

Manuel pour l'utilisateur du logiciel Presto-HF

Fabien Seyfert

9 juillet 2025

Table des matières

1	Introduction	2
2	Algorithmique générale	2
2.1	Détermination des composantes de retard	2
2.2	Complétion analytique	3
2.3	Détermination des shifts en fréquence	5
2.4	Détermination d'un modèle rationnel stable de degré de MacMil- lan donné	6
3	Un exemple pas à pas	7
3.1	Initialisation	7
3.2	Une structure pour les données	7
3.3	Affichage des données	8
3.4	Compensation des composantes de retard et calcul des complétions	8
3.5	Détermination du modèle rationnel	9
3.6	Tout en un	10
3.7	Lien avec RGC	10
4	Description détaillée des procédures principales	10
5	Figures	13

1 Introduction

Le document suivant constitue le manuel utilisateur de la boîte à outils matlab Presto-HF. Cette dernière développée à l’Inria Sophia-Antipolis au sein du projet Miaou a pour but de fournir un ensemble de méthodes pour l’extraction d’un modèle rationnel stable de taille 2×2 à partir de données fréquentielles ponctuelles relatives à la matrice de répartition d’un filtre passe-bande hyperfréquence. Cette boîte à outil utilise le moteur d’approximation rationnelle RARL2. Enfin pour extraire une matrice de couplage à partir d’un modèle rationnel on pourra utiliser RGC.

Le document suivant est structuré en trois parties. Dans un premier temps nous exposerons la démarche mathématique et algorithmique que nous avons choisie d’implémenter au sein de Presto-HF. Ensuite nous décrirons pas à pas le traitement d’un exemple sur la base de données d’un filtre dix pôles. Enfin nous terminerons en détaillant les arguments d’entrée-sortie des principales fonctions de Presto-HF.

2 Algorithmique générale

Au niveau conceptuel la stratégie d’identification se décompose en trois étapes qui sont :

- Identification et compensation des composantes de retard
- Détermination d’un modèle stable d’ordre élevé par complétion analytique des données
- Détermination d’un modèle rationnel stable de degré de MacMillan prescrit.

Dans ce qui suit nous noterons par (w_i) les points de fréquence normalisés correspondant aux mesures. Ceux-ci sont obtenus classiquement par la transformation passe-bas qui envoie la bande passante du filtre sur l’intervalle de fréquences normalisées $[-1, 1]$. Les mesures concernant la matrice de répartition du filtre au point w_i seront notées

$$S(w_i) = \begin{bmatrix} S_{1,1}(w_i) & S_{1,2}(w_i) \\ S_{2,1}(w_i) & S_{2,2}(w_i) \end{bmatrix}.$$

2.1 Détermination des composantes de retard

Nous supposons qu’un « bon » modèle pour nos données mesurées est donné par :

$$H(w) = \frac{1}{q(jw)} \begin{bmatrix} e^{j(w\alpha_{1,1}+\beta_{1,1})}p_{1,1}(jw) & e^{j(w(\frac{\alpha_{1,1}+\alpha_{2,2}}{2})+\beta_{2,1})}p_{1,2}(jw) \\ e^{j(w(\frac{\alpha_{1,1}+\alpha_{2,2}}{2})+\beta_{2,1})}p_{2,1}(jw) & e^{j(w\alpha_{2,2}+\beta_{2,2})}p_{2,2}(jw) \end{bmatrix} \quad (1)$$

où $(p_{i,j}, q)$ sont des polynômes de la variable w , les valeurs $(\alpha_{i,i})$ sont les composantes de retard induites par les accès et les valeurs $(\beta_{i,j})$ modélisent des

shifts en fréquence possiblement induits par les instruments de mesure. Afin de pouvoir nous ramener à un problème d'identification rationnelle il nous faut d'abord identifier les composantes $(\alpha_{i,i})$ et $(\beta_{i,j})$. Nous considérerons dans un premier temps l'identification des composantes de retard $(\alpha_{i,i})$.

Soit I un sous-ensemble des fréquences de mesure défini en fonction du paramètre w_c de la manière suivante :

$$I = \{w_i, |w_i| \geq w_c\}. \quad (2)$$

On choisira w_c de manière à ce que le module de $S_{1,1}$ et $S_{2,2}$ se comporte de manière « régulière » sur I , i.e cela nécessite implicitement la pratique de mesures sur une bande plus large que la seule bande passante du filtre. Pour les fréquences de I nous ferons l'hypothèse forte que les composantes rationnelles de (1) sont bien approchées par les premiers n_c termes de leur développement de Taylor à l'infini (i.e développement en $\frac{1}{w}$). Notons ici que la fonction $e^{j\alpha w}$ n'admet pas de développement de Taylor à l'infini. Nous définissons (ici par exemple pour la voie (1,1)) le problème d'optimisation dont la fonction coût est la suivante :

$$\psi(\alpha) = \min_{(a_0, a_1 \dots a_{n_c-1}) \in \mathbb{C}^{n_c-1}} \sum_{w_i \in I} \left| \sum_{k=0}^{n_c} \frac{a_k}{w_i^k} - S_{1,1}(w_i) e^{-j\alpha w_i} \right|^2 \quad (3)$$

