

# RAPPORT DU PROJET DE FIN D'ETUDE

Encadrant: Ould Boumama Belkacem

Élèves: Hamzaoui Oussama et Ziou Ismail

PFE 2014/2015

# Sommaire

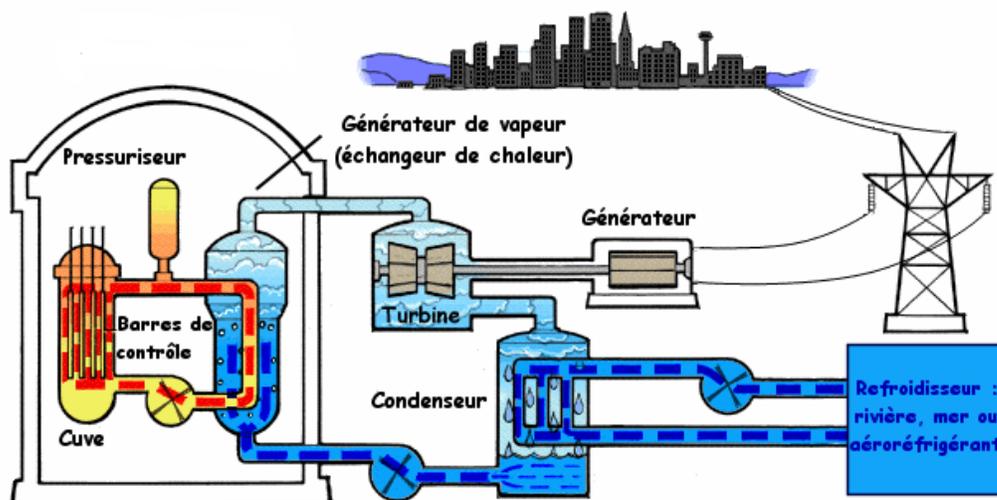
<b>PRESENTATION DU SYSTEME .....</b>	<b>2</b>
<b>SUPERVISION DU GENERATEUR DE VAPEUR.....</b>	<b>3</b>
<b>1) LA SUPERVISION .....</b>	<b>3</b>
<b>2) PROBLEMATIQUE .....</b>	<b>5</b>
<b>FILTRAGE DES SIGNAUX.....</b>	<b>6</b>
<b>1) FILTRE À MOYENNE GLISSANTE.....</b>	<b>6</b>
<b>2) ELIMINATION DES BRUITS RESTANTS.....</b>	<b>9</b>
<b>ISOLATION DES ERREURS .....</b>	<b>11</b>

## Présentation du système

Le générateur de vapeur couramment appelé GV est l'une des composantes principales des réseaux électriques thermiques et nucléaires. La fonction du générateur de vapeur est d'échanger la chaleur entre le circuit primaire chauffé par le réacteur (pétrole, charbon, réacteur nucléaire...) et le circuit secondaire qui fait tourner la turbine à vapeur - ou bien transporte la chaleur produite dans le cas d'un réseau de chaleur. Son principe de fonctionnement se base sur le stockage d'un certain volume d'eau à température ambiante dans des réservoirs, cette eau sera pompée par la suite et portée à ébullition à très haute température au sein d'une chaudière. Grâce aux générateurs de vapeur, l'eau du circuit secondaire est portée à ébullition, à un certain niveau de pression. La vapeur s'échappe alors sous pression et fait tourner le groupe turbo-alternateur, une turbine couplée à un alternateur, situé dans la salle des machines.

Le système est composé essentiellement donc de :

- Un réservoir
- Des pompes
- Une chaudière
- Un groupe turbo-alternateur
- Un condensateur



# Supervision du générateur de vapeur

## 1) La supervision

La supervision est un procédé industriel qui permet une surveillance automatique d'un système industriel. En informatique industrielle, la supervision des procédés peut être une application de surveillance, de contrôle-commande évolué, de diagnostic, de suivi et de pilotage informatique de procédés de fabrication automatisés. Celle-ci concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus pour ensuite en déduire après traitement le bon fonctionnement du système ou pas, en déduire l'évolution du système, détecter des défauts de capteurs ...

