

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERRI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Recueil de

P.V DE DELIBERATION

De Soutenances de Projets de Fin d'Etudes
D'Ingénieurs

Promotion 2010

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERY, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

Du Master Académique en Automatique
Option Commande des Systèmes

Thème

*Conception et réalisation d'un logiciel sous
Matlab dédié à l'analyse et la simulation des
systèmes dynamiques d'ordre fractionnaire*

Proposé par : Pr. Saïd DJENNOUNE

Présenté par :

Dirigé par : Pr. Saïd DJENNOUNE

M^{elle} Zina BOUDJEMAA
M^{elle} Chaféa HADDAD

Soutenu le : / 06 / 2010

Promotion 2010



Remerciements

Nous remercions avant tout le Bon Dieu de nous avoir permis de voir cet instant tant attendu.

Au terme de ce modeste travail, nous adressons nos vifs remerciements à Monsieur Saïd. DJENNOUNE d'avoir proposé et dirigé ce mémoire. Nous lui sommes très reconnaissantes pour sa gentillesse, sa disponibilité et ses précieux conseils et surtout pour sa confiance qu'il nous a témoignée.

Nos sincères remerciements envers M. MANSOURI pour ses précieuses aides, nous aimions témoigner sa générosité et sa disponibilité.

Nous témoignons notre reconnaissance à M. GUERMAH d'avoir participé à la correction de notre mémoire.

Nous tenons à remercier en particulier Mlle AÏF MESSAOUD pour son aide et sa gentillesse.

Nous remercions M. MAJDI et Mme DJAMAÏ pour leurs conseils et orientations.

Nos remerciements vont également aux présidents ainsi qu'aux membres du jury pour l'honneur qu'ils nous font de juger notre travail.

Sans oublier de remercier tous les enseignants ayant contribué à notre formation de près ou de loin.

Et tous les amis et camarades, qui ont de près ou de loin contribué à l'aboutissement de ce travail, ne serait ce par un modeste sourire d'encouragement à notre égard.

Dédicaces

À la mémoire de mon cher père, que j'ai tant souhaité lui faire vivre cet événement important dans ma vie.

À ma très chère adorable maman à qui je suis très reconnaissante, à celle qui est toujours présente et continue de l'être pour faire mon bonheur. Merci pour t'être sacrifiée pour que tes enfants grandissent et prospèrent. Merci de trimer sans relâche, malgré les péripéties de l'âge, de la santé, de la vie, au bien-être de tes enfants. Enfin ! Merci tout simplement d'être... ma mère.

À tous mes frères spécialement Karim

À mes sœurs Fassadit, Sissi, Lamia, Sophia, Sara et Siham .

À mon oncle Krim, ma tante Taous, cousins et très adorables cousines

À la mémoire de ma grande mère.

Vous avez de près ou de loin contribué à ma formation.

Affectueuse reconnaissance et sincère gratitude.

À mes Amis Mohand, Med-Said, et Amar

À mes amies Chaféa, Samia, Fatma, Sabrina, Faisa, et Samira

Mes copines de chambre : Manoun, Souad, Celia, Djidji et à leurs familles.

À mes camarades de la section Master Automatique, Master Electrotechnique et tous ceux de la faculté de Génie Electrique et d'Informatique et tous mes enseignants de l'Université Mouloud Mammeri de Fizi-Cuzou.

Je dédie ce travail à mon très chère binôme Chaféa et sa famille.

Zina

Dédicaces

À la mémoire de mon père.

Et j'en viens à ma famille, à ma mère, à mon adorable mère, à celle qui est toujours présente et continue de l'être pour faire mon bonheur. Merci pour t'être sacrifiée pour que tes enfants grandissent et prospèrent. Merci de trimer sans relâche, malgré les péripéties de l'âge, de la santé, de la vie, au bien-être de tes enfants. Enfin ! Merci tout simplement d'être... ma mère. Merci aussi à mes deux frères Mohand-Arezki et Farid. Vous vous êtes dépensés pour moi sans compter. Merci d'être toujours à mes côtés, par votre présence, par votre amour, pour donner du goût et du sens à notre vie de famille.

Avec toute ma tendresse.

À mes neveux et nièces.

À mes oncles, tantes, cousins et cousines.

Vous avez de près ou de loin contribué à ma formation.

Affectueuse reconnaissance et sincère gratitude.

À mes amis Amar, Mohand, Med-Said, Dahman...

À mes amies Zina, Hassina, Miassa, Sabrina, Djahida, Amel,

Sabiha, Lamia.....

Mes copines de chambre : Manoun, Souad, Celia, Djidji et à leurs familles.

À mes camarades de la section Master Automatique et tous ceux de la faculté de Génie Electrique et d'Informatique et tous mes enseignants de l'Université Mouloud Mammeri de Fizi-Ouzou.

Je dédie ce travail à mon très chère binôme Zina et sa famille.

Chaféa

Sommaire

Table des matières

Notations

Introduction générale

Chapitre I : Généralités sur le calcul et les systèmes fractionnaires

I. Introduction	3
II. Intégrale fractionnaire	3
II.1. Définition.....	3
II.2. Propriétés	4
III. Dérivation fractionnaire	5
III.1. Définition de Riemann Liouville.....	5
III.2. Définition de Caputo.....	7
III.3. Définition de Grünwald Letnikov	9
III.4. Propriétés	11
IV. Utilisation des opérateurs d'intégration et de dérivation non entière	12
IV.1. Exemple de modélisation du phénomène de diffusion de chaleur par un modèle non entier.....	13
V. Systèmes d'ordre fractionnaire	16
V.1. Equation différentielle linéaire d'ordre non entier	17
V.2. Fonction de transfert d'ordre non entière	17
V.3. Modèle d'état non entier	18
V.4. Propriétés des systèmes fractionnaires	19
V.4.1. Stabilité des systèmes fractionnaire	19
V.4.2. Commandabilité et observabilité	20
VI. Conclusion	22

Chapitre II : Développement des algorithmes

I. Introduction	23
II. Calcul de la dérivée et de l'intégrale non entière.....	23
III. Résolution d'une équation différentielle linéaire d'ordre non entier	25
IV. Résolution d'une équation polynômiale à puissance non entière.....	28
IV.1. Propriétés	31
V. Calcul de la réponse temporelle du modèle d'état non entier	32
VI. Passages entre les différentes représentations d'un système fractionnaire	34
VI.1. Equation différentielle d'ordre non entier vers une représentation d'état non entière	34
VI.1.1. Equation différentielle d'ordre non entier vers une fonction de transfert d'ordre non entier	34
VI.1.2. Fonction de transfert non entière vers une représentation d'état non entière.....	35
VI.2. Modèle d'état non entier vers une fonction de transfert	40
VII. Approximations d'un dérivateur d'ordre non entier.....	41
VII.1. La méthode de Carlson	41
VII.2. Méthode d'approximation d'Oustaloup	42
VII.3 Méthode d'approximation de Charef.....	45
VII.4 Méthode d'Oustaloup modifiée.....	48
VIII. Conclusion	51

Chapitre III : Réalisation du logiciel et application

I. Introduction	52
II. Création d'une interface graphique sous Matlab	52
III. Eléments d'une interface graphique	53
III.1. Les contrôles	53
III.2. Les menus	54
IV. Exemple de création d'une interface.....	55
V. Présentation générale du logiciel	59

VI. Application	61
VI.1. Calcul fractionnaire	61
VI.1.1. Dérivation/Intégration fractionnaire	61
VI.1.2. La transformée de Laplace (TL) de la dérivée /intégrale fractionnaire..	64
VI.1.3. Calcul des racines d'un polynôme d'ordre fractionnaire	64
VI.2. Modèles de représentation.....	66
VI.2.1. Passage d'une équation différentielle d'ordre fractionnaire vers une fonction de transfert.....	67
VI.2.2. Passage de la représentation transfert non entière à la représentation d'état non entière.....	68
VI.2.3. Passage d'une équation différentielle non entière vers une représentation d'état	71
VI.2.4. Passage d'un modèle d'état d'ordre non entier vers une fonction de transfert	72
VI.3. Analyse et simulation des systèmes fractionnaires	74
VI.3.1. Analyse fréquentielle sans approximation	74
VI.3.2 Analyse fréquentielle avec approximation.....	75
VI.3.3. Etude et analyse d'un modèle d'état fractionnaire.....	76
VI.3.4. Résolution numérique d'une équation différentielle	77
VI.4. Méthodes d'approximation d'un dérivateur généralisé	79
VI.4.1. Méthode de Carlson	79
VI.4.2. Approximation d'Oustaloup	80
VI.4.3. Approximation de Charef	81
VI.4.4. Approximation d'Oustaloup modifiée.....	82
VI.5. Présentation de l'aide	83
VII. Conclusion	84
Conclusion générale	85
Bibliographie	87

ANNEXES

A. Outils mathématiques de base

A.1. La fonction Gamma..... 91

A.1.1. Définition..... 91

A.1.2. Propriétés..... 91

A.2. Mittag-Leffler 92

B. Méthodes d'approximation

B.1. Méthode de Carlson 93

B.2. Méthode d'Oustaloup 95

B.3. Méthode de Charef..... 98

C. Table des dérivées d'ordre fractionnaire de quelques fonctions usuelles..... 102

D. Table des transformées de Laplace et Laplace inverse..... 103

Notations

Notations

- $I^k f(t)$: ($k \in \mathbb{N}$), l'intégration répétée k fois de la fonction $f(t)$
- $I^\alpha f(t)$: ($\alpha \in \mathbb{R}$), l'intégration non entière d'ordre α de la fonction $f(t)$
- $\Gamma(\lambda)$: ($\lambda \in \mathbb{R}^* \setminus \mathbb{Z}_-$) fonction Gamma.
- $y_\alpha(t)$: facteur d'oubli
- \mathcal{L} : symbole de la transformation de Laplace.
- \mathcal{L}^{-1} : symbole de la transformation de Laplace inverse
- ${}^R D_t^\alpha f(t)$: dérivée d'ordre α de la fonction $f(t)$ nulle pour $t \leq t_0$ selon la définition de Riemann
- ${}^C D_t^\alpha f(t)$: dérivée d'ordre α de la fonction $f(t)$ nulle pour $t \leq t_0$ selon la définition de Caputo
- $\binom{n}{j}$: ($j \in \mathbb{N}$), désigne la combinaison de j élément parmi n
- $\binom{\alpha}{j}$: ($\alpha \in \mathbb{R}$), désigne le binôme de Newton généralisé à des ordres réels
- D^α : opérateur de dérivation d'ordre non entier α
- $D^\alpha f(Kh)$: désigne la valeur de la dérivée $\alpha^{\text{ème}}$ de $f(t)$ à l'instant Kh
- h : période d'échantillonnage
- s : opérateur de Laplace