La compensation optimale sera alors définie comme la valeur α_0 de α pour laquelle ψ atteint son minimum sur \mathbb{R} . L'idée intuitive derrière cette manière de faire consiste à exploiter le fait que pour des fréquences assez lointaines de la bande passante le modèle rationnel du filtre se comporte comme un polynôme en $\frac{1}{w}$ de bas degré : la composante de retard perturbe cette propriété et permet ainsi son identification. Si l'analyse théorique de cette méthodologie est encore en cours (pour essayer de quantifier sa précision dans le pire des cas), les expériences menées sur des données théoriques et mesurées plaident très largement en sa faveur ; les valeurs $n_c = 3$ et $w_c = 2.5$ donnent de bons résultats sur l'ensemble des exemples fournis par Alcatel.

On procède de même sur la voie (2,2), ce qui permet donc d'avoir des estimées pour tous les $(\alpha_{i,i})$ du modèle (1). La détermination des $\beta_{i,j}$ se fera après l'étape de complétion analytique qui suit.

2.2 Complétion analytique

A bien y regarder la méthode précédente fournit pour les voies (1,1) et (2,2) un comportement estimé (sous forme de deux polynômes en $\frac{1}{w}$) de ces dernières pour les fréquences en dehors de la bande de mesure. Mais comment s'assurer que ce comportement est bien conforme au caractère stable et dissipatif du filtre étudié ? Un léger perfectionnement du problème d'optimisation associé au terme de droite de (3) permet, comme nous allons le voir, de répondre à la question.

Dans le cas d'une fraction rationnelle la stabilité est caractérisée par l'appartenance des pôles de cette dernière au demi-plan gauche ouvert. Ceci est

équivalent au caractère analytique de cette fonction dans le demi-plan droit. Maintenant, afin d'assurer la convexité du problème de complétion que nous allons poser, nous considérerons un ensemble de fonctions analytiques plus général que les seuls fonctions rationnelles stables. Il s'agit de l'espace de Hilbert H^2 formé des fonctions h analytiques dans le demi-plan droit (ouvert) et dont l'intégrale suivante reste bornée sur toutes les droites verticales du demi-plan droit, i.e

$$\forall \sigma > 0, \int_{-\infty}^{+\infty} \frac{|h(\sigma + iw)|^2}{1 + w^2} dw < \infty. \quad (4)$$

Une propriété importante des fonctions de H^2 est que ces dernières admettent une extension (L^2 intégrable) à l'axe imaginaire. En ce sens H^2 apparaît comme un sous-espace de l'espace de Hilbert L^2 des fonctions complexes définies sur l'axe imaginaire dont la norme dérive du produit scalaire suivant :

$$\langle u, v \rangle = \int_{-\infty}^{+\infty} \frac{u(iw)v(\bar{i}w)}{1 + w^2} dw.$$

Le supplémentaire orthogonal de H^2 dans L^2 est noté \overline{H}_0^2 et on a donc :

$$L^2 = H^2 \oplus \overline{H}_0^2. \quad (5)$$

Plus concrètement l'équation (5) est souvent interprétée comme le fait que toute fonction L^2 intégrable sur l'axe imaginaire se décompose (orthogonalement) comme la somme d'une fonction stable dans H^2 et d'une fonction instable dans \overline{H}_0^2 . On notera P_{H^2} et $P_{\overline{H}_0^2}$ les projections orthogonales de L^2 sur H^2 (resp. sur \overline{H}_0^2).

La variable S désigne maintenant nos données dont les composantes de retard ont été compensées par l'étape précédente. Nous supposons que sur l'intervalle de mesure

$$J = [\min(w_i), \max(w_i)]$$

chacune des composantes de S définit une fonction continue dont la valeur entre deux points de mesure est définie par interpolation (par des splines par exemple). On définit alors un problème de complétion polynomiale sous la forme du programme d'optimisation suivant (ici pour $S_{1,1}$) :

$$\begin{aligned} \min_p \sum_{w_i \in I} |S_{1,1}(w_i) - p(\frac{1}{w_i})|^2 \\ \begin{cases} p \in \mathbb{C}^{n_c}[X] \\ \|P_{\overline{H}_0^2}(S_{1,1} \wedge p)\|^2 \leq E_c \\ \forall w \in J_c, |p(\frac{1}{w})|^2 \leq 1. \end{cases} \end{aligned} \quad (6)$$