La supervision se base sur un modèle pré-établi du système. Un modèle étant un ensemble d'équations modélisant le système, et qui sont vérifiées dans le cas d'un bon fonctionnement de ce dernier. De là vient le principe même de la supervision, car trouver une incohérence au niveau d'une équation signifie que la partie du système régie par cette équation est défaillante.

Pour comprendre mieux ce principe, on propose d'étudier le modèle du réservoir d'eau. Ce dernier est régi par une équation différentielle liant le débit d'entrée au débit de sortie et à la hauteur d'eau dans le réservoir.

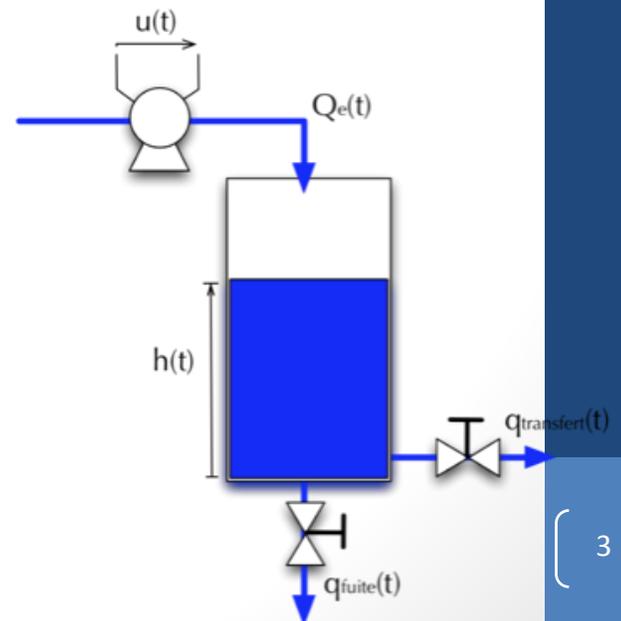
Dans un premier temps, on imaginera que la vanne permettant de simuler une fuite au niveau du réservoir est fermée, et donc le débit qui lui est associé est naturellement nul.

L'équation régissant ce système est :

$$S. \frac{dh(t)}{dt} = Q_e(t) - Q_{transfert}(t)$$

Dans un fonctionnement normal (sans fuite) l'équation peut être donc écrite sous la forme :

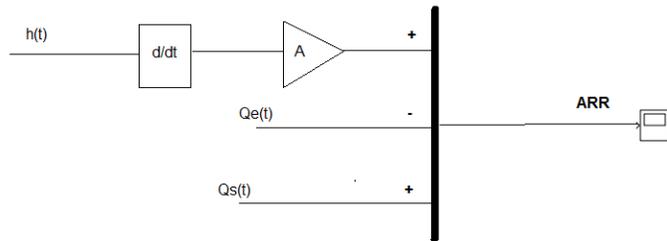
$$S. \frac{dh(t)}{dt} - Q_e(t) + Q_{transfert}(t) = 0$$



Pour superviser le réservoir on reconstituera à l'aide d'un logiciel approprié comme Matlab/Simulink cette équation à partir des données recueillis par les capteurs, qui sont dans notre cas :

- Capteur de niveau d'eau dans le réservoir
- Capteur de débit d'entrée
- Capteur de débit de sortie

En reconstituant l'équation sous un logiciel, on crée ce qu'on appelle un résidu. Ce résidu qu'on va nommer ARR va donc avoir comme équation :



$$S \cdot \frac{dh(t)}{dt} - Q_e(t) + Q_{transfert}(t) = ARR$$

Sous un fonctionnement normal du réservoir ce résidu vaut donc zéro, et une surveillance de ce résidu nous permet donc de détecter un mauvais fonctionnement du système ou la défaillance d'un capteur si jamais ce résidu prend une valeur autre que zéro.

Le but étant d'avoir un suivi continu de l'évolution du système, les valeurs relevées par les capteurs doivent être traitées et utilisé tout au long du fonctionnement du système afin de garder l'aspect temps réelle de la supervision. Cela nous permet de détecter immédiatement toutes failles ou erreurs qui peuvent survenir tout au long du fonctionnement du générateur de vapeur.