$D^{(\alpha)}(x)$:	$(x \in \mathbb{R}^n, \alpha \in \mathbb{R}_+^n)$, le $i^{\text{ème}}$ élément du vecteur $x(t)$ est dérivé à la $i^{\text{ème}}$ composante du vecteur α
$D^\alpha x$:	$(x \in \mathbb{R}^n)$, tous les éléments du vecteur $x(t)$ sont dérivés au même ordre α
I_k :	matrice identité de dimension k
\mathcal{C} :	matrice de commandabilité
θ :	matrice d'observabilité
$\Delta_{ne}(s)$:	polynôme d'ordre non entier de variable s
$\Delta_{fr}(s)$:	polynôme d'ordre fractionnaire de variable s
$\Delta_{en}(p)$:	polynôme d'ordre entier de variable s
$ s $:	module du nombre complexe s
$\arg(s)$:	argument du nombre complexe s
E_α :	fonction Mittag Leffler
$D_{borné}(s)$:	fonction de transfert du dérivateur généralisé borné en fréquences
$D_{borné}(s)$:	fonction de transfert du dérivateur généralisé
δ et η :	paramètres de récurrence de l'approximation

Tables des figures

I.1 : Diffusion de chaleur dans un mur.....	12
I.2: Domaine de stabilité des systèmes commensurables dans le plan complexe.....	18
II.1: Coupure du plan complexe suivant l'axe	29
II.2 : Diagramme asymptotique de Bode de $D_{borné}(s)$ et de $D_N(s)$	43
II.3 : Diagramme asymptotique d'amplitude de Bode de $H(s)$ et son approximation.....	46
II.4 : Interface d'accueil du logiciel.....	59
III.1 : Configuration principale du GUIDE de Matlab.....	55
III.2: Création d'une nouvelle interface graphique sur le GUIDE.....	55
III.3 : Propriétés d'une interface graphique.....	56
III.4: Réalisation d'une interface graphique.....	57
III.5 : Programme ajouté au fichier « exemple.m ».....	58
III.6 : Interface qui simule une fonction $f(t)$	58
III.7 : Interface d'accueil du logiciel.....	59
III.8 : Interface du menu principal.....	60
III.9 : Interface de sélection pour le calcul fractionnaire.....	61
III.10 : L'interface de la dérivée fractionnaire	62
III.11: L'interface de l'intégrale d'ordre fractionnaire.....	63
III.12: Courbes d'intégrale pour différents ordres fractionnaires.....	63
III.13 : L'interface de la TL de la dérivée/intégrale fractionnaire.....	64
III.14 : L'interface de résolution d'un polynôme fractionnaire.....	66
III.15 : Interface de sélection des modèles de représentation.....	67
III.16 : L'interface du passage d'une équation différentielle vers une fonction de transfert.....	68
III.17: L'interface du passage d'une fonction de transfert vers une représentation d'état (cas commensurable).....	69
III.18 :L'interface du passage d'une fonction de transfert vers une représentation d'état (cas non commensurable).....	71
III.19 : L'interface de passage d'une équation différentielle d'ordre fractionnaire vers une représentation d'état.....	71

III.20 : L'interface du passage d'une représentation d'état vers une fonction de transfert (cas commensurable).....	72
III.21 : L'interface du passage d'une représentation d'état vers une fonction de transfert(cas non commensurable)	73
III.22 : L'interface de sélection pour l'analyse et la simulation des systèmes	
III.23 : L'interface de l'analyse fréquentielle (Sans approximation).....	75
III.24 : L'interface de l'analyse fréquentielle (avec approximation).....	76
III.25 : L'interface de l'analyse d'un modèle d'état fractionnaire	77
III.26 :L'interface résolution d'une équation différentielle d'ordre commensurable	78
III.27 :L'interface résolution d'une équation différentielle d'ordre non commensurable	78
III.28 : L'interface de sélection pour les méthodes d'approximation d'un dérivateur généralisé	79
III.29 : L'interface de l'approximation de Carlson.....	80
III.30 : L'interface d'approximation d'Oustaloup.....	81
III.31 : L'interface d'approximation de Charef	82
III.32 : L'interface d'approximation d'Oustaloup modifiée.....	83
III.33 : Exemple d'interface de l'aide.....	84
III.34 : Exemple d'interface d'un message d'erreur.....	84

*Introduction
générale*

Quand on introduit la notion de dérivée, on se rend vite compte qu'on peut appliquer le concept de dérivée à la fonction dérivée elle-même et par là, même introduire la dérivée seconde, puis les dérivées successives d'ordre entier. L'intégration, opération inverse de la dérivée, peut éventuellement être considérée comme une dérivée d'ordre "moins un". On peut aussi se demander si ces dérivées d'ordres successifs ont un équivalent d'ordre fractionnaire. Selon un exposé historique détaillé dans [27] [22], la dérivation numérique d'ordre fractionnaire remonte à diverses correspondances entre Gottfried Leibniz, Guillaume de l'Hospital et Johann Bernoulli à la fin du 17^{ème} siècle. En 1695, L'Hospital a adressé une lettre à Leibniz (l'inventeur de la notation de la dérivée $(d^n y/dx^n)$) posa la question, "Qu'en est il si $n = 1/2$?", question à laquelle Leibniz répondit : "le résultat de $d^{1/2}x$ sera égal à $x\sqrt{dx}:x$, un paradoxe duquel des conséquences utiles seront un jour tirées".

On pourrait penser que cette recherche de dérivation fractionnaire est une question de mathématiques "pures" sans intérêt pour l'ingénieur. Cependant le calcul fractionnaire relève d'un thème de recherche récent motivé par la modélisation d'un nombre important de phénomènes physiques et plus particulièrement les phénomènes de diffusion et aussi la conception des régulateurs robustes. C'est pour cela que l'intérêt de la dérivation non entière ne cesse de grandir, notamment dans le domaine de l'automatique pour la modélisation, l'identification et la commande des systèmes.

Vu l'importance du calcul fractionnaire, il est indispensable de mettre en œuvre un logiciel d'analyse et de simulation, notre objectif est de réaliser une boîte à outils d'analyse et de simulation des systèmes dynamiques d'ordre fractionnaire qui s'inscrit dans ce sens.

Le travail que nous présentons se résume en trois chapitres, dans le premier chapitre nous donnons quelques notions fondamentales des systèmes d'ordre fractionnaires, les différents modèles de représentation des systèmes dynamiques fractionnaires et leurs propriétés.

Le deuxième chapitre est consacré au développement des algorithmes numériques nécessaires à la simulation des systèmes d'ordre non entier, ainsi que nous présentons les passages entre les différentes représentations des systèmes fractionnaires et quelques méthodes d'approximation des systèmes fractionnaires.

Le dernier chapitre présente notre logiciel qui est dédié à l'analyse et la simulation des systèmes dynamiques d'ordre non entier.

Pour finir, nous avons présenté une synthèse générale de notre travail qui fait l'objet d'une conclusion générale et de quelques perspectives.

A ce sujet difficile. Bonne lecture !

Chapitre I :
***Généralités sur le
calcul et les systèmes
fractionnaires***

I. Introduction

L'objectif de ce chapitre est d'introduire les définitions de base et les propriétés fondamentales des opérateurs intégr-différentiateurs d'ordre non entier et d'illustrer par quelques exemples ces opérateurs. Nous motivons l'utilisation de cet outil dans la modélisation de quelques systèmes physiques.

L'opérateur d'intégration et la dérivation d'ordre non entier est la généralisation de la fonction de dérivation entière à des ordres non entiers quelconques. Cette généralisation peut être obtenue à partir de l'intégration non entière, donnant ainsi la définition de Riemann-Liouville et la définition de Caputo [23, 14, 29]. Une autre généralisation basée sur la définition usuelle de la dérivation entière, est proposée par Grünwald-Letnikov [27, 29, 6].

Nous avons consacré la fin de ce chapitre pour la présentation des différents modèles de représentation des systèmes dynamiques fractionnaires [22, 35] et leurs propriétés (stabilité, commandabilité et observabilité).

II. Intégrale fractionnaire

II.1. Définition [23]

L'intégration répétée k (entier naturel) fois de la fonction $f(t)$ réelle de la variable réelle t continue et intégrable sur $[t_0, +\infty[$ s'exprime par :

$$I_{t_0}^k f(t) = \int_{t_0}^t dt_n \int_{t_0}^{t_n} dt_{n-1} \dots \int_{t_0}^{t_3} dt_2 \int_{t_0}^{t_2} f(t_1) dt_1 = \frac{1}{(k-1)!} \int_{t_0}^t (t - \tau)^{k-1} f(\tau) d\tau \quad (\text{I.1})$$

t_0 : Instant initial.

Avec la généralisation de la fonction factorielle à un ordre réel donnée par la fonction Gamma [29, 34], on obtient la fonction d'intégration non entière suivante :

$$I_{t_0}^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t (t - \tau)^{\alpha-1} f(\tau) d\tau, \alpha \in \mathbb{R}_+^* \quad (I.2)$$

Γ : étant la fonction d'Euler définie par :

$$\Gamma(\lambda) = \int_0^{+\infty} v^{\lambda-1} e^{-v} dv \quad (I.3)$$

La définition (I.2) peut être interprétée comme l'aire de la surface que définit la fonction $f(t)$ pondérée par le facteur d'oubli représenté par la fonction :

$$y_{\alpha}(t) = \frac{t^{\alpha-1}}{\Gamma(\alpha)} \quad (I.4)$$

La formule (I.2) peut être aussi interprétée comme le produit de convolution de la fonction causale $f(t)$ et la fonction causale $y_{\alpha}(t)$, qui est exprimée comme suit :

$$I_{t_0}^{\alpha} f(t) = y_{\alpha}(t) * f(t) \quad (I.5)$$

Pour $t_0 = 0$, on note $I_{t_0}^{\alpha} = I^{\alpha}$

Si $\alpha = 1$, $I^1 f(t)$ est une intégrale classique.

II.2. Propriétés [23 ,29]

a) Composition de deux intégrales non entières

Les opérateurs d'intégration non entière vérifient la propriété de semi-groupe soit :

$$I_{t_0}^{\alpha_1} f(t) \circ I_{t_0}^{\alpha_2} f(t) = I_{t_0}^{\alpha_1} (I_{t_0}^{\alpha_2} f(t)) = I_{t_0}^{\alpha_1 + \alpha_2} f(t) \quad \text{avec} \begin{cases} \alpha_1 > 0 \\ \alpha_2 > 0 \end{cases} \quad (I.6)$$

b) Transformée de Laplace

L'interprétation de l'équation (I.2) comme un produit de convolution permet le calcul de la transformée de Laplace de l'intégrale d'ordre réel d'une fonction temporelle causale.

