

## LFR-FSW - Support #706

### Pb aux bornes des intervalles pour le rejet des fréquences polluées par les roues inertielles

16/06/2016 07:37 AM - paul leroy

<b>Status:</b>	Closed	<b>Start date:</b>	16/06/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Gerald Saule	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>revision:</b>	r0		

#### Description

Mail Gérald Saule 09/06/2016

J'ai une question au sujet de la prise en compte du "LFR filtering of SC reaction wheel emissions". Cela recoupe le point 1 de la note <https://hephaistos.lpp.polytechnique.fr/redmine/issues/691#note-3> d'Alexis.

Afin de voir quelle convention tu as prise, peux-tu me décrire comment se comportera le FSW dans les deux situations suivantes représentatives (sachant que 14.50Hz est à l'intersection des deux "LFR\_BP2\_F0/LFR\_BP2\_AUTO\_F0" de TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2 pour N=0 et N=1):

\*Pour CP\_RPW\_SC\_RWn\_Fm=14.475Hz: les bornes de l'intervalle pollué par l'émission RW sont CP\_RPW\_SC\_RWn\_Fm+/-SY\_LFR\_SC\_RW\_DELTA\_F (=14.45 et 14.50Hz). Le LFR\_BP2\_AUTO\_F0 (de TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2) pour N=1 va t'il intégrer dans sa moyenne le bin de rang 15 (couvrant [14.5;15.5]) ?

\*Pour CP\_RPW\_SC\_RWn\_Fm=14.525Hz : les bornes de l'intervalle pollué par l'émission RW sont CP\_RPW\_SC\_RWn\_Fm+/-SY\_LFR\_SC\_RW\_DELTA\_F (=14.50 et 14.55Hz). Le LFR\_BP2\_AUTO\_F0 (de TM\_LFR\_SCIENCE\_NORMAL\_BP2\_F2) pour N=0 va t'il intégrer dans sa moyenne le bin de rang 14 (couvrant [13.5;14.5]) ?

NB: Avec SY\_LFR\_SC\_RW\_DELTA\_F=0.025 Hz (default value), et CP\_RPW\_SC\_RW[1-4]\_F[1-2]\_FLAG=enable.

#### History

##### #1 - 16/06/2016 07:47 AM - paul leroy

Questions complémentaires.

- 1) Au niveau des intervalles des FFT, pour f2 par exemple, la bin 0 est la composant continue, la bin 1 va de la fréquence 0Hz à 1Hz et la bin 255 va de 254Hz à 255Hz je pense. Pourquoi ce 0.5Hz? Les limites tombent sur des Hz entiers non?
- 2) Il ne s'agit pas plutôt de LFR\_BP2\_F\*2\*/LFR\_BP2\_AUTO\_F\*2\* au lieu de LFR\_BP2\_F\*0\*/LFR\_BP2\_AUTO\_F\*0\*

##### #2 - 16/06/2016 07:49 AM - paul leroy

- Assignee set to Gerald Saule

##### #3 - 16/06/2016 03:52 PM - thomas chust

- 1) Bah non. Quand on fait une FFT de N points, donc avec  $\Delta f = f_e/N$  de résolution, chacune de ses raie  $f_i$  est **centrée** sur la fréquence  $f_i = i \times \Delta f$ . En d'autres termes, une raie  $f_i$  est la raie correspondant à l'énergie comprise sur la bande de fréquence  $[f_i - \Delta f/2, f_i + \Delta f/2]$ . Non ?

=> pour répondre à Gérald., le bin1 du canal F2 correspond à [0.5;1.5] Hz

Ensuite le choix d'un intervalle ouvert à droite ou à gauche est une convention à se fixer une fois pour toute.

##### #4 - 16/06/2016 10:20 PM - paul leroy

Dans la FFT, le spectre est discrétisé, donc la question des intervalles de fréquences ne se pose peut-être pas. La question équivalente dans le

domaine temporel reviendrait à se demander à quel interval appartient un point qui ne serait pas pris à un instant d'échantillonnage.