Le système mise en place dispose pour cela d'un logiciel DSPACE, permettant le relevé des différentes valeurs des capteurs composant le générateur pour ainsi pouvoir traité celle-ci au niveau d'un programme Matlab délivrant ainsi les valeurs des alarmes.

## 2) Problématique

Les signaux issues des capteurs sont sujet à des « bruitations » pouvant venir de l'environnement extérieur (champ magnétique, champ électrique ...) , ou d'une fluctuation des valeurs mesurées, mais aussi et principalement de la dérivation numérique de ces signaux. Car une dérivation exacte n'étant pas possible numériquement, les logiciels tel que Matlab utilisent des approximations, qui des fois rajoutent du bruit sur notre signal d'origine.

L'objectif donc de notre projet est de filtrer ces signaux pour qu'ils soient plus exploitables, car des signaux noyés dans le bruit génèrent de fausses alarmes, ce qui à l'échelle industrielle peut être très nuisant car ceci fait intervenir les équipes de maintenance et peut interrompre la production pour une alarme qui est faussement déclenchée à cause des parasites présents dans les signaux.

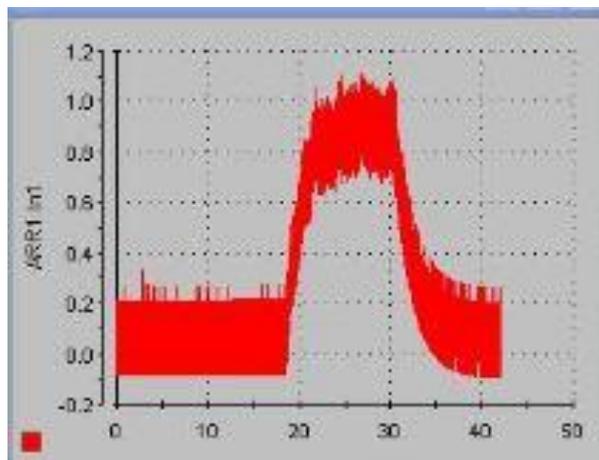
Les ARRs cité précédemment ou plus précisément les équations modélisant les différents systèmes composant le générateur de vapeur, permettent la détection d'une erreur ou une autre dans un des systèmes sans donner de plus ample précision. Etant donné le fait que chaque équation comporte un certain nombre de variable et que chacune de ces composantes peut être sujet à une erreur, l'isolation de l'erreur se définit comme étant une nécessité dans l'évolution de notre travail.

En d'autre mot, prenant l'exemple cité précédemment, le réservoir d'eau peut avoir différente causes d'enclenchement de l'ARR ou de la déstabilisation de l'équation. La présence de fuite peut être le déclencheur majeur d'une erreur mais une anomalie au niveau des capteurs de débit peut être aussi une raison.

Le dernier chapitre de notre travail se résume en une analyse des nombreuses équations régissant le fonctionnement de notre système, afin de trouver une analogie entre équation et l'isolation des erreurs.

## Filtrage des signaux

Dans cette partie on détaillera les étapes suivies pour filtrer le signal, qui était initialement noyé dans le bruit et tout à fait inexploitable.



(Signal initial non filtré)

### 1) Filtre à moyenne glissante

Après avoir recherché différentes méthodes de filtrages utilisés en industrie, la méthode apparaissant comme la plus efficaces et la plus répandues s'est avéré être la moyenne glissante. On a donc opté pour utiliser cette méthode.

La moyenne glissante est donc tout d'abord comme son nom l'indique une moyenne, qui est calculée dans une fenêtre temporelle avec un nombre d'échantillons qui est paramétrable. Elle est dite glissante, car elle est recalculée de façon continue en utilisant pour chaque calcul un sous-ensemble d'éléments où chaque nouvel élément remplace le plus ancien créant ainsi une moyenne glissante temporellement.

Cette explication reste néanmoins empirique, on démontrera dans ce qui suit aux travers de la théorie du traitement de signal, que la moyenne glissante est bien un filtre qui peut servir à éliminer les parasites.