Lemme 1 [35] :

La transformée de Laplace de la fonction t^α ($n \in \mathbb{R}_+$) est

$$\mathcal{L}\{t^\alpha\} = s^{-\alpha-1}\Gamma(\alpha) \quad (I.7)$$

Soit alors

$$\mathcal{L}\{y_\alpha(t)\} = \mathcal{L}\left\{\frac{t^{\alpha-1}}{\Gamma(\alpha)}\right\} = s^{-\alpha} \quad (I.8)$$

La transformée de Laplace de l'opérateur d'intégration fractionnaire s'écrit sous la forme :

$$\mathcal{L}\{I^\alpha f(t)\} = \mathcal{L}\{y_\alpha(t) * f(t)\} = \mathcal{L}\{y_\alpha(t)\} \cdot \mathcal{L}\{f(t)\} = s^{-\alpha} F(s) \quad (I.9)$$

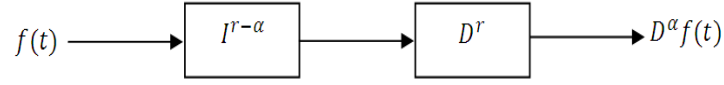
III. Dérivation fractionnaire

Différentes approches ont été utilisées pour généraliser la notion de dérivation aux ordres non entiers :

- La limite du taux d'accroissement d'une fonction se généralise sous la formule de Grünwald-Letnikov, qui est très utile dans le calcul numérique.
- L'intégration qui est l'opération inverse de la dérivation, mène à travers la formule intégrale de Liouville aux formules de Riemann Liouville et de Caputo.

III.1. Définition de Riemann Liouville

La dérivée à un ordre α suivant la définition de Riemann-Liouville [23] est obtenue en intégrant d'abord la fonction $f(t)$ à l'ordre non entier ($r - \alpha$), puis en dérivant le résultat ainsi obtenu à l'ordre entier (r), tel que $r - 1 < \alpha < r$ $\alpha \in \mathbb{R}$, $r \in \mathbb{N}$.



Elle est notée par :

$$\left\{ \begin{array}{l} {}_{t_0}^R D_t^\alpha f(t) = \frac{d^r}{dt^r} \left\{ \frac{1}{\Gamma(r-\alpha)} \int_{t_0}^t (t-\tau)^{r-\alpha-1} f(\tau) d\tau \right\} \\ r-1 < \alpha < r \end{array} \right. \quad (\text{I.10})$$

${}_{t_0}^R D_t^\alpha f(t)$: Symbolise la dérivée d'ordre non entier α par rapport à t de la fonction $f(t)$ entre t_0 et t selon la définition de Riemann Liouville.

La relation (I.10) est aussi notée :

$${}_{t_0}^R D_t^\alpha f(t) \triangleq D_t^\alpha I_{t_0}^{r-\alpha} f(t) \quad (\text{I.11})$$

La transformée de Laplace de la dérivée d'ordre α par rapport à t , de la fonction $f(t)$ causale selon la définition de Riemann est donnée [23] par :

$$\mathcal{L}\{ {}_{t_0}^R D_t^\alpha f(t) \} = s^\alpha F(s) - \sum_{k=0}^{r-1} s^{r-k-1} D^k (I_{t_0}^{r-\alpha} f(t)) \Big|_{t=0} \quad (\text{I.12})$$

A titre d'illustration supposons qu'on veuille calculer la dérivée à l'ordre $\alpha = 3,7$ d'une fonction $f(t)$ selon la définition de Riemann, les étapes sont :

- 1) calcul de l'entier r à partir de la condition de l'expression (I.10), on obtient :

$$r = 4.$$

- 2) On intègre la fonction $f(t)$ à l'ordre $r - \alpha = 0,3$

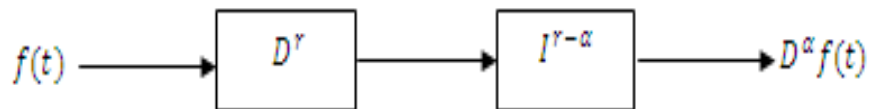
$$g(t) = I^{0,3} f(t)$$

- 3) On dérive la fonction résultante $g(t)$, à l'ordre $r = 4$, comme suit :

$${}^R D^{3,7} f(t) \triangleq D^4 I^{0,3} f(t) = D^4 g(t)$$

III.2. Définition de Caputo [23,34]

Contrairement à la définition de Riemann-Liouville appelée aussi dérivée à gauche, la dérivée de Caputo est obtenue en dérivant d’abord la fonction à l’ordre non entier puis en intégrant le résultat obtenu à l’ordre non entier .



Elle est notée :

$$(I.13)$$

Définie par :

$$\text{---}$$

$$(I.14)$$

: Symbolise la dérivée d’ordre non entier par rapport à de la fonction entre et selon la définition de Caputo.

La notation représente la dérivée à l’ordre entier de la fonction .

Remarque :

La dérivée de Caputo nécessite que la fonction et ses dérivées successives soient nulles pour , ce qui la rend plus restrictive que la définition de Riemann-Liouville.

La transformée de Laplace de la dérivée non entière d’une fonction causale selon la définition de Caputo est donnée par la relation bien connue :

$$\mathcal{L}\{ {}^C D_t^\alpha f(t) \} = s^\alpha F(s) - \sum_{k=0}^{r-1} s^{\alpha-1-k} D^k f(t) \Big|_{t=0} \quad (\text{I.15})$$

Où : $D^k f(t)|_{t=0}$ représente la $k^{\text{ème}}$ dérivée entière de $f(t)$ lorsque $t = 0$. Dans ce cas, les conditions initiales s'expriment en fonction des valeurs en 0 des dérivées entières $D^k f(t)$ de $f(t)$, ($k = 0, \dots, r - 1$). Ceci est un avantage car les conditions initiales sont exprimées par la dérivée entière de la fonction $f(t)$ qui possède un sens physique.

A titre d'illustration supposons qu'on voudrait calculer la dérivée à l'ordre $\alpha = 3,7$ d'une fonction $f(t)$ selon la définition de Caputo, les étapes sont :

- 1) Calcul de l'entier r à partir de (I.14), $r = 4$.
- 2) Dérivée la fonction à l'ordre $r = 4$.

$$g(t) = D^4 f(t)$$
- 3) Intégrer la fonction $g(t)$ à l'ordre $r - \alpha = 0.3$

$$H(t) = I^{0.3} g(t)$$

La définition de Riemann Liouville et Caputo sont équivalentes pour des fonctions f telle que $f(0) = 0$.

Sinon on a :

$${}^R D^\alpha f(t) I^{r-\alpha} f(t) = I^{r-\alpha} {}^C D^\alpha f(t) f(t) + f(0^+) y_{r-\alpha}(t) \quad (\text{I.16})$$

Avec :

$$Y_{r-\alpha}(t) = \frac{(t)^{r-\alpha-1}}{\Gamma(r-\alpha)} \quad (\text{I.17})$$

Remarque :

Si f est une fonction causale constante, la dérivée de l'ordre α au sens de Caputo est nulle mais ce n'est pas le cas au sens de Riemann Liouville, si $f(0^+) \neq 0$ car celle-ci fait intervenir une fonction $Y_{r-\alpha}(t)$ qui tend vers l'infini en $t = 0$.

Par contre pour f à dérivée continue au voisinage de 0 la dérivée au sens de Caputo d'ordre strictement inférieur à 1 est continue en 0 et tend vers 0 en $t = 0$.

Dans la suite de ce mémoire, nous considérons des systèmes relaxés avec des conditions initiales nulles, et la transformée de Laplace des deux définitions de Riemann-Liouville et Caputo se réduit indifféremment à :

$$\mathcal{L}\{D^\alpha f(t)\} = s^\alpha F(s) \quad (\text{I.18})$$

III.3. Définition de Grünwald Letnikov [29, 27]

Nous rappelons la définition classique de la dérivée par (I.19)

h : Taux des accroissements

$$Df(t) = \frac{df(t)}{dx} = \lim_{h \rightarrow 0} \frac{f(t) - f(t-h)}{h} \quad (\text{I.19})$$

On déduit la dérivée seconde :

$$D^2 f(t) = \lim_{h \rightarrow 0} \frac{f(t) - 2f(t-h) + f(t-2h)}{h^2} \quad (\text{I.20})$$

Ainsi la troisième dérivée :

$$D^3 f(t) = \lim_{h \rightarrow 0} \frac{f(t) - 3f(t-h) + 3f(t-2h) - f(t-3h)}{h^3} \quad (\text{I.21})$$

Et plus généralement en élevant à la puissance n l'équation (I.21) et en utilisant la formule de Newton :

$$D^n f(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{j=0}^n \frac{(-1)^j n(n-1)+\dots+(n-j+1)}{j!} f(t - jh) \quad (\text{I.22})$$

L'équation (I.22) peut s'écrire sous la forme condensée:

$$f^{(n)}(t) = \frac{d^n f(t)}{dt^n} = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{j=0}^n (-1)^j \binom{n}{j} f(t - jh) \quad (\text{I.23})$$

Avec
$$\binom{n}{j} = \frac{n(n-1)\cdots(n-j+1)}{j!} = \frac{n!}{j!(n-j)!} \quad (\text{I.24})$$

La notation $\binom{n}{j}$ ou n étant un nombre entier, représente la combinaison de j élément parmi n .

Avec h infiniment petit :

$$f^{(n)}(t) = \frac{d^n f(t)}{dt^n} = \frac{1}{h^n} \sum_{j=0}^n (-1)^j \binom{n}{j} f(t - jh) \quad (\text{I.25})$$

L'extension de l'équation (I.23), à des valeurs non entières $\alpha \in \mathbb{R}$ étant immédiat :

$$D^\alpha f(t) = \frac{1}{h^\alpha} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (\text{I.26})$$

La notation $\binom{\alpha}{j}$ désigne le binôme de Newton généralisé à des ordres réels :

$$\binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} \quad (\text{I.27})$$

Où
$$\Gamma(j+1) = j! \quad (\text{I.28})$$

On obtient alors la formule de Grünwald-Letnikov :

$$D^\alpha f(t) = \frac{1}{h^\alpha} \sum_{j=0}^{\infty} (-1)^j \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} f(t - jh) \quad (\text{I.29})$$

Si la fonction $f(t)$ est nulle pour $t < 0$, ($f(t)$ causale) avec $K = t/h$ alors la formule (I.29) s'écrit :

$$D^\alpha f(Kh) = \frac{1}{h^\alpha} \sum_{j=0}^K (-1)^j \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} f((K-j)h) \quad (\text{I.30})$$

Pour les ordres de dérivation entiers $\alpha = n \in \mathbb{N}$, la somme de l'équation (I.29) est limitée à $(n + 1)$ termes. La valeur de la dérivée à un instant t est alors une combinaison linéaire des $(n + 1)$ valeurs de la fonction $f(t - jh), j = 0, \dots, n$. La dérivation entière donne ainsi une caractérisation locale de la fonction. Par contre, pour des ordres de dérivation non entiers, les coefficients de pondération $\left[(-1)^j \binom{\alpha}{j}\right]$ ne s'annulent pas. La valeur de la dérivée à un instant donné K est alors une combinaison linéaire de toutes les valeurs de la fonction $f(t - jh), j = 0, \dots, K$. Cela montre qu'à l'inverse de la dérivation entière, la dérivation non entière donne une caractérisation globale de la fonction.

