Z","HEADING","PITCH","ROLL"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1624","101","8221.6","2479.7","-2386.06568","3.57288","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1631","102","8221.6","2479.7","-2371.53663","2.78716","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1622","103","8221.6","2479.7","-2357.00357","3.59217","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1641","104","8221.6","2479.7","-2342.46652","1.84659","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1625","105","8221.6","2479.7","-2327.93846","3.28034","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1627","106","8221.6","2479.7","-2313.41041","2.15115","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1645","107","8221.6","2479.7","-2298.88535","6.88525","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1626","108","8221.6","2479.7","-2284.3563","4.70708","0","0"
"01/06/2007 01:00:00.000000000","01/06/2007 01:12:00.000000000","1643","109","8221.6","2479.7","-2269.83924","5.74915","0","0"
```

Fig. B – *Structure du fichier de données. De gauche à droite : date et heure de début, date et heure de fin, adresse IP de l'OM, numéro de l'étage, coordonnées absolues X, Y et Z, angles Heading, Pitch, Roll.*

Le fichier est nommé « alignement » dans les lignes de code.
Deux boucles sont utilisées pour traiter tout le fichier :

```
for(n=0 ;n<tothits ;n++)
{
        //la fonction rewind permet de positionner le curseur au début du fichier
        //alignement.
        g=1;rewind(alignement);
        while(g!=0)
        {
                //La fonction fgets lit une ligne entière du fichier et la stocke dans info, une
                //chaine de caractères de taille 500.
                fgets(info,500,alignement);

                //La fonction sscanf permet de stocker les informations de la chaine info
                //dans des variables. On peut donc extraire chaque nombre de la ligne.
                sscanf(info,"%d/%d/%d %d:%d:%d.%d %d/%d/%d %d:%d:%d.%d %d %d %f %f
                %f %f %f %f",&jour,&mois,&annee,&heures,&minutes,&secondes,&in1,
                &jour2,&mois2,&annee2,&heures2,&minutes2,&secondes2,&in2,&ip,&in3,
                &x,&y,&z,&heading,&pitch,&roll) ;

                //Code
                //…
        }
}
```