Notre signal étant inconnu à l'avance, on accordera donc à chaque nouvelle valeur envoyée par le capteur la même importance que les autres. Ce qui traduit mathématiquement par un même coefficient pour toutes les valeurs.

On considérera que notre filtre est de taille N, ce qui induit donc l'expression suivante de la moyenne :

$$y_k = \frac{1}{N} \cdot \sum_{i=0}^{N-1} x_{k-i}$$

D'un point de vue de la théorie du traitement de signal, cette expression représente la somme de N échantillons divisée par N.

Ce qui nous ramène donc à l'écriture de la fonction de transfert de ce filtre qui sera sous la forme :

$$H(z) = \frac{1 + z^{-1} + z^{-2} + \dots + z^{-(N-1)}}{N}$$

Dans le domaine fréquentiel, cette fonction de transfert s'écrit sous la forme :

$$H(f) = \sum_{i=0}^{N-1} \frac{1}{N} \cdot e^{-2\pi j f i T}$$

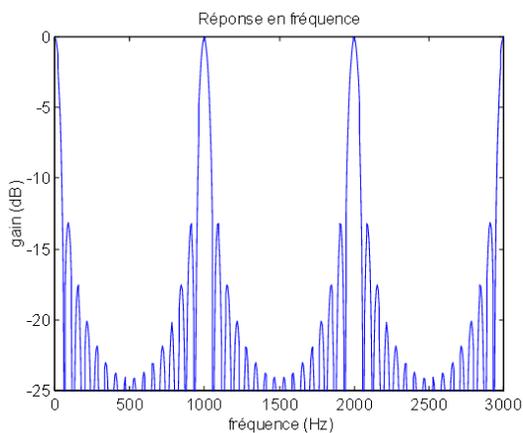
Cette fonction de transfert peut être écrite tout simplement comme ceci :

$$H(f) = \frac{1}{N} \cdot \frac{\sin(2\pi f \cdot \frac{NT}{2})}{\sin(2\pi f \cdot \frac{T}{2})} \cdot e^{-2\pi j f \frac{N-1}{2} \cdot T}$$

En module on a donc :

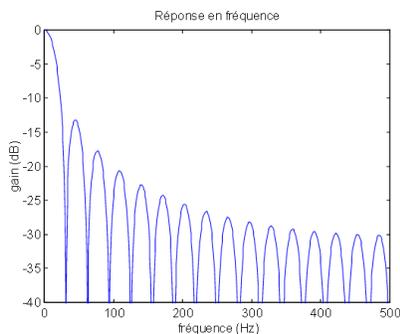
$$A(f) = \left| \frac{\sin(2\pi f \cdot \frac{n \cdot T_e}{2})}{n \cdot \sin(2\pi f \cdot \frac{T_e}{2})} \right|$$

Ce qui nous donne pour un nombre d'échantillons  $N=16$  la réponse fréquentielle suivante :

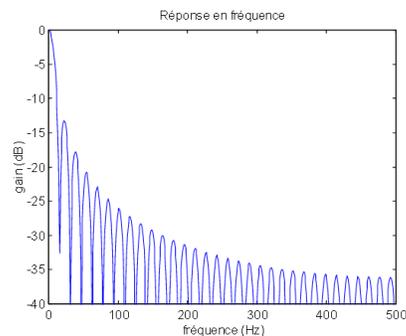


On remarque bien que malgré l'atténuation de bruits de basses fréquences, il existe des fréquences avec un gain de 0dB et qui ne sont donc pas réduites.

Ce problème peut être résolu en augmentant le nombre d'échantillons traités par opération. Les figures ci-dessous montrent l'évolution de la réponse du filtre avec l'évolution du nombre d'échantillons pris.