III.4. Propriétés [23,36]

- 1) La dérivée d'ordre fractionnaire $D_{t_0}^\alpha f(t)$ d'une fonction temporelle $f(t)$ analytique est aussi analytique.
- 2) La dérivée non entière de l'intégrale du même ordre d'une fonction temporelle est telle que :

$$D_{t_0}^\alpha \circ I_{t_0}^\alpha f(t) = f(t) \quad \text{Avec } \alpha \in \mathbb{R}_+^* \quad (\text{I.31})$$

- 3) La dérivée d'ordre fractionnaire est linéaire. En effet si a et b sont deux constantes alors :

$$D_{t_0}^\alpha [af(t) + bg(t)] = aD_{t_0}^\alpha f(t) + bD_{t_0}^\alpha g(t) \quad (\text{I.32})$$

- 4) La dérivée d'ordre fractionnaire est commutative et vérifie la propriété de semi groupe.

$$D_{t_0}^\beta \left(D_{t_0}^\alpha f(t) \right) = D_{t_0}^\alpha \left(D_{t_0}^\beta f(t) \right) = D_{t_0}^{\alpha+\beta} f(t) \quad (\text{I.33})$$

IV. Utilisation des opérateurs d'intégration et de dérivation non entière

L'intérêt d'étude du calcul fractionnaire est motivé par les applications nombreuses de la dérivation non entière dans divers domaines des sciences et de la technologie. La modélisation de certains phénomènes physiques caractérisés par la propriété de longue mémoire et la structure de dimension infinie peut s'effectuer par l'introduction de la dérivée fractionnaire dans les équations de la dynamique de ces processus. Ce sont les phénomènes simples à géométrie fractale qui ont permis de pointer l'importance de ce type d'équation et de donner une interprétation analytique en termes de théorie des distributions. Cette analyse a largement été confirmée par l'expérience en dépit de réserves toujours vivantes.

Les applications du calcul fractionnaire aux phénomènes de diffusion ou de dispersion sont de loin les plus nombreuses :

- Ligne de transmission (télécommunication)
- Diffusion de la chaleur,
- Diffusion acoustique,
- Diffusion de matières par eau dans le sol,
- Diffusion électrochimique,
- Diffusion électromagnétique,

-La viscoélasticité est un autre domaine où l'application de la dérivation non entière est très répandue.

Depuis ces découvertes beaucoup de travaux de recherche ont montré l'importance du calcul fractionnaire pour l'analyse des systèmes et encouragé l'extension de ses applications à des domaines aussi variés (L'économie et finance, Biomédical, Robotique, et l'Electricité...).

En automatique, l'idée d'utiliser des régulateurs d'ordre fractionnaire pour la commande des systèmes dynamiques revient à Oustaloup au début des années 1990, avec le célèbre régulateur CRONE (Commande Robuste d'Ordre Non Entier) et plusieurs variantes ont vu le jour depuis (1^{ère}, 2^{ème}, 3^{ème} génération).

IV.1. Exemple de modélisation du phénomène de diffusion de chaleur par un modèle non entier [1, 34, 6]

L'étude du lien entre l'opérateur de dérivation fractionnaire et les phénomènes de diffusion a déjà fait l'objet de nombreuses recherches, les auteurs ont établi analytiquement, le passage de l'équation de diffusion à une équation différentielle non entière dont les ordres de dérivation sont multiples de 0.5. Cela a motivé l'utilisation des systèmes non entiers pour la modélisation de ces derniers.

- Présentation

Soit un transfert de chaleur dans des milieux semi-infinis qui sont considérés homogènes de conductivité thermique λ_c , de diffusivité α_c , de température initiale nulle en tout point. Le milieu est soumis à une densité du flux $\phi(t)$ sur la surface.

Le transfert de chaleur est décrit par le système d'équations aux dérivées partielles :

$$\frac{\partial T(r,t)}{\partial t} = \frac{\alpha_c}{r^p} \frac{\partial}{\partial r} \left(r^p \frac{\partial T(r,t)}{\partial r} \right) , \quad 0 < r < \infty , t > 0 \quad (\text{I.34})$$

$$\alpha_c = \frac{\lambda_c}{\rho c_p} \quad (\text{I.35})$$

Où :

r : Abscisse du point de mesure,

c_p : Chaleur massique du milieu,

ρ : Densité massique du milieu,

p : la variable caractérisant le type la géométrie considéré, à savoir :

$p = 0$ pour une géométrie plane ;

$p = 1$ pour une géométrie cylindrique ;

$p = 2$ pour une géométrie sphérique.

Considérons figure (I.1) la configuration simple d'un mur plan homogène de longueur L soumis à l'une des extrémités A à un flux de chaleur $\varphi(0, t)$ (en l'abscisse $x = 0$), la face

B , à l'autre extrémité, transmet la chaleur au milieu ambiant à travers une résistance thermique R :

$\phi(L, t) = \frac{T(L, t)}{R}$; la température $T(x, t)$ du mur est supposée uniforme en chaque point du plan parallèle aux faces A et B de la paroi.

La modélisation théorique consiste à exprimer la relation entre la température T en un point de la barre et le flux de chaleur considéré respectivement comme sortie $y(t)$ et entrée $u(t)$ du système thermique.

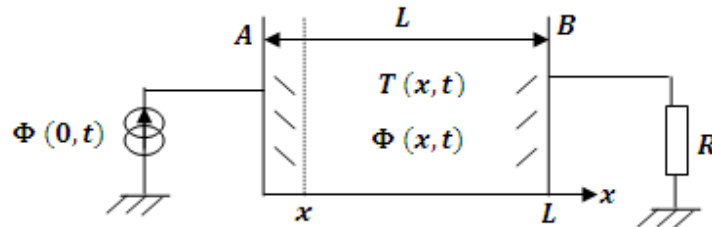


Figure I.1 : Diffusion de chaleur dans un mur

L'étude d'un milieu semi-infini plan correspond au cas $p=0$ et $r=x$, x désignant l'abscisse du point de mesure de température à l'intérieur du milieu. Le système (I.34) se simplifie alors conformément à :

$$\left\{ \begin{array}{l} \frac{\partial T(x, t)}{\partial t} = \alpha_c \frac{\partial^2 T(x, t)}{\partial x^2} \quad , \quad 0 < x < \infty \quad , \quad t > 0 \\ -\lambda_c S_m \frac{\partial T(x, t)}{\partial x} = \phi(x, t) \quad , \quad x = 0 \quad , \quad t > 0 \end{array} \right. \quad \begin{array}{l} \text{(I.36)} \\ \text{(I.37)} \end{array}$$

Condition aux limites

$$\phi(0, t) = u(t) \quad \text{(I.38)}$$

$$\phi(L, t) = \frac{T(L, t)}{R} \quad \text{(I.39)}$$

$$T(x, 0) = 0 \quad \text{(I.40)}$$

La transformée de Laplace des équations (I.36) et (I.37) sont respectivement données par :

$$s\bar{T}(x, s) - T(x, 0) = \alpha_c \frac{\partial^2 \bar{T}(x, s)}{\partial x^2}, \quad \bar{T}(x, s) = \mathcal{L}\{T(x, t)\} \quad (\text{I.41})$$

$$\bar{\phi}(x, s) = -\lambda_c S_m \frac{\partial \bar{T}(x, s)}{\partial x} \quad (\text{I.42})$$

La solution générale de l'équation différentielle (I.41) par rapport à la variable x est de la forme :

$$\bar{T}(x, s) = P(s)e^{\left(x\sqrt{\frac{s}{\alpha_c}}\right)} + Q(s)e^{\left(-x\sqrt{\frac{s}{\alpha_c}}\right)} \quad (\text{I.43})$$

$P(s)$ et $Q(s)$ dépendent des conditions aux limites.

En remplaçant (I.43) dans (I.42) on aura :

$$\bar{\phi}(x, s) = -\lambda_c S_m \sqrt{\frac{s}{\alpha_c}} \left[P(s)e^{\left(x\sqrt{\frac{s}{\alpha_c}}\right)} - Q(s)e^{\left(-x\sqrt{\frac{s}{\alpha_c}}\right)} \right] \quad (\text{I.44})$$

En $x = 0$ on a:

$$\bar{\phi}(0, s) = U(s) = -\lambda_c S_m \sqrt{\frac{s}{\alpha_c}} [P(s) - Q(s)] \quad (\text{I.45})$$

En $x = L$ on a:

$$T(L, t) = 0 \Rightarrow \bar{T}(x, s) = 0 \quad (\text{I.46})$$

Donc:

$$P(s) = -Q(s)e^{-2L\sqrt{\frac{s}{\alpha_c}}} \quad (\text{I.47})$$

L'équation (I.47) permet d'écrire :

$$Q(s) = \frac{\bar{\phi}(x,s)}{-\lambda_c S_m \sqrt{\frac{s}{\alpha_c}} \left(1 + e^{-2L\sqrt{\frac{s}{\alpha_c}}}\right)} \quad (\text{I.48})$$

En remplaçant $P(s)$ et $Q(s)$ par leurs expressions dans la sortie (I.43) :

$$\bar{T}(x,s) = \frac{\bar{\phi}(x,s) \left[-e^{(x-2L)\sqrt{\frac{s}{\alpha_c}}} + e^{-x\sqrt{\frac{s}{\alpha_c}}} \right]}{-\lambda_c S_m \sqrt{\frac{s}{\alpha_c}} \left(1 + e^{-2L\sqrt{\frac{s}{\alpha_c}}}\right)} \quad (\text{I.49})$$

D'où la fonction de transfert :

$$H(s) = \frac{\bar{T}(x,s)}{\bar{\phi}(x,s)} = \frac{-e^{(x-2L)\sqrt{\frac{s}{\alpha_c}}} + e^{-x\sqrt{\frac{s}{\alpha_c}}}}{-\lambda_c S_m \sqrt{\frac{s}{\alpha_c}} \left(1 + e^{-2L\sqrt{\frac{s}{\alpha_c}}}\right)} \quad (\text{I.50})$$

$$H(x,s) = \frac{\sinh\left((L-x)\sqrt{\frac{s}{\alpha_c}}\right)}{S_m \lambda_c \sqrt{\frac{s}{\alpha_c}} \cosh\left(L\sqrt{\frac{s}{\alpha_c}}\right)} \quad (\text{I.51})$$

La résolution de l'équation de chaleur pour ces différentes conditions aux limites a montré que sa solution est une fonction irrationnelle en \sqrt{s} .

V. Systèmes d'ordre fractionnaire

Comme dans le cas des systèmes entier, plusieurs formes de représentation [22, 6, 35] sont faites pour décrire un système non entier (équation différentielle, fonction de transfert, et représentation d'état),

V.1. Equation différentielle linéaire d'ordre non entier

Dans le cadre de la description temporelle du comportement dynamique d'un système fractionnaire linéaire monovariante, une équation différentielle [22, 35] d'ordre non entier est établie sous la forme :

$$a_0 D^{\alpha_0} y(t) + a_1 D^{\alpha_1} y(t) + \dots + a_L D^{\alpha_L} y(t) = b_0 D^{\beta_0} u(t) + \dots + b_M D^{\beta_M} u(t) \quad (I.52)$$

Où $a_i \in \mathbb{R}$ et $b_j \in \mathbb{R}$ désignent respectivement les coefficients de l'entrée $u(t)$ et de la sortie $y(t)$ du système.