Dans ce qui précède $\mathbb{C}^{n_c}[X]$ désigne l'ensemble des polynômes à coefficients complexes de degré au plus n_c , J_c désigne le complémentaire de l'intervalle de mesure J , la notation $S_{1,1} \wedge p$ désigne la fonction définie sur J par $S_{1,1}(w)$ et

sur J_c par $p(\frac{1}{w})$, la double barre $||.||$ désigne la norme L^2 et E_c est un paramètre du problème d'optimisation qui prend ses valeurs dans \mathbb{R}^+ . A haute voix le programme (??) s'énonce comme suit : trouver la complétion polynomiale qui rende le mieux compte de nos données sur I tout en satisfaisant aux contraintes suivantes :

- les données complétées par p ont une partie instable inférieure à E_c
- le module de la complétion reste inférieur à 1 sur J_c .

Le programme d'optimisation (??) prend tout son sens lorsque l'ensemble admissible associé est non-vide, c'est pourquoi Presto-HF résout dans un premier temps le programme suivant :

$$\begin{aligned} \min_p & \|P_{\overline{H}_0^2}(S_{1,1} \wedge p)\|^2 \\ \left\{ \begin{array}{l} p \in \mathbb{C}^{n_c}[X] \\ \forall w \in J_c, |p(\frac{1}{w})|^2 \leq 1. \end{array} \right. \end{aligned} \quad (7)$$

dont l'ensemble admissible n'est pas vide (il contient 0). Si la valeur optimale E_m de (7) est inférieure à E_c alors le programme (??) sera résolu (son ensemble admissible est alors non-vide), dans le cas contraire la solution de (7) est renvoyée.

Comme nous le verrons plus tard dans la description détaillée des procédures de Presto-HF la valeur de E_c est donnée sous la forme d'un pourcentage de la norme L^2 des données sur J . De même la contrainte de 1 pour le module pourra être remplacée par une valeur calculée à partir des valeurs du module des données en bord de bande de mesure. Ceci est particulièrement vrais pour la complétion des éléments diagonaux $S_{1,2}$ et $S_{2,1}$ dont le module est supposé chuter en dehors de la bande de mesure. De plus, pour ces éléments un certain nombre de zéros à l'infini (qui correspondent à des zéros de p en zéro) pourra être prescrit à l'avance par l'utilisateur.

Nous noterons $m_{i,j}$ les quatre polynômes obtenus grâce à l'étape de complétion que nous venons de décrire.

2.3 Détermination des shifts en fréquence

Nous disposons maintenant d'une complétion « réaliste » de nos données pour des fréquences en dehors de la bande de mesure J . Le modèle électrique du filtre spécifie que la matrice S de ce dernier doit être égale à l'identité à l'infinie. Il est donc raisonnable d'identifier

$$\beta_{1,1} = \arg(m_{1,1}(0)) \text{ et } \beta_{2,2} = \arg(m_{2,2}(0)).$$

Pour simplifier les notations notons à nouveau S les données où les voies (1,1) et (2,2) ont été compensées (i.e multipliées par $e^{-i\beta_{1,1}}$ et $e^{-i\beta_{2,2}}$) grâce à la remarque précédente.

La matrice de répartition G d'un filtre idéal est supposée être conservative, ce qui s'exprime mathématiquement par :

$$\overline{G}^t G = Id. \quad (8)$$

Si l'on explicite composantes par composantes l'équation (8) on obtient $|G_{1,1}|^2 = 1$, $|G_{2,2}|^2 = 1$ et l'équation d'orthogonalité suivante :

$$\forall w \in \mathbb{R}, G_{1,1}(iw)\overline{G_{1,2}(iw)} + G_{2,2}(iw)\overline{G_{1,2}(iw)} \quad (9)$$

en rappelant que $G_{1,2} = G_{2,1}$. La compensation $\beta_{2,1}$ sera déterminée pour respecter au mieux la relation d'orthogonalité (9) sur l'ensemble des points de mesure. Une analyse simple montre qu'un tel $\beta_{2,1}$ n'est défini qu'à π près : Presto-HF fera donc ici un choix arbitraire (ce dernier est déterministe et gouverné par un paramètre, voir section 4). Cela est cependant moins grave qu'il n'y paraît puisqu'on montre que ce choix n'influe au final que sur le signe de certains couplages qui lui est déterminé par des considérations technologiques relatives aux vis et iris du filtre.