**N= 32**

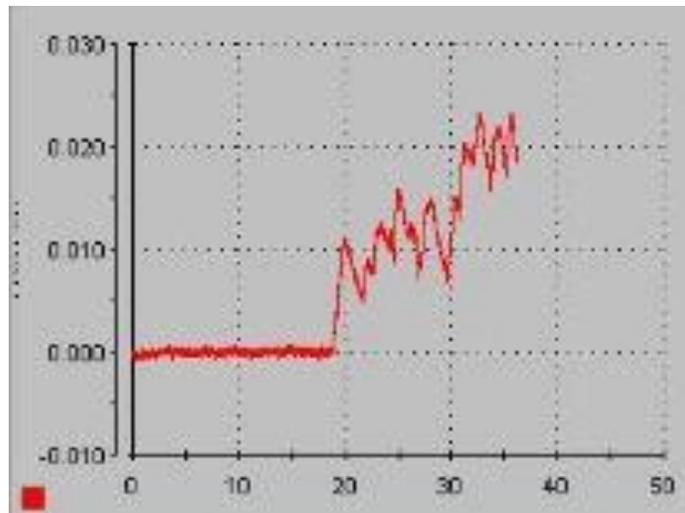


**N=64**

Malheureusement prendre un nombre important d'éléments pour calculer la moyenne prends un temps de calcul énorme, et donc pour garder la caractère temps-réel du système on se contentera de prendre  $N=16$ .

## 2) Elimination des bruits restants

Comme on l'a cité dans la partie précédente, la moyenne glissante malgré son efficacité laisse passer des bruits à certaines fréquences. Pour remédier à ça, il a fallu observer la nature du signal à la sortie du filtre moyenneur, on observe un filtre à l'allure suivante :



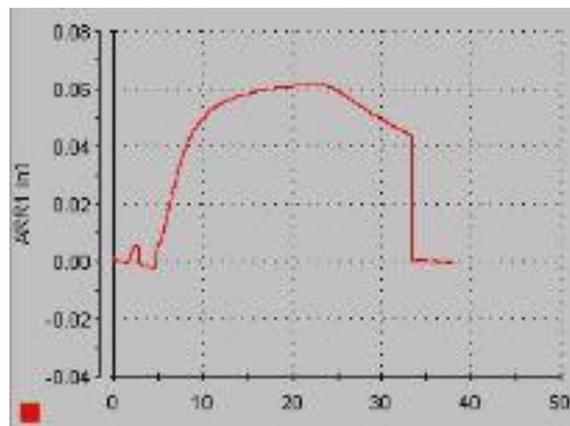
On remarque donc qu'il y a une nette amélioration au niveau de l'élimination des bruits, malgré la présence prédictible d'un bruit certes minime mais nuisant à des fréquences fixes.

La première méthode utilisée pour traiter ce bruit a été de calculer la fréquence d'apparition de bruit et d'appliquer un filtre passe-bas qui permettra de filtrer ce bruit. Mais cette solution n'est d'une part pas très efficace car un simple changement sur le signal, changera ces fréquences non traitées par le filtre moyenneur et rendra donc ce filtrage inutile.

Il a fallu donc penser à une solution plus adaptative, et vu la nature du signal. Nous avons pensé à calculer la moyenne du signal en chaque front montant de ce dernier, mais pour cela le problème de définir une période de calcul de la moyenne s'est posé car comme pour le filtre passe-bas le signal peut changer et donc décaler le calcul de la moyenne des fronts montants du signal.

On a eu donc l'idée d'utiliser le signal lui-même comme référence pour le calcul de la moyenne. Le principe sera donc de calculer la moyenne du signal entre chaque front montant de ce dernier, ce qui éliminera les fluctuations et nous permettra de lisser le signal.

L'application de ce principe nous permet de trouver un résidu bien filtré, dont on peut dégager une information concernant les défaillances du système, les alarmes ...



(Résidu filtré)

## Isolation des erreurs

Les méthodes basées sur l'identification de paramètres visent à estimer la cohérence entre les paramètres du modèle de bon fonctionnement du système et les paramètres du modèle du système identifié en temps réel. Elles travaillent dans l'espace de structure et sont donc plus pertinentes d'une défaillance liée à une modification structurelle du système. Elles facilitent ainsi l'opération de diagnostic dans le cas où les paramètres du modèle présentent une signification physique immédiate vis-à-vis du processus.

Les écarts ou les résidus constituent une signature, un identificateur d'une défaillance. Le processus de diagnostic consiste à comparer les relevées sur le processus physique à une bibliothèque de signatures théoriques, représentatives de défaillances identifiées afin de localiser l'origine de la défaillance et de l'identifier finement. Il est donc nécessaire de disposer également d'un ou plusieurs modèles de défauts.