Voici un début de réponse sur la façon dont le filtre est généré pour un mask (représenté par un char\*, soit 16 octets), pour une fréquence (fonction définie dans tc\_load\_dump\_parameter.c). A la relecture, j'ai utilisé deux fois la fonction floor, j'aurais peut-être dû utiliser ceil pour la borne sup de l'intervalle à rejeter.

```
void setFBinMask( unsigned char *fbins_mask, float rw_f, unsigned char deltaFreq, unsigned char flag )
{
    /** This function executes specific actions when a TC_LFR_UPDATE_INFO TeleCommand has been received.
    *
    * @param fbins_mask
    * @param rw_f is the reaction wheel frequency to filter
    * @param delta_f is the frequency step between the frequency bins, it depends on the frequency channel
    * @param flag [true] filtering enabled [false] filtering disabled
    *
    * @return void
    *
    */

    float fmin;
    float fMAX;
    int binBelow;
    int binAbove;
    unsigned int whichByte;
    unsigned char selectedByte;
    int bin;

    whichByte = 0;
    bin = 0;

    // compute the frequency range to filter [ rw_f - delta_f/2; rw_f + delta_f/2 ]
    fmin = rw_f - filterPar.sy_lfr_sc_rw_delta_f / 2.;
    fMAX = rw_f + filterPar.sy_lfr_sc_rw_delta_f / 2.;

    // compute the index of the frequency bin immediately below fmin
    binBelow = (int) ( floor( ((double) fmin) / ((double) deltaFreq) );

    // compute the index of the frequency bin immediately above fMAX
    binAbove = (int) ( floor( ((double) fMAX) / ((double) deltaFreq) );

    for (bin = binBelow; bin <= binAbove; bin++)
    {
        if ( (bin >= 0) && (bin <= 127) )
        {
            if (flag == 1)
            {
                whichByte = bin >> 3; // division by 8
                selectedByte = (unsigned char) ( 1 << (bin - (whichByte * 8)) );
                fbins_mask[whichByte] = fbins_mask[whichByte] & (~selectedByte);
            }
        }
    }
}
```

**#5 - 17/06/2016 11:01 AM - thomas chust**

Je réalise qu'on est sur une mauvaise approche:

- la question n'est pas que vau la largeur de la raie  $f_i$  d'une FFT mais exactement l'inverse: lorsqu'on introduit un signal à une fréquence  $f_{RW}$  qu'elles sont les raies de la FFT qui sont touchées ?

Et la réponse me semble bien plus drastique que ce que l'on a envisagé jusq'alors:

- même sans fenêtrage de Hanning, si  $f_{RW}$  tombe exactement entre 2 raies ces 2 raies vont être polluées à égalité => il faut virer ces 2 raies ...  
- avec fenêtrage de Hanning, ce qui est notre cas toujours, même réponse si  $f_{RW}$  tombe entre 2 raies mais lorsque  $f_{RW}$  tombe pile sur une raie, les 2 raies adjacentes ne sont atténuées que par un facteur 0.25 (en énergie, les ASM sont des FFT au carré). Il sera donc peut-être utile d'envisager de virer aussi ces 2 raies adjacentes => 3 raies à virer et pas seulement 1 ...

Une proposition pourrait être la suivante:

- si l'intervalle  $f_i \pm \Delta f_{RW}$  est inclu dans l'intervalle  $f_i \pm \Delta f/m$  (m valant a priori au moins 4) on vire la raie  $f_i$  et les 2 raies adjacentes  $f_{i-1}$  et  $f_{i+1}$   
- dans tous les autres cas en vire les 2 raies qui entourent  $f_{RW}$

**#6 - 21/06/2016 10:05 PM - paul leroy**

Ca me va!

**#7 - 22/06/2016 12:03 PM - thomas chust**

- File *lfr\_mask\_criteria\_hanning.html* added  
- File *lfr\_mask\_criteria\_hanning.ipynb* added  
- Assignee changed from Gerald Saule to paul leroy

Afin de définir un critère un peu objectif de la largeur de "centralité"  $\Delta f/m$  à considérer, j'ai fait un petit notebook, ci-joint. Cela me semble bien clair maintenant et j'ai considéré un critère de 10 %. Paul, toujours partant pour implémenter cela ?

**#8 - 28/06/2016 03:19 PM - paul leroy**

- Assignee changed from paul leroy to Gerald Saule

OK, ça me paraît clair. Je vous mettrai une copie du code implémenté dans l'issue Redmine.

**#9 - 22/06/2018 05:22 PM - bruno katra**

- Status changed from New to Closed

Algo entièrement revu et testé en R3++ OK  
ON CLOTURE

**Files**

<i>lfr_mask_criteria_hanning.html</i>	198 KB	22/06/2016	thomas chust
<i>lfr_mask_criteria_hanning.ipynb</i>	6.37 KB	22/06/2016	thomas chust