Les ordres de dérivation. $[\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_L, \beta_0, \beta_1, \beta_2, \dots, \beta_M]$ étant des nombres entiers, non entiers, réels.

D^α Désigne l'opérateur de dérivation d'ordre α .

L'expression généralisée de (I.52) est donnée par:

$$a_0 y(t) + \sum_{i=1}^L a_i D^{\alpha_i} y(t) = \sum_{j=1}^M b_j D^{\beta_j} u(t) + b_0 u(t) \quad (I.53)$$

Une équation différentielle d'ordre non entier est dite d'ordre commensurable α lorsque tous les ordres α_i, β_i sont multiples du même nombre non entier α , c'est-à-dire :

$$\begin{cases} \alpha_i = n_i \times \alpha & i = 0, 1, \dots, L \\ \beta_j = m_j \times \alpha & j = 0, 1, \dots, M \end{cases} \quad 0 < \alpha < 1 \quad (I.54)$$

Dans ce cas l'équation différentielle généralisée (I.53) devient :

$$a_0 y(t) + \sum_{i=1}^L a_i D^{n_i \alpha} y(t) = \sum_{j=1}^M b_j D^{m_j \alpha} u(t) + b_0 u(t) \quad (I.55)$$

V.2. Fonction de transfert d'ordre non entier

L'application de la transformée de Laplace à l'équation différentielle linéaire d'ordre non entier (I.52), en considérant les conditions initiales nulles, permet de déduire la fonction de transfert non entière correspondante :

$$H(s) = \frac{Y(s)}{U(s)} = \frac{\sum_{i=0}^M b_j s^{\beta_j}}{\sum_{j=0}^L a_i s^{\alpha_i}} \quad (\text{I.56})$$

Lorsque les ordres de dérivation α_i et β_j sont quelconques le système est appelé système fractionnaire généralisé, lorsqu'ils sont multiples d'un même nombre réel, on parle d'un système d'ordre commensurable.

La fonction de transfert non entière (I.56) s'écrit :

$$H(s) = \frac{\sum_{i=0}^M b_j s^{m_j \alpha}}{\sum_{j=0}^L a_i s^{n_i \alpha}} \quad , 0 < \alpha < 1 \quad (\text{I.57})$$

Dans le cas particulier où l'ordre α est l'inverse d'un entier naturel λ le système est dit d'ordre rationnel :

$$H(s) = \frac{\sum_{i=0}^M b_j s^{i/\lambda}}{\sum_{j=0}^L a_i s^{j/\lambda}} \quad \lambda \in \mathbb{N}^* \quad (\text{I.58})$$

V.3. Modèle d'état non entier

Comme dans le cas entier, une représentation d'état non entière comporte deux équations :

- Une équation d'état non entière dans laquelle le vecteur d'état ne fait plus l'objet d'une dérivation unitaire mais d'une dérivation d'ordre α entier, non entier, réel.
- Une équation d'observation identique à celle des systèmes entiers.

La représentation d'état est ainsi définie par le système d'équations :

$$\begin{cases} D^{(\alpha)} x(t) = A x(t) + B u(t) \\ y(t) = C x(t) + E u(t) \end{cases} \quad (\text{I.59})$$

Où:

$$D^{(\alpha)}(x) = [D^{\alpha_1}x_1, D^{\alpha_2}x_2 \dots D^{\alpha_n}x_n]^T \quad (\text{I.60})$$

Avec $x \in \mathbb{R}^n, u \in \mathbb{R}^l, y \in \mathbb{R}^q, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times l}, C \in \mathbb{R}^{q \times n}, D \in \mathbb{R}^{q \times l}$

Dans lequel :

u : Vecteur des entrées de dimension $(l \times 1)$;

x : Vecteur d'état non entier de dimension $(n \times 1)$;

y : Vecteur des sorties de dimension $(q \times 1)$;

α : L'ordre de dérivation non entier ;

Les matrices A, B, C et E sont toutes à éléments constants.

Dans le cas des systèmes commensurables le modèle d'état non entier (I.59) devient :

$$\begin{cases} D^\alpha x(t) = A x(t) + B u(t) \\ y(t) = C x(t) + E u(t) \end{cases} \quad (\text{I.61})$$

Où :

$$D^\alpha(x) = D^\alpha[x_1, x_2 \dots x_n]^T \quad (\text{I.62})$$

V.4. Propriétés des systèmes fractionnaires

V.4.1. Stabilité des systèmes fractionnaire

La définition de stabilité au sens BIBO (Bounded Input, Bounded Output), dite aussi stabilité externe, est donnée par la définition suivante :

Définition 1: *Un système est dit BIBO stable si et seulement si, à une entrée bornée il correspond une sortie bornée.*

Dans le cas des systèmes linéaires non entiers d'ordre commensurable, comme dans le cas entier, la condition de stabilité est que l'équation caractéristique du système n'admet aucune racine à partie réelle positive. La condition de stabilité établie par Matignon pour les systèmes non entiers commensurables nécessite que les pôles p_i du système vérifient la condition suivante :

Théoreme [35]

Un système fractionnaire d'ordre commensurable est BIBO stable si et seulement si toutes les racines p_i du polynôme entier correspondant à son polynôme caractéristique ont leurs arguments qui vérifient la condition suivante:

$$|\arg(p_i)| > \frac{\pi}{2\lambda} \quad (I.63)$$

$\arg(p_i)$: est l'argument de la racine p_i .

λ : ordre commensurable.

Le domaine de stabilité est représenté dans la figure (I.3)

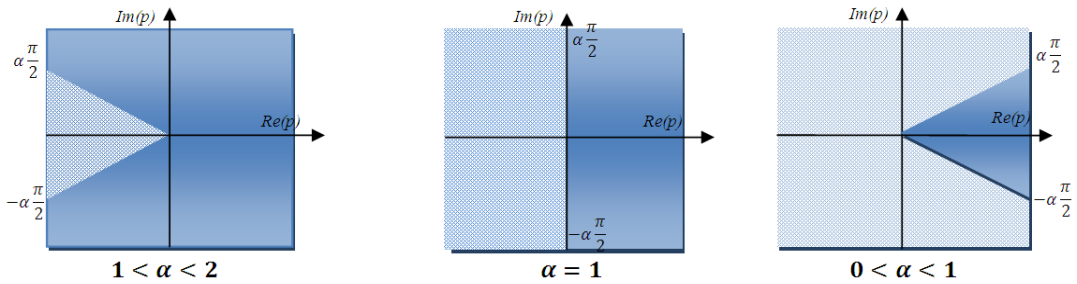


Figure I.2: domaine de stabilité des systèmes commensurables

dans le plan ($p = s^\alpha$) complexe.

V.4.2. Commandabilité et observabilité

La définition de la commandabilité [22, 35, 27] des systèmes non entier est la même que celle de la théorie des systèmes linéaires entiers. Cependant les résultats obtenus de l'étude de la commandabilité et de l'observabilité ne concerne que les systèmes non entiers commensurables.

Définition 2: Le système non entier d'ordre commensurable de l'équation (1.61) est dit commandable si pour un temps donné t_0 il existe un temps fini $t_1 > t_0$ tel que, quelque

soient deux états $x(t_0) = x_0$ et $x(t_1) = x_1$ dans l'espace d'état, il existe une entrée de commande $u(t)$, $t \in [t_0, t_1]$ qui permet de transférer l'état $x(t)$ de x_0 à x_1 en un temps fini t_1 .

La condition de commandabilité est alors la même que pour le cas des systèmes entiers. Le système non entier d'ordre commensurable est commandable si le rang de la matrice de commandabilité égal à n .

$$C = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (I.64)$$

$$\text{rang}(C) = n \quad (I.65)$$

De la même manière la condition de l'observabilité des systèmes d'ordre non entier commensurables est établie en utilisant la définition d'observabilité des systèmes entiers donnée par :

Définition 3: Les systèmes non entiers d'ordre commensurable de l'équation (I.63) est observable pendant l'intervalle $[t_0, t_1]$, $t_1 > 0$, si n'importe quel état $x(t_0)$ peut être déduit à partir des observations de la sortie $y(t)$ et de l'entrée $u(t)$ pendant un temps fini $t \in [t_0, t_1]$. Dans ce cas aussi, la condition d'observabilité du système est que le rang de la matrice d'observabilité égal à n .

$$\theta = \begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} \quad (I.66)$$

$$\text{rang}(\theta) = n \quad (I.67)$$

Ces définitions de commandabilité et d'observabilité calquées sur la théorie des systèmes entiers posent des problèmes dans le cas des systèmes fractionnaires généralisés, du fait de la propriété de mémoire longue des systèmes non entiers. En effet

la dimension n du vecteur d'état n'est pas égale à la dimension du système comme nous le verrons dans la représentation d'état déterminée au deuxième chapitre.

VI. Conclusion

Nous avons donné dans ce chapitre les définitions de base des opérateurs intégrodifférentiels d'ordre non entier du calcul fractionnaire.

Certains phénomènes physiques particulièrement liés aux propriétés de diffusion et de mémoire longue sont décrit au moyen de ces opérateurs.

Nous avons donc les différents types de représentation des systèmes d'ordre fractionnaire et leurs propriétés.

Chapitre II :

Développement des algorithmes

I. Introduction

L'objectif de ce chapitre est le développement des algorithmes de calcul fractionnaire, d'analyse et de simulation des systèmes d'ordre fractionnaire.

Les problèmes traités sont :

- Calcul de la dérivée et de l'intégrale d'ordre fractionnaire
- Résolution numérique et solution d'une équation différentielle ordinaire à coefficients constants d'ordre non entier
- Calcul des racines d'un polynôme à puissances non entières
- Détermination des approximations des systèmes d'ordre fractionnaire par les modèles d'ordre entier ; ces approximations sont nécessaires pour la réalisation physique ou la simulation car les composants physiques et les outils de calcul ne peuvent pas le faire actuellement.

II. Calcul de la dérivée et de l'intégrale non entière [27]

La définition de Grünwald-Letnikov permet le calcul numérique de l'intégration/dérivation à un ordre α réel d'une fonction causale $f(t)$. Les deux fonctionnalités étant rassemblées dans un seul module dit "dérivateur généralisé". Il s'agit alors d'intégration lorsque $\alpha < 0$ et de dérivation lorsque $\alpha > 0$.