Il est possible de procéder, à l'image du raisonnement humain en diagnostic, en faisant un parallèle entre l'expérience des défaillances déjà rencontrées, leurs effets et les résidus relevés dans la situation présente. Pour cette stratégie on parle de Matching. On obtient alors à partir de cette opération une matrice ( Fault Signature Matrix).

Var	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24
P16	1	1						1	1	1								1	1	1	1			
L1	1	1	1						1	1	1													
F3	1	1	1	1	1																			
m05	1	1							1	1														1
T2		1			1					1	1													
Mo2			1																					
O4				1		1																		
Wd					1	1																		
P7				1	1	1	1	1					1				1							
P12								1	1	1	1						1							
P10					1	1		1	1															
T20			1					1	1															
F23										1	1													
T21										1	1													
T22										1	1													
L8					1	1																		1
L18	1	1								1	1				1						1	1	1	
T5															1									
T6															1	1								
P14																1	1							
P15																	1	1						
P15																		1	1					
P11																			1	1				
T17																				1	1			
L19																					1	1	1	
L9																						1	1	
m01																								1

La mise en place d'une telle matrice nous permet de parler de notion de détectabilité et isolabilité. Il est donc possible de relever les erreurs décelable mais aussi celles qui sont isolable.

## Détectabilité

*Une erreur  $E_j$  est détectable si il existe au moins un des scalaires de son vecteur est différent de zero*

## Isolabilité

*Une erreur  $E_j$  est isolable si celle-ci a un vecteur différent des autres erreurs en plus du fait qu'elle soit détectable*

La compréhension du travail effectué peut être assez complexe à comprendre en vue de la taille de la matrice des erreurs du système en entier. Il est plus facile d'appréhender le sens de la méthode utilisé sur un système moins complexe.

Le réservoir d'eau est un système ayant une matrice de défaut dut au fait qu'il ne soit régit que par deux équations et donc de deux ARR.

$$P - \frac{1}{A/g} \cdot \int \dot{m} dt = ARR1$$

$$T - \frac{1}{m \cdot C_v} \int \dot{H} dt = ARR2$$

$\dot{m}$  : Débit en sortie du réservoir

P : Pression à l'intérieur du réservoir

T : Température de l'eau

H : enthalpie

A : Section de la cuve

$C_v$  : Chaleur spécifique a volume constant

g : Constante de gravité

Erreur	Arr1	Arr2
Leak (m)	1	1
Capteur de pression	1	0
Capteur de température	0	1

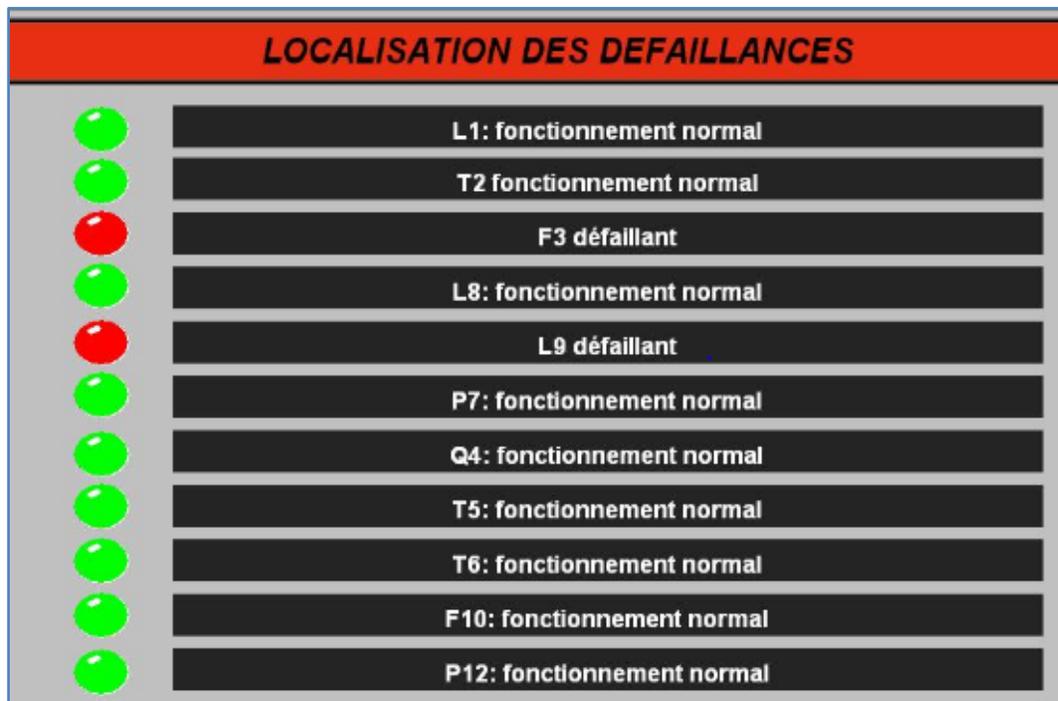
On place un '1' si la variable fait partie des valeurs de l'équation Arr et un '0' dans le cas contraire. Ceci nous permet de déterminer les erreurs détectable et ceux qui sont isolable afin de mettre en place une stratégie de supervision adéquate aux données dont on dispose. Dans le cas présent l'ensemble des erreurs possible dans notre système sont détectable, étant donné que la définition citée précédemment s'applique à toutes les erreurs. Celles-ci sont aussi isolables du fait que chaque vecteur diffère des autres.