2.4 Détermination d'un modèle rationnel stable de degré de MacMillan donné

On note à nouveau S les données cette fois compensées par l'ensemble des procédures précédemment décrites et $m_{i,j}$ les polynômes qui déterminent les complétions associées. On définit voie par voie,

$$F_{i,j} = P_{H^2}(S_{i,j} \wedge m_{i,j}).$$

Les fonctions $F_{i,j}$ peuvent être vues comme les représentants stables de nos données initiales. C'est à partir de ces fonctions que nous allons déterminer un modèle rationnel stable de degré de MacMillan fixé de nos données. Pour simplifier on pourra considérer ici que le degré de MacMillan de la matrice de répartition associée à un filtre se confond avec ce qu'on appelle classiquement son ordre, i.e la taille de la matrice de couplage de sa modélisation électrique. Le logiciel RARL2 s'attaque au problème suivant,

$$\min_R |||F - R|||^2 = \sum_{i,j} ||F_{i,j} - R_{i,j}||^2 \quad (10)$$

$$\begin{cases} R \text{ rationnelle de taille } (2 \times 2) \\ \text{Deg}(R) \leq n_f \\ R \text{ stable} \end{cases}$$

où n_f est le degré de MacMillan cible et constitue un paramètre d'entrée du logiciel. Sans rentrer dans les détails de l'algorithme qui sous-tend RARL2 on pourra retenir :

- la paramétrisation des matrices rationnelles stables qu'utilise l'algorithme repose sur le calcul de paramètres de Schur
- afin de garantir la stabilité de l'approximant calculé il est essentiel que $F_{i,j} \in H^2$.

Cette dernière remarque explique notre « acharnement » à obtenir un modèle de nos données dans H^2 avant de passer à l'approximation rationnelle proprement dite.

La boîte à outils Presto-HF fait appel à RARL2 pour l'étape d'approximation rationnelle.

3 Un exemple pas à pas

Nous traiterons ici l'exemple du calcul d'un modèle rationnel à partir des mesures d'un filtre dix pôles fournit par Alcatel.

3.1 Initialisation

Afin de pouvoir appeler les procédures de Presto-HF l'utilisateur doit indiquer à matlab où trouver ces dernières. Il est conseillé de faire cela en appelant la procédure `PrestoInit` située dans le répertoire `presto`. Pour ce faire on pourra par exemple se placer sous matlab dans ce répertoire (utiliser `cd`) et taper

```
>> PrestoInit
```

3.2 Une structure pour les données

Le point d'entrée de Presto-HF est constitué par une structure élémentaire à l'image de la variable `Spb` décrite par ce qui suit,

```
Spb =
```

```
value: [2x2x801 double]  
freq: [1x801 double]  
name: 'alc200601'
```

Le champ `value` de `Spb` contient les mesure de la matrice de répartition sous la forme d'un tableau de dimension $2 \times 2 \times nd$ où nd est le nombre de point de mesures. Dans notre exemple $nd = 801$. Le champ `freq` est un vecteur ligne spécifiant les fréquences **normalisées passe-bas** des différentes mesures. Enfin, le champ `name` spécifie le nom que l'utilisateur veut attribuer à son jeu de données. Ce champ est optionnel au sens où il n'est requis de manière obligatoire par aucune des fonctions. Presto-HF ne possède pour l'instant qu'un parseur élémentaire (appelable par `ParseS`) pour charger des données à partir d'un fichier ascii ; il est donc recommandé à chaque utilisateur d'écrire un parseur pour son format ascii favori.

Pour des raisons de commodité beaucoup des paramètres relatifs aux procédures de Presto-HF ont une valeur par défaut, dont nous pensons qu'elle « fera l'affaire » dans la plus-part des cas. L'utilisateur peut cependant spécifier la valeur de son choix pour ces paramètres en ajoutant un champ `params` à sa structure de départ comme c'est le cas pour la structure `Spb1` ci dessous :

```
Spb1 =
```

```
freq: [401x1 double]
```

```

    value: [2x2x401 double]
    params: @GetMyDefaultValue

```

Le champ `params` est un pointeur de fonction (voir manuel matlab, fonction `handle`) qui indique ici le chemin d'appel d'une procédure qui redéfinit tout ou partie des paramètres de Presto-HF. A titre d'exemple le code de la procédure `GetMyDefaultValue` est ici le suivant,

```

function DVC=GetMyDefaultValue()

%      ***** Completion and delay determination *****

DVC.CDAFS.delay_range=[-0.3:.001:0.3];
DVC.CDAFS.poly_order=4;
DVC.CDAFS.omega_lim=3.3;
DVC.CDAFS.causal_bound(1,1)=0.7;
DVC.CDAFS.causal_bound(2,2)=0.9;
DVC.CDAFS.modulus_factor(1,1)=1.01;
DVC.CDAFS.modulus_factor(2,2)=1.01;

```

La procédure `GetMyDefaultValue` renvoie une structure dont chaque champ correspond à un paramètre que l'utilisateur désire spécifier. Ici par exemple l'utilisateur désire que l'ordre des polynômes utilisés lors de la détermination du retard et de l'étape de complétion soit de 4. La description de l'ensemble des paramètres ainsi que de leurs syntaxes sera faite à la section 4.