Nous rappelons la formule de Grünwald-Letnikov donnée par :

$$D^\alpha f(t) = \frac{1}{h^\alpha} \sum_{j=0}^{\infty} (-1)^j \binom{\alpha}{j} f(t - jh) \quad (\text{II.1})$$

En discrétisant le temps à un pas d'échantillonnage h , soit $t = Kh$, en tenant compte de la causalité la formule devient :

$$D^\alpha f(Kh) = \frac{1}{h^\alpha} \sum_{j=0}^K (-1)^j \binom{\alpha}{j} f((K - j)h) \quad (\text{II.2})$$

On pose :

$$c(j) = \frac{1}{h^\alpha} (-1)^j \binom{\alpha}{j} \quad (\text{II.3})$$

En recourant à la loi de récurrence entre les coefficients $c(j)$ et $c(j-1)$ on aboutit à :

$$\begin{cases} c(0) = \frac{1}{h^\alpha} \\ c(j) = c(j-1) \frac{j-\alpha-1}{j} \end{cases} \quad j = 1, \dots, K \quad (\text{II.4})$$

Alors (II.1) peut être écrite sous la forme :

$$D^\alpha f(Kh) = \sum_{j=0}^K c(j) f((K-j)h) \quad (\text{II.5})$$

L'algorithme est donné dans la table (II.1) :

<p>* Entrée des données:</p> <ul style="list-style-type: none"> - Expression analytique de la fonction $f(t)$, ordre non entier α, - Le temps initial $t_0 = 0$, le temps final t_f, et le pas d'échantillonnage h, <p>* Calculer la dérivée $D^\alpha f(Kh)$ notée s.</p> <p>Faire pour $K = 1, \dots, N$ ($N = \frac{t_f - t_0}{h}$)</p> <ol style="list-style-type: none"> 1. Calculer $c(0) = \frac{1}{h^\alpha}$ 2. Faire pour $j = 1$ jusqu'à K $c(j) = c(j-1) \frac{j-\alpha-1}{j}$ 3. Initialiser $s = 0$ 4. Faire pour $j = 0$ jusqu'à K $s = s + c(j) f((K-j)h)$ <p>$D^\alpha f(Kh) = s$</p>

Table II. 1 : Algorithme de calcul de la dérivée et l'intégrale non entière

La relation (II.5) montre, qu'à chaque itération, le calcul de la dérivée généralisée se fait non seulement au moyen de la valeur présente de la fonction mais aussi de toutes ses valeurs passées. Cela signifie que la masse de calcul et la capacité mémoire requise augmente infiniment avec le temps, ce qui est impossible dans la réalité. La solution proposée c'est de retenir les échantillons récents de la fonction selon une longueur de mémoire spécifiée, c'est le principe de la mémoire limitée (Short Memory Principal), illustré par l'équation suivante :

$$D^\alpha f(Kh) = \sum_{j=0}^{L_f} c(j) f((K-j)h) \quad (\text{II.6})$$

L_f : Longueur de la mémoire courte choisie.

III. Résolution d'une équation différentielle linéaire d'ordre non entier [31, 7]

Un système monovarié d'ordre fractionnaire d'entrée $u(t)$ et de sortie $y(t)$ à temps invariant est décrit par une équation différentielle linéaire d'ordre non entier à coefficients constants. Pour la résolution de cette équation différentielle nous utilisons la définition de la dérivée non entière de Grünwald-Letnikov qui se prête naturellement au développement algorithmique. Considérons l'équation différentielle linéaire généralisée d'ordre non entier suivante :

$$\sum_{i=0}^L a_i D^{\alpha_i} y(t) = \sum_{k=0}^M b_k D^{\beta_k} u(t) \quad (\text{II.7})$$

La résolution de l'équation différentielle linéaire d'ordre non entier décrite par (II.7) est basée sur la discrétisation à un pas d'échantillonnage $h > 0$ de cette équation en utilisant la définition de Grünwald-Letnikov (II.1), pour $t = Kh$.

(K étant un nombre entier).

En remplaçant $D^{\alpha_i} y(Kh)$ et $D^{\beta_k} u(Kh)$ par la définition de Grünwald-Letnikov donnée par :

$$D^\alpha f(Kh) = \frac{1}{h^\alpha} \sum_{j=0}^K \binom{\alpha}{j} (-1)^j f((K-j)h) \quad (\text{II.8})$$

On obtient :

$$\sum_{i=0}^L a_i \frac{1}{h^{\alpha_i}} \sum_{j=0}^K \left((-1)^j \binom{\alpha_i}{j} y((K-j)h) \right) = \sum_{k=0}^M b_k \frac{1}{h^{\beta_k}} \sum_{j=0}^K \left((-1)^j \binom{\beta_k}{j} u((K-j)h) \right) \quad (\text{II.9})$$

On extrait l'échantillon $y(Kh)$, on obtient :

$$\sum_{i=0}^L a_i \frac{1}{h^{\alpha_i}} \left[y(Kh) + \sum_{j=1}^K \left((-1)^j \binom{\alpha_i}{j} y((K-j)h) \right) \right] = \sum_{k=0}^M b_k \frac{1}{h^{\beta_k}} \sum_{j=0}^K \left((-1)^j \binom{\beta_k}{j} u((K-j)h) \right) \quad (\text{II.10})$$

L'équation de récurrence donnant $y(Kh)$ s'écrit donc :

$$y(Kh) = - \frac{\sum_{i=0}^L \frac{a_i}{h^{\alpha_i}} \sum_{j=1}^K \left((-1)^j \binom{\alpha_i}{j} y((K-j)h) \right)}{\sum_{i=0}^L \frac{a_i}{h^{\alpha_i}}} + \frac{\sum_{k=0}^M \frac{b_k}{h^{\beta_k}} \sum_{j=0}^K \left((-1)^j \binom{\beta_k}{j} u((K-j)h) \right)}{\sum_{i=0}^L \frac{a_i}{h^{\alpha_i}}} \quad (\text{II.11})$$

On pose:

$$\begin{cases} Y_L(Kh) = \sum_{j=1}^K \left((-1)^j \binom{\alpha_i}{j} y((K-j)h) \right) \\ U_M(Kh) = \sum_{j=0}^K \left((-1)^j \binom{\beta_k}{j} u((K-j)h) \right) \end{cases} \quad (\text{II.12})$$

Et

$$\begin{cases} a_l = \frac{\frac{a_i}{h^{\alpha_i}}}{\sum_{i=0}^L \frac{a_i}{h^{\alpha_i}}} \\ b_m = \frac{\frac{b_k}{h^{\beta_k}}}{\sum_{i=0}^L \frac{a_i}{h^{\alpha_i}}} \end{cases} \quad (\text{II.13})$$

La solution de l'équation différentielle à chaque pas Kh , est obtenue donc par l'équation de récurrence suivante :

$$y(Kh) = - \sum_{l=0}^L a_l Y_L(Kh) + \sum_{k=0}^M b_m U_M(Kh) \quad (\text{II.14})$$

L'algorithme de calcul numérique de la solution de l'équation différentielle est dans la table ci-dessous :

*Entrée des données:

- Le vecteur des ordres non entiers α_i correspondant a la sortie $y(t)$.
- Le vecteur des ordres non entiers β_k correspondant a l'entrée $u(t)$.
- Le vecteur des coefficients de sortie a_i , le vecteur des coefficients d'entrée b_k .
- L'expression analytique de l'entrée $u(t)$. Le temps initial $t_0 = 0$, le temps final t_f et le pas d'échantillonnage h .

*Calculer la sortie $y(Kh)$

Faire pour $K = 0, \dots, N$ $\left(N = \frac{t_f - t_0}{h} \right)$

1) *Initialiser* $Y_L(K) = 0$

Faire pour $j = 1, 2, \dots, K$

$$Y_L(K) = Y_L(K) + \left((-1)^j \binom{\alpha_i}{j} y((K-j)h) \right)$$

2) *Initialiser* $S = 0$

Faire pour $i = 0, 1, 2, \dots, L$

$$S(K) = S(K) + \frac{a_i}{h^{\alpha_i}} Y_L(K)$$

3) *Initialiser* $U_M = 0$

Faire pour $j = 0, 1, \dots, K$

$$U_M(K) = U_M(K) + \left((-1)^j \binom{\beta_k}{j} u((K-j)h) \right)$$

<p>4) Initialiser $S1 = 0$ Faire pour $k = 0, 1, 2, \dots, M$</p> $S1(K) = S1(K) + \frac{b_k}{h^{\beta_k}} U_M(K)$ <p>5) Initialiser $s = 0$ Faire pour $i = 0, 1, \dots, L$ $s = s + \frac{a_i}{h^{\alpha_i}}$</p> <p>6) $y(K) = \frac{-S(K) + S1(K)}{s}$</p> <p>$y = y(K)$</p>
--

Table II.2 Algorithme de résolution d'une équation différentielle d'ordre non entier

IV. Résolution d'une équation polynômiale à puissance non entière [22]

Soit à résoudre une équation à puissances non entières réelles suivante :

$$\Delta_{ne}(s) = a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \dots + a_1 s^{\alpha_1} + a_0 = 0 \quad (\text{II.15})$$

Avec : $a_i \in \mathbb{R}$, $\alpha_i \in \mathbb{R}_+$, $i = (1, 2, \dots, n)$.

La variable s peut s'exprimer par :

$$s = |s| e^{j\varphi} \quad , \quad \text{avec } \varphi = \varphi_0 + 2k\pi \quad (\text{II.16})$$

La puissance quelconque de s s'écrit comme suit :

$$s^{\alpha_i} = (|s| e^{j(\varphi_0 + 2k\pi)})^{\alpha_i} = |s|^{\alpha_i} e^{j\alpha_i \varphi_0} e^{j2\alpha_i k\pi} \quad (\text{II.17})$$

Dans le cas où l'ordre α_i est non entier, le terme $e^{j2\alpha_i k\pi}$ dépend de k , c'est-à-dire que s^{α_i} a plusieurs sens, ceci implique que l'équation (II.15) est multiforme.

Pour rendre l'équation uniforme, il faut éviter que l'argument de s décrive un tour complet, ce qui est possible en effectuant une coupure du plan complexe, suivant l'axe \mathbb{R}^- , c'est-à-dire $s \in \mathbb{C} - \mathbb{R}^-$.

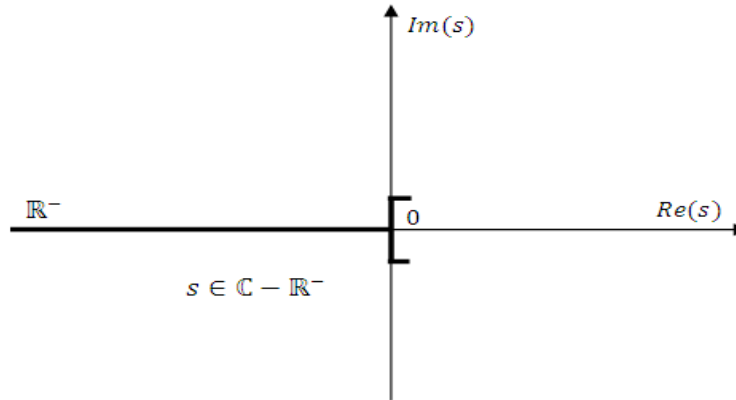


Figure II.1 : Coupure du plan complexe suivant l'axe \mathbb{R}^-

La résolution d'une équation (II.15) se fait en approximant les puissances réelles par des puissances fractionnaires.