Il est donc possible à l'aide de ce tableau de mettre en place un programme qui à travers la matrice peut déduire ou isoler l'erreur. Dans le cas du réservoir le script peut s'écrire de la manier suivante :

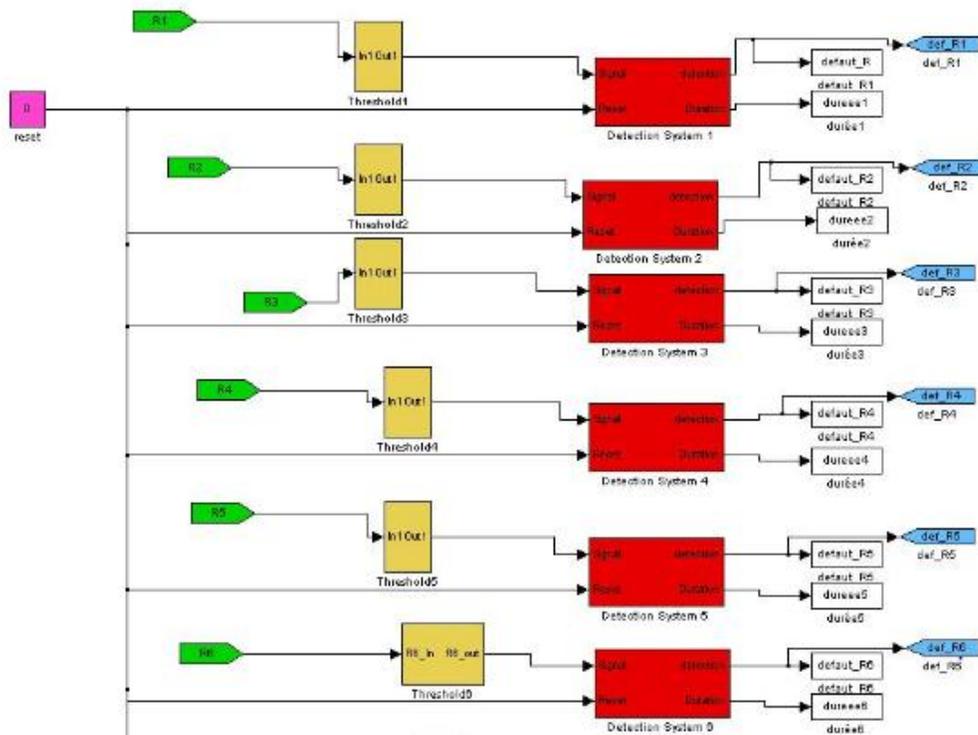
```
If (Arr1==0 )
Temperature=1;
If (Arr1==1 && Arr2=0)
Pression=1;
If (Arr1=1 && Arr2=1)
Leak=1;
```

La complexité du script mise en place pour le système en entier est d'une plus grande complexité (Annexes 1).