3.3 Affichage des données

La procédure `Plot` permet d'avoir accès au diagramme de Nyquist de la structure `Spb` en tapant,

```
>> PlotS(Spb)
```

Matlab renvoie alors une fenêtre graphique constituée du diagramme de Nyquist des 4 voies (voir figure 1). Les indications en pourcentage portées au dessus des diagrammes des voies (1,2) et (2,1) indiquent la différence relative entre les mesures de ces 2 voies : rappelons qu'aux erreurs de mesure près ces deux voies doivent être identiques. Pour obtenir le diagramme de Bode utiliser,

```
>> PlotS(Spb,'b')
```

dont le résultat est visualisé ici sur la figure 2.

3.4 Compensation des composantes de retard et calcul des complétions

La compensation des composantes de retard ainsi que le calcul des complétions sont effectués au sein de la procédure `CompensateDelayAndFreqShift` qui s'utilise comme suit,


```
>> Spbc=CompensateDelayAndFreqShift(Spb);
[1,1]: Err. 1/s serie app. (order 3) 0.618%
No constraint optimisation needed
[1,2]: Err. 1/s serie app. (order 3) 0.006%
[2,1]: Err. 1/s serie app. (order 3) 0.006%
[2,2]: Err. 1/s serie app. (order 3) 0.376%
No constraint optimisation needed
```

Les pourcentages indiqués sur la sortie écran de cette procédure indiquent l'erreur relative entre la complétion et les données sur l'intervalle I (cf. section 2.2). Le commentaire `No constraint optimisation needed` indique qu'aucune des contraintes du programme d'optimisation (??) n'est active. Le résultat peut être visualisé par la commande,

```
>> PlotS(Spb);
```

dont le résultat est la figure 3. On notera que sur cette figure les données sont compensées et la complétion apparaît en trait rouge. Comme le montre la sortie suivante,

```
>> Spbc
```

```
Spbc =
```

```
value: [2x2x801 double]
freq: [1x801 double]
name: 'alc200601'
comp: {2x2 cell}
a: [2x2 double]
b: [2x2 double]
comp_iso_flag: 0
```

la structure `Spbc` est une version enrichie de `Spb`.

3.5 Détermination du modèle rationnel

Avant de passer à l'étape d'approximation rationnelle il nous faut d'abord calculer les coefficients de Fourier de nos données complétées. Ceci est obtenu grâce à la commande suivante,

```
>> Spc=CompFourierCoeffs(Spb);
S(1,1) ratio unstable/norme_data = 0.156581%
S(1,2) ratio unstable/norme_data = 0.098339%
S(2,1) ratio unstable/norme_data = 0.100487%
S(2,2) ratio unstable/norme_data = 0.189753%
```

Les pourcentages de la sortie écran représentent le ratio entre la norme de la partie instable de nos données complétées et la norme des données brutes. Ces ratios sont repris sur les diagrammes associés qui sont à nouveau obtenus par,

```
>> PlotS(Spc)
```

dont la sortie est donnée par la figure 4.

On obtient maintenant un modèle rationnel de degré de MacMillan 10 de la manière suivante,

```
Spbr=RatApp(Spbc,10);
```

et la figure 5 représente le diagramme de Nyquist correspondant obtenu par,

```
>> PlotS(Spbr);
```

Sur cette figure les annotations *err* et *err.sup* indiquent l'erreur relative l^2 et l^∞ entre le modèle rationnel et les données.

3.6 Tout en un

Les opérations que nous venons de décrire peuvent être enchaînées grâce à la procédure `AllSteps` de la manière suivante,

```
>> Spbr=AllSteps(Spb,10);
```

Nous conseillons l'emploi de cette procédure dans le cas général. L'exécution pas à pas sera utile lorsqu'on désire ajuster certains paramètres spécifiques à telle ou telle procédure.

3.7 Lien avec RGC

On peut utiliser RGC pour calculer la forme en flèche et une réalisation contrainte du modèle rationnel calculé par Presto-HF. Afin de faciliter cette étape on prévoit de spécifier et implémenter une interface entre ces deux logiciels.

4 Description détaillée des procédures principales

Comme nous l'avons déjà évoqué Presto-HF utilise pour certaines de ses procédures des valeurs par défaut et des constantes. Les valeurs par défaut sont utilisées lorsqu'une procédure est appelée avec une liste d'arguments non exhaustive. Dans ce cas une valeur par défaut est assignée aux arguments manquants. Les constantes elles ne sont pas accessibles via les arguments d'appel. Cependant l'ensemble des valeurs (par défaut et constantes) sont spécifiées dans un fichier unique qui se trouve sous `presto_default_values/PrestoDefaultValuesAndConstants.m`. Une modification dans ce fichier sera alors répercutée de manière générale à

tous les appels de fonction ultérieurs : cette manière de faire est conseillée si la modification en question a un caractère générale, c'est à dire si l'on pense que dans la plus-part des cas cette valeur est adaptée. Si par contre cette modification concerne le traitement d'un exemple pathologique on fera appel au mécanisme décrit à la section 3.2 et qui passe par la définition d'un champ **params** supplémentaire dans la structure qui contient les données. Dans ce qui suit une liste descriptive des paramètres d'entrées, de leurs valeurs par défaut ainsi que des constantes est donnée pour chaque fonction.