La puissance réelle de rang i doit s'écrire donc sous la forme :

$$\alpha_i = \frac{r_i}{q} + e_i \quad (\text{II.18})$$

$$r_i, q \in \mathbb{N}, r_i \geq 1, q \geq 2$$

e_i : erreur de rationalisation de α_i

En remplaçant l'approximation des α_i dans (II.15), on obtient l'équation d'ordre fractionnaire :

$$\Delta_{fr}(s) = a_n s^{r_n/q} + a_{n-1} s^{r_{n-1}/q} + \dots + a_1 s^{r_1/q} + a_0 = 0 \quad (\text{II.19})$$

Posons :

$$p = s^{1/q} \quad (\text{II.20})$$

Nous obtenons l'équation d'ordre entier (II.21) qui peut être résolue en utilisant des outils existants.

$$\Delta_{en}(p) = a_n p^{r_n} + a_{n-1} p^{r_{n-1}} + \dots + a_1 p^{r_1} + a_0 = 0 \quad (\text{II.21})$$

Les racines p_i de (II.21) s'écrivent :

$$p_i = |p_i| e^{\arg(p_i)} \quad (i = 1, \dots, r_n) \quad (\text{II.22})$$

On déduit les racines s_i de l'équation (II.19) :

$$\begin{cases} |s_i| = |p_i|^q \\ \arg(s_i) = q \arg(p_i) + 2qk\pi \end{cases} \quad k = 0, \bar{1}, \bar{2}, \dots \quad (\text{II.23})$$

On vérifie l'existence des racines fractionnaires s_i par la condition :

$$-\frac{1}{2q} - \frac{\arg(p_i)}{2\pi} < k < \frac{1}{2q} - \frac{\arg(p_i)}{2\pi} \quad (\text{II.24})$$

IV.1. Propriétés

- Les racines du polynôme fractionnaire $\Delta_{fr}(s)$, sont les approximations des racines du polynôme non entier $\Delta_{ne}(s)$.

- Pour les racines réelles négatives de $\Delta_{en}(p)$, le polynôme fractionnaire ne lui correspond aucune racine, en raison de la coupure du plan complexe, donc $\Delta_{ne}(s)$ n'a pas de racines.

- Pour une même racine du polynôme $\Delta_{en}(p)$, le polynôme $\Delta_{ne}(s)$ peut avoir plusieurs racines lorsque k a plusieurs valeurs qui vérifient la condition (II.24), ou aucune racine lorsqu'il n'y a aucune valeur de k qui vérifie la même condition.

- Un polynôme non entier, peut avoir un nombre beaucoup plus grand de racines que le polynôme entier qui lui correspond par le changement de variable $p = s^\alpha$, car le nombre des racines de $\Delta_{ne}(s)$ ne peut être déduit ni à partir de la puissance la plus élevée de sa variable, ni à partir du nombre de ses coefficients.

- On ne peut pas reconstruire le polynôme non entier à partir de ses racines comme dans le cas classique.

Remarque :

L'algorithme est donné dans la table (II.3) :

*Entrée des données :

- les ordres non entiers α_i
- les coefficients du polynôme a_i

*Calculer les racines

- 1) Mettre α_i sous la forme : $\alpha_i = \frac{r_i}{q} + e_i$
- 2) Calculer les racines du polynôme entier (II.21)
Faire pour $i = 1, \dots, r_n$
$$p_i = |p_i| e^{\arg(p_i)}$$
- 3) Calculer les racines du polynôme fractionnaire en (s)
Faire pour $k = 0, \mp 1, \mp 2, \dots$

$$\begin{cases} |s_i| = |p_i|^q \\ \arg(s_i) = q \arg(p_i) + 2qk\pi \end{cases}$$

Si la condition est vérifiée

$$-\frac{1}{2q} - \frac{\arg(p_i)}{2\pi} < k < \frac{1}{2q} - \frac{\arg(p_i)}{2\pi}$$

Alors s_i est une racine de $p_{ne}(s)$

Table II. 3 : Algorithme de résolution d'une équation polynômiale d'ordre non entier

V. Calcul de la réponse temporelle du modèle d'état non entier

Etant donnée un système linéaire monovariante non entier d'ordre commensurable $0 < \alpha < 1$ dont le modèle d'état est donné par :

$$\begin{cases} D^\alpha x(t) = A x(t) + B u(t) \\ y(t) = C x(t) + E u(t) \end{cases} \quad x(0) = x_0, x \in \mathbb{R}^n \quad (\text{II.25})$$

En appliquant la transformée de Laplace à l'expression $D^\alpha x(t)$, en utilisant la définition de Grünwald Letnikov, nous obtenons:

$$X(s) = (s^\alpha I_n - A)^{-1} B U(s) + (s^\alpha I_n - A)x_0 \quad (\text{II.26})$$

I_n : matrice identité.

On peut alors déterminer l'expression temporelle du vecteur d'état $x(t)$ par :

$$x(t) = \mathcal{L}^{-1}[X(s)] = \mathcal{L}^{-1}[(s^\alpha I_n - A)^{-1} B U(s) + (s^\alpha I_n - A)x_0] \quad (\text{II.27})$$

Définissant alors, comme dans le cas entier, la matrice de transition par :

$$\psi(t) = \mathcal{L}^{-1}[(s^\alpha I_n - A)^{-1}], \text{ pour } t > 0 \quad (\text{II.28})$$

On obtient :

$$x(t) = \psi(t)x_0 + \psi(t) * [B u(t)] \quad (\text{II.29})$$

C'est-à-dire :

$$x(t) = \psi(t)x_0 + \int_0^t \psi(t - \tau) Bu(\tau) d\tau \quad (\text{II.30})$$

Où:

$$\psi(t) = E_\alpha(At^\alpha) = \sum_{k=0}^{\infty} \frac{A^k t^{k\alpha}}{\Gamma(1+k\alpha)} \quad (\text{II.31})$$

E_α : est la fonction de Mittag-Leffler qui est la généralisation de la fonction exponentielle. En effet, lorsque ($\alpha = 1$) le développement de la formule (II.31) donne e^{At} .

La réponse du modèle d'état est donnée par :

$$y(t) = C\psi(t)x_0 + \int_0^t C\psi(t - \tau) Bu(\tau) d\tau + Eu(t) \quad (\text{II.32})$$

Le calcul de la réponse du modèle d'état consiste donc à :

- Evaluer la fonction de Mittag- Leffler
- Calculer le produit de convolution $\psi(t) * [Bu(t)]$

Le calcul du produit de convolution est relativement complexe pour une matrice d'état A quelconque. Ceci reste à faire.

VI. Passages entre les différentes représentations d'un système fractionnaire

VI.1. Equation différentielle d'ordre non entier vers une représentation d'état non entière

Le passage d'une équation différentielle linéaire d'ordre non entier à coefficients constants vers une représentation d'état non entière se fait en deux étapes :

- Passage d'une équation différentielle linéaire d'ordre non entier vers une fonction de transfert d'ordre non entier.
- Passage d'une fonction de transfert d'ordre non entier vers une représentation d'état non entière.

VI.1.1. Equation différentielle d'ordre non entier vers une fonction de transfert d'ordre non entier

Soit l'équation différentielle linéaire d'ordre non entier à coefficients constants définie par :

$$\begin{aligned} a_0 D^{\alpha_0} y(t) + a_1 D^{\alpha_1} y(t) + \dots + a_L D^{\alpha_L} y(t) \\ = b_0 D^{\beta_0} u(t) + b_1 D^{\beta_1} u(t) + \dots + b_M D^{\beta_M} u(t) \end{aligned} \quad (\text{II.33})$$

Le passage de l'équation différentielle linéaire d'ordre non entier à coefficients constants vers la fonction de transfert d'ordre non entier se fait comme le cas d'un système entier. Considérons les conditions initiales nulles, on applique la définition de la transformée de Laplace sur la dérivation non entière $D^\alpha f(t)$.

$$\mathcal{L}[D^\alpha f(t)] = s^\alpha F(s) \quad (\text{II.34})$$

On aura :

$$\begin{aligned} a_0 s^{\alpha_0} Y(s) + a_1 s^{\alpha_1} Y(s) + \dots + a_L s^{\alpha_L} Y(s) = \\ b_0 s^{\beta_0} U(s) + b_1 s^{\beta_1} U(s) + \dots + b_M s^{\beta_M} U(s) \end{aligned} \quad (\text{II.35})$$

D'où la fonction de transfert

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^{\beta_0} + b_1 s^{\beta_1} + b_2 s^{\beta_2} + \dots + b_M s^{\beta_M}}{a_0 s^{\alpha_0} + a_1 s^{\alpha_1} + a_2 s^{\alpha_2} + \dots + a_L s^{\alpha_L}} \quad (\text{II.36})$$

Dans le cas des systèmes commensurables d'ordre α , la fonction $G(s)$ s'écrit :

$$G(s) = \frac{b_m s^{(m)\alpha} + b_1 s^{(m-1)\alpha} + b_2 s^{(m-2)\alpha} + \dots + b_0}{s^{n\alpha} + a_1 s^{(n-1)\alpha} + a_2 s^{(n-2)\alpha} + \dots + a_n} \quad (\text{II.37})$$

VI.1.2. Fonction de transfert non entière vers une représentation d'état non entière

On distingue trois cas de passage d'une fonction de transfert non entière vers une représentation d'état non entière

- Cas des systèmes commensurable.
- Cas des systèmes non commensurables ou le numérateur est constant.
- Cas des systèmes non commensurables ou le numérateur est un polynôme.

a) Cas commensurable

Disposons d'une fonction de transfert $G(s)$ d'un système fractionnaire monovariante linéaire invariant donnée par l'expression :

$$G(s) = \frac{b_m s^{(m)\alpha} + b_1 s^{(m-1)\alpha} + b_2 s^{(m-2)\alpha} + \dots + b_0}{s^{n\alpha} + a_1 s^{(n-1)\alpha} + a_2 s^{(n-2)\alpha} + \dots + a_n} \quad (\text{II.38})$$

Effectuons le changement de variable $p = s^\alpha$, $G(s)$ s'écrit :

$$G(p) = \frac{b_m p^m + b_1 p^{m-1} + b_2 p^{m-2} + \dots + b_0}{p^n + a_1 p^{n-1} + a_2 p^{n-2} + \dots + a_n} \quad (\text{II.39})$$

A la fonction de transfert $G(p)$, on peut associer le modèle d'état suivant la forme canonique désirée (forme commandable, forme observable, Jordan,..).

$$\begin{cases} \dot{x}(t) = A x(t) + B u(t) \\ y(t) = C x(t) + E u(t) \end{cases} \quad (\text{II.40})$$

En remplaçant la dérivée de premier ordre (\dot{x}) par la dérivée d'ordre α ($D^\alpha(x)$), on obtient le modèle d'état non entier commensurable :

$$\begin{cases} D^\alpha x(t) = A x(t) + B u(t) \\ y(t) = C x(t) + E u(t) \end{cases} \quad (\text{II.41})$$

b) Cas où $G(s)$ admet un numérateur constant

La fonction de transfert $G(s)$ s'écrit sous la forme :

$$G(s) = \frac{b_0}{s^{\alpha_n + a_1} s^{\alpha_{n-1} + \dots + a_{n-1}} s^{\alpha_1 + a_n}} \quad (\text{II.42})$$

En supposant que : $\alpha_n > \alpha_{(n-1)} > \dots > \alpha_2 > \alpha_1$

Pour calculer le modèle d'état de $G(s)$, nous utilisons la méthode classique d'obtention d'une représentation d'état de la forme canonique commandable pour les systèmes entiers.