Après la mise en place du script permettant la détection des erreurs ainsi que l'isolation de ces derniers. Une interface d'affichage sur Dspace a été mise en place afin de faciliter la compréhension des données obtenue.



Le dernier point étudié et la configuration des seuils de détection. Ce travail permet de détecter le moment de déclenchement des résidus afin de permettre l'utilisation du script. Pour cela un block simulink a été créé. Celui-ci permet d'envoyer un signal binaire au script mise en place après un dépassement des seuils.



Le filtrage effectué dans la première partie du projet a facilité ce travail. Etant donnée la netteté des signaux la mise en place de seuil a été plus facile.

### **Conclusion :**

La première partie du projet consistait à mettre en place les conditions idéales de travail. Le traitement de signal est l'une des filières les plus dominantes du monde de la science électroniques. Certes il est facile de filtrer un signal en imposant un filtre assez puissant, l'inconvénient d'une telle technique est la perte d'un des plus importants critères des signaux. La nature temps réel des signaux utilisé est l'un des caractères les plus importants dans ce projet. Etant donné que le sujet porte sur la supervision et la détection d'erreur, le temps de réponse est une caractéristique primordiale. La deuxième partie du travail n'a été que plus facile en vue des signaux obtenue. Ayant une grande quantité d'équations et de données, le traitement de celle-ci a dû être mise en place à travers un script permettant la détection des erreurs selon l'état des résidus.

# Annexe

```
if ((r1&r2)==1)&& ~((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1)
    L1=1;
else L1=0;
end
if ((r2)==1)&& ~((r1&r2)==1)
    T2=1;
else T2=0;
end
if (r24)==1
    F3=1;
else F3=0;
end
if (r4&r5&r11&r23)==1
    L8=1;
else L8=0;
end
if (( r23==1)&& ~((r4&r5&r11&r23)==1))
    L9=1; else L9=0;
end
if (r4&r5&r7&r12&r16)==1
    P7=1;
else P7=0;
end
if ((r6)==1)&& ~((r3&r4&r5&r6&r7&r12&r16)==1)
    Q4=1;
else Q4=0;
end
if ((r15)==1)&& ~((r15&r16)==1)
    T5=1;
else T5=0;
end
if (r15&r16)==1
    T6=1;
    P7=1;
% Mode dégradé du système
else T6=0;
    P7=0;% la valeur P7 depend de T6 en reconfiguration
end
if (r4&r5&r7&r8)==1 F10=1;
else F10=0;
end
if (r7&r8&r9&r10&r17)==1
    P12=1;
```

# Annexe

```
else P12=0;
end
if (((r18)==1) && ~((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
    P13=1;
else P13=0;
end
if (((r17)==1) && ~((r7&r8&r9&r10&r17)==1))
    P14=0;
else P14=0;
end
if (r13&r19)==1
    P15=1;
else P15=0;
end
if (r1&r2&r8&r9&r10&r18&r19&r20&r21)==1
    P16=1;
else P16=0;
end
if (((r21)==1) && ~((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
    T17=1;
else T17=0;
end
if (r9&r10&r14&r22)==1
    L18=1;
else L18=0;
end
if (((r22)==1) && ~((r9&r10&r14&r22)==1))
    L19=1;
else L19=0;
end
if (((r10)==1) && ~((r7&r8&r9&r10&r17)==1) && ~
((r9&r10&r14&r22)==1) && ~
((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
    F23=1;
else F23=0;
end
if (((r10)==1) && ~((r7&r8&r9&r10&r17)==1) && ~
((r9&r10&r14&r22)==1) && ~
```

```
((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
  T21=1;
  else T21=0;
end
if (((r10)==1)&& ~((r7&r8&r9&r10&r17)==1)&& ~
(r9&r10&r14&r22)==1) && ~
(r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
  T22=1;
  else T22=0;
end
if (((r20)==1)&& ~ ((r1&r2&r8&r9&r10&r18&r19&r20&r21)==1))
  P11=1;
  else P11=0;
end
```