Signature PlotS(S, g_t, l_f)

Sortie(s) aucune

Entrée(s)

S: La structure dont on désire l'affichage graphique

g_t=DVC.PS.graphic_type_flag: Indique si l'on désire un diagramme de Nyquist ou Bode. Valeurs possibles {'b','n'}

l_f=DVC.PS.legend_flag: Indique si une légende doit apparaître pour chaque courbe (data, compl., app. rat.). Valeurs possibles {0,1}

Constante(s) aucune

Description

La fonction PlotS crée une fenêtre graphique contenant les diagrammes de Nyquist ou de Bode correspondant à l'entrée S. Le nombre de courbe tracées dépend de la présence ou non de certains champs dans la structure S (complétion, approx. rationnelle ...)

Signature [So]=CompensateDelayAndFreqShift(S,zeros,plotflag)

Sortie(s)

So: S enrichie d'information sur la compensation du retard et la complétion. Les données de So sont compensées.

Entrée(s)

S: Les données qui doivent être compensées. Ces dernières doivent être contenues dans les champs **freq** et **value** de S.

zeros=DVC.CDAFS.zeros_at_inf: Le nombre de zéros à l'infini pour $S_{1,2}$ et $S_{2,1}$. Cette information est utilisée au moment du calcul de la complétion.

plotflag=DVC.CDAFS.plot_flag: Si plotflag est différent de 0 le résultat est affiché dans une fenêtre graphique

Constante(s)

DVC.CDAFS.sign_in_zero_trans_12: Le shift en fréquence appliqué à $S_{1,2}$ et $S_{2,1}$ est défini à π près. Cette constante fixe le signe de la partie imaginaire de $S_{1,2}$ en zéros (au point de fréquence le plus proche de zéro). Valeurs possibles {-1,1}.

DVC.CDAFS.delay_range: Un ensemble de valeurs possibles pour les composantes de retard. Par exemple [-0.1 :0.001 :0.1].

DVC.CDAFS.poly_order: Degré du polynôme en $1/s$ utilisé pour la complétion.

DVC.CDAFS.omega_lim: Correspond à w_c dans la définition de l'intervalle I , voir (2). **Important** : le cardinal de I doit être assez important pour que l'erreur d'approximation par des polynômes de degré fixé ait un sens, i.e $\text{card}(|w_k| > \text{omega_lim}) \gg \text{poly_order}$.

DVC.CDAFS.error_lim: Si l'erreur d'approximation à compensation optimale est supérieure à error_lim alors un warning est envoyé vers la fenêtre de commande.

DVC.CDAFS.causal_bound(1,1): Correspond à E_c dans le programme d'optimisation ?? et concerne la complétion de la voie (1,1). Ici cette borne est exprimée en pourcentage de la norme L^2 des données.

DVC.CDAFS.causal_bound(2,2): Même chose mais pour la voie (2,2).

DVC.CDAFS.modulus_factor(1,1): La contrainte sur le module de la complétion est exprimée ici de la manière suivante : $\text{modulus_factor} * \max_w(|S_{1,1}|)$.

DVC.CDAFS.modulus_factor(2,2): Même chose que précédemment mais pour $S_{2,2}$.

DVC.CDAFS.number_of_control_points: Nombre de points de contrôle utilisés pour la contrainte en module. Ce nombre devra être choisi en fonction de DVC.CDAFS.poly_order (de bons résultats sont observés pour $\text{number_of_control_points} \geq 5 * \text{poly_order}$).

DVC.CDAFS.number_of_fourier_coeffs: Nombre de coefficients de Fourier utilisés pour estimer la partie instable des données.

DVC.CDAFS.iso_flag: Permet une variation sur la norme pour l'approximation. Pas décrit en détail ici.

Description none

Rend effectif les algorithmes de compensation et de complétion explicités aux sections 2.1,2.2,2.3.

Signature [So]=CompFourierCoeffs(Si)

Sortie(s)

So: Si enrichie des coefficients de Fourier de chaque voie

Entrée(s)

Si: Structure de données avec informations sur la complétion

Constante(s)

DVC.CFC.n_Fourier: Nombre de coefficients. de Fourier à calculer

DVC.CFC.n_comp: Nombre de points utilisés pour évaluer la complétion

DVC.CFC.n_spline: Nombre de points utilisés pour l'évaluation des intégrales de Fourier.

Description

La procédure calcule des coefficients de Fourier d'indice positif et négatif. Les coefficients d'indice positif seront utilisés lors de la phase d'approximation rationnelle, ceux d'indice négatif servent à calculer la norme de la partie instable des données complétées.

Signature [SS]=RatApp(S,n,Is_type,solver)

Sortie(s)
SS: S enrichie par les numérateurs et dénominateur du modèle rationnel

Entrée(s)
S: Structure de données pour laquelle les coefficients de Fourier ont déjà été calculés.
n: Degré de MacMillan cible

Constante(s)
Is_type=DVC.RA.divide_by_z_minus_one: Information sur le type d'isométrie utilisée pour passer au cercle. Pas décrit ici en détail.
solver=DVC.RA.solver_flag: Spécifie quel moteur d'approximation rationnel doit être utilisé : RARL2 ou hyperion. Valeurs possibles {'RARL2','HARL2'}

Description
Le moteur hyperion n'est disponible que sous Linux.

Signature [S]=AllSteps(S,n,zeros_at_inf,solver_flag,plot_flag)

Sortie(s)
S: S enrichie par les étapes de compensation-complétion, de calcul coefficients de Fourier de la phase d'approximation rationnelle

Entrée(s)
S: Structure contenant les données passe-bas
n: Degré de MacMillan cible
zeros_at_inf=DVC.AS.zeros_at_inf: Nombre de zéros à l'infini pour la complétion des voies (1,2) et (2,1)
solver_flag=DVC.AS.solver_flag: Type de moteur utilisé par l'approximation rationnelle. Valeurs possibles {'RARL2','HARL2'}
plot_flag=DVC.AS.plot_flag: Visualisation ou non des résultats. Valeurs possibles {0,1}

Constante(s) aucune

Description
Enchaîne les procédures CompensateDelayAndFreqShift, CompFourierCoeffs et RatApp.

5 Figures

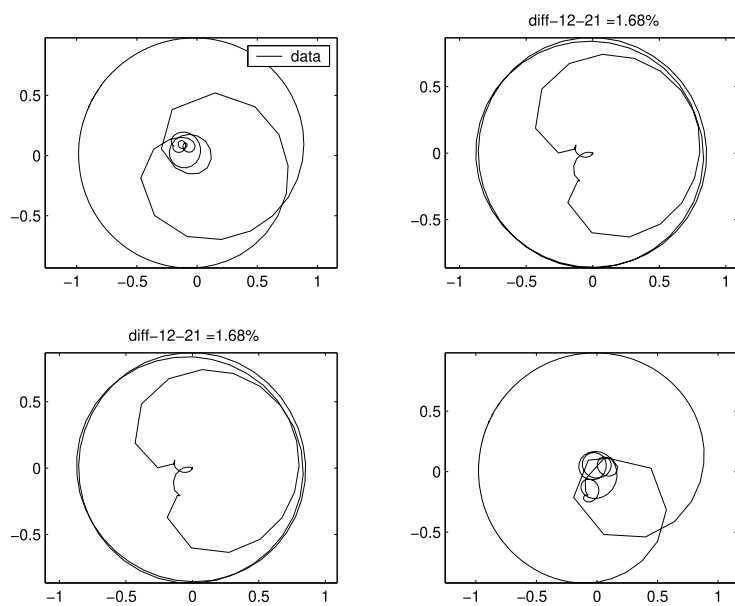


FIGURE 1 – Diagramme de Nyquist des données

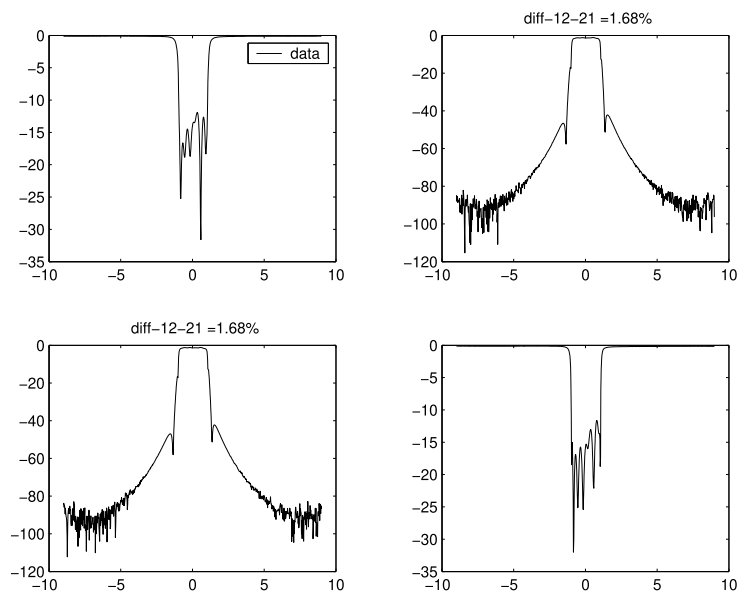


FIGURE 2 – Diagramme de Bode des données

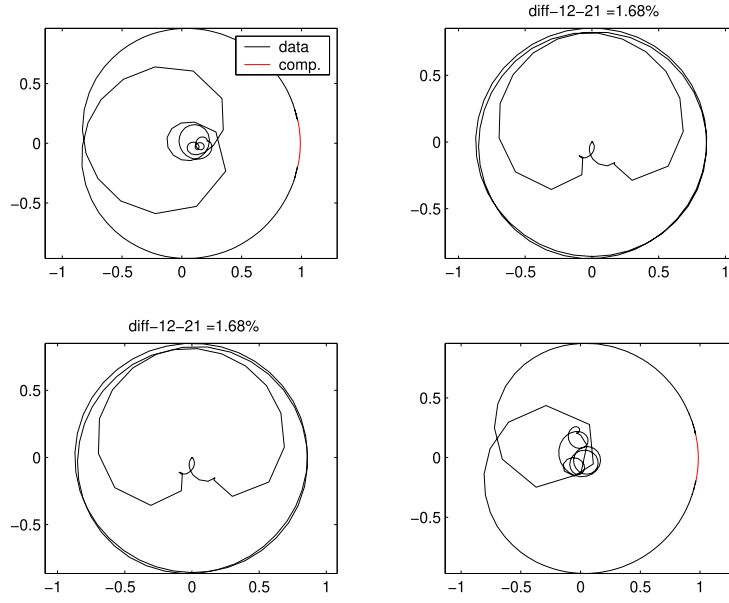


FIGURE 3 – Complétion et données

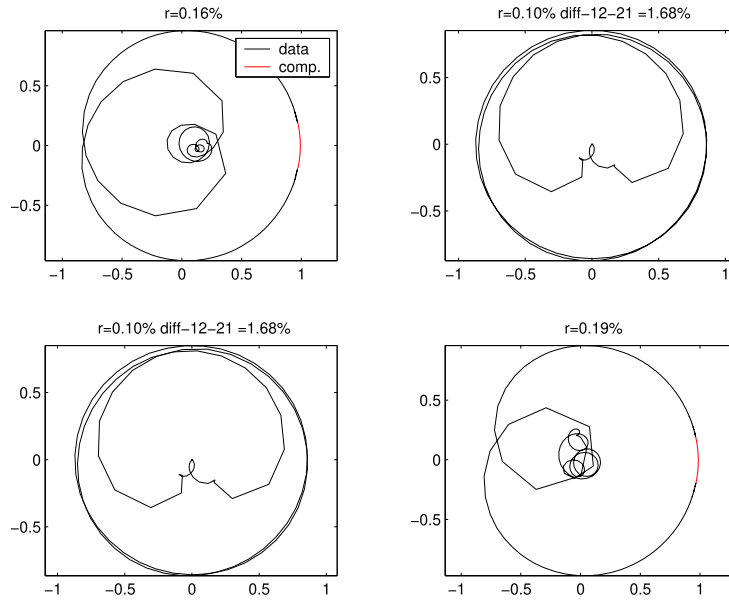


FIGURE 4 – Complétion et données après calcul coeff. Fourier

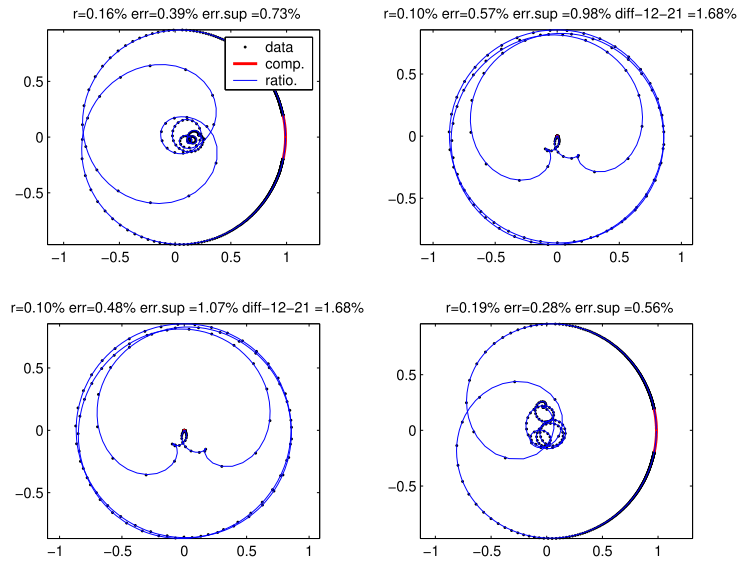


FIGURE 5 – Complétion, données et approximation rationnelle