L'équation différentielle linéaire d'ordre non entier associée à $G(s)$ est :

$$D^{\alpha_n} y(t) = -a_1 D^{\alpha_{n-1}} y(t) - \dots - a_{n-1} D^{\alpha_1} y(t) - a_n y(t) + b_0 u(t) \quad (\text{II.43})$$

Considérons alors le vecteur d'état : (dans ce qui suit, on omet La variable t pour ne pas surcharger les expressions)

$$\begin{cases} x_1 = y \\ x_2 = D^{\alpha_1} x_1 = D^{\alpha_1} y \\ x_3 = D^{\alpha_2 - \alpha_1} x_2 = D^{\alpha_2 - \alpha_1} (D^{\alpha_1} y) = D^{\alpha_2} y \\ \vdots \\ x_i = D^{\alpha_{i-1} - \alpha_{i-2}} x_{i-1} = D^{\alpha_{i-1} - \alpha_{i-2}} (D^{\alpha_{i-2} - \alpha_{i-3}} y) = \dots = D^{\alpha_{i-1}} y \\ \vdots \\ x_n = D^{\alpha_{n-1} - \alpha_{n-2}} x_{n-1} = D^{\alpha_{n-1}} y \end{cases} \quad (\text{II.44})$$

$$D^{(\alpha)}(x) = \begin{cases} D^{\alpha_1} x_1 = x_2 \\ D^{\alpha_2 - \alpha_1} x_2 = x_3 \\ D^{\alpha_3 - \alpha_2} x_3 = x_4 \\ \vdots \\ D^{\alpha_i - \alpha_{i-1}} x_i = x_{i+1} \\ \vdots \\ D^{\alpha_n - \alpha_{n-1}} x_n = D^{\alpha_n - \alpha_{n-1}} (D^{\alpha_{n-1} - \alpha_{n-2}} x_{n-1}) = \dots = D^{\alpha_n} y \end{cases} \quad (\text{II.45})$$

La dernière composante du vecteur $D^{(\alpha)}(x)$ ($D^{\alpha_n} y$) s'écrit en fonction des autres dérivées de $y(t)$ selon l'équation (II.43). On peut alors l'exprimer en fonction des différentes composantes du vecteur d'état $x(t)$ par :

$$D^{\alpha_n - \alpha_{n-1}} x_n = -a_1 x_n - a_2 x_{n-1} - a_{n-1} x_2 - \dots - a_n x_1 + b_0 u \quad (\text{II.46})$$

Le modèle d'état non entier correspondant au modèle transfert (II.42) non entier est donné donc par (II.45) :

$$\begin{cases} D^{(\alpha)}(x) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_2 & -a_1 \end{bmatrix} (x) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \\ y = [b_0 \ 0 \ 0 \ \dots \ 0 \ 0](x) \end{cases} \quad (\text{II.47})$$

$$\text{Avec} \begin{cases} x = [x_1 \ x_2 \ \dots \ x_n]^T \\ D^{(\alpha)}(x) = [D^{\alpha_1} x_1 \ D^{\alpha_2 - \alpha_1} x_2 \ \dots \ D^{\alpha_n - \alpha_{n-1}} x_n]^T \end{cases} \quad (\text{II.48})$$

c) Cas ou le numérateur de $G(s)$ est un polynôme [1, 22]

Nous présentons dans ce cas la méthode générale présentée dans [22]. Soit la fonction de transfert d'ordre non entier $G(s)$, supposée propre ($\beta_m < \alpha_n$) :

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \dots + b_1 s^{\beta_1} + b_0}{s^{\alpha_n + a_1} s^{\alpha_{n-1}} + \dots + a_{n-1} s^{\alpha_1} + a_n} \quad (\text{II.49})$$

On suppose que $\alpha_n > \alpha_{(n-1)} > \dots > \alpha_2 > \alpha_1$ et $\beta_m > \beta_{(m-1)} > \dots > \beta_2 > \beta_1$

Dans le cas où tous les ordres de dérivation α_i et β_j sont différents, le nombre de variable du modèle d'état d'un système non entier est égal $n + m$.

La construction du modèle d'état d'ordre non entier sous la forme canonique commandable est décrite par les étapes de l'algorithme ci-dessous :

- 1) Construire le vecteur $\tilde{\alpha}$ par la concaténation des deux vecteurs des ordres non entiers du numérateur et ceux du dénominateur α_i et β_i .

$$\tilde{\alpha} = [\tilde{\alpha}_{n+m} \quad \tilde{\alpha}_{n+m-1} \quad \tilde{\alpha}_{n+m-2} \quad \dots \quad \tilde{\alpha}_2 \quad \tilde{\alpha}_1] \quad (\text{II.50})$$

$$\text{Tel que:} \quad \tilde{\alpha}_{n+m} > \tilde{\alpha}_{n+m-1} > \tilde{\alpha}_{n+m-2} > \dots > \tilde{\alpha}_2 > \tilde{\alpha}_1 \quad (\text{II.51})$$

- 2) Les termes $\tilde{\alpha}_i$ et \tilde{c}_i qui représentent respectivement la $(n + m)^{\text{ème}}$ ligne de la matrice A et le vecteur C du modèle d'état non entier sont déterminés par la procédure de sélection suivante:

$$\left\{ \begin{array}{l} \tilde{\alpha}_{n+m} = a_n \quad \tilde{c}_1 = b_0 \\ \text{si } \tilde{\alpha}_i = \beta_j \text{ alors } \tilde{c}_{i+1} = b_j \text{ et } \tilde{\alpha}_{n+m-i} = 0 \quad i = 1, \dots, n + m - 1 \\ \text{si } \tilde{\alpha}_i = \alpha_j \text{ alors } \tilde{c}_{i+1} = 0 \text{ et } \tilde{\alpha}_{n+m-i} = a_{n-j} \quad i = 1, \dots, n + m - 1 \end{array} \right. \quad (\text{II.52})$$

3) Déterminer le vecteur d'état selon l'équation (II.44)

$$D^{(\alpha)}(x) = [y \ D^{\alpha_1}y \ D^{\alpha_2}y \ \dots \ D^{\alpha_{n-1}}y]^T \quad (\text{II.53})$$

4) La dérivée d'ordre non entier du vecteur d'état $x(t)$

$$D^{(\tilde{\alpha})}(x) = [D^{\tilde{\alpha}_1}x_1 \ D^{\tilde{\alpha}_2-\tilde{\alpha}_1}x_2 \ \dots \ D^{\tilde{\alpha}_{n+m}-\tilde{\alpha}_{n+m-1}}x_{n+m}]^T \quad (\text{II.54})$$

5) Construire alors le modèle d'état non entier correspondant :

$$\left\{ \begin{array}{l} D^{(\tilde{\alpha})}(x) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ & \ddots & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\tilde{\alpha}_{n+m} & -\tilde{\alpha}_{n+m-1} & -\tilde{\alpha}_{n+m-2} & \dots & -\tilde{\alpha}_2 & -\tilde{\alpha}_1 \end{bmatrix} (x) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \\ y = [\tilde{c}_1 \ \tilde{c}_2 \ \tilde{c}_3 \ \dots \ \tilde{c}_{n+m-1} \ \tilde{c}_{n+m}](x) \end{array} \right. \quad (\text{II.50})$$

Donc le modèle de transfert non entier est donné par :

$$H(s) = \frac{\tilde{c}_1 + \tilde{c}_2 s^{\tilde{\alpha}_1} + \tilde{c}_3 s^{\tilde{\alpha}_1 + \tilde{\alpha}_2} + \dots + \tilde{c}_{n+m} s^{\tilde{\alpha}_1 + \tilde{\alpha}_2 + \dots + \tilde{\alpha}_{n+m-1}}}{s^{\tilde{\alpha}_1 + \tilde{\alpha}_2 + \dots + \tilde{\alpha}_{n+m-1} + \dots + \tilde{\alpha}_{n+m-2}} s^{\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_{n+m-1}} s^{\tilde{\alpha}_1 + \tilde{\alpha}_{n+m}}} \quad (\text{II.51})$$

Remarque :

L'Algorithme qui vient d'être présenté suppose des relations quelconques entre les ordres non entiers de dérivation du numérateur et ceux du dénominateur de $G(s)$. Néanmoins, lorsqu'il existe des ordres de dérivation β_i qui sont égaux à ceux de du dénominateur, cela engendre des dérivées nulles dans le vecteur $D^{(\tilde{\alpha})}(x)$. Dans ce cas, la dimension du modèle d'état non entier peut être réduite, en effectuant une simplification selon la relation donnée par l'étape supplémentaire :

6) si $(\tilde{\alpha}_i - \tilde{\alpha}_{i-1} = 0)$

Eliminer la $i^{\text{ème}}$ ligne de la matrice A et le $(i + 1)^{\text{ème}}$ élément nul du vecteur C .

VI.2. Modèle d'état non entier vers une fonction de transfert

Le modèle d'état d'un système d'ordre non entier, continu invariant s'écrit sous la forme :

$$\begin{cases} D^{(\alpha)}(x) = A x + B u \\ y = C x + E u \end{cases} \quad (\text{II.52})$$

Où :

$$D^{(\alpha)}(x) = [D^{\alpha_1}x_1, D^{\alpha_2}x_2 \dots D^{\alpha_n}x_n]^T \quad (\text{II.53})$$

La relation entre les matrices du modèle d'état (II.52) est la fonction de transfert $G(s)$ peut être calculée en utilisant la transformée de Laplace, et en considérant les conditions initiales nulles, on obtient :

$$G(s) = C[(s^{(\alpha)}I_n - A)^{-1}]B + E \quad (\text{II.54})$$

Où :

$$s^{(\alpha)}I_n = \text{diag}[s^{\alpha_1}, s^{\alpha_2} \dots, s^{\alpha_n}] \quad (\text{II.55})$$

VII. Approximations d'un dérivateur d'ordre non entier

Nous présentons quelques méthodes d'approximation [34] des systèmes d'ordre fractionnaire par transmittance d'ordre entier, établie par souci de leur réalisabilité physique et parce que les outils de calcul et de simulation ne traitent pas encore directement la dérivation/intégration d'ordre fractionnaire.

VII.1. La méthode de Carlson

La méthode de Carlson à été initialement développée pour approximer l'expression $\left(\frac{1}{s}\right)^{1/n}$, qui décrit le condensateur fractionnaire général pour tout entier $n > 1$.

Le problème d'approximation est formulé sous la forme d'une équation $f(x)$ non linéaire monovariante à résoudre par le processus itératif de Newton :

$$f(x) = x^n - a \Rightarrow x = a^{1/n} \quad (\text{II.58})$$

Avec : $a = 1/s$ et $s = j\omega$.

Après un développement mathématique (voir l'Annexe B) la formule d'approximation de $(1/s)^{1/n}$ est la suivante:

$$x_i = x_{i-1} \frac{(n-1)x_{i-1}^n s + (n+1)}{(n+1)x_{i-1}^n s + (n-1)} \quad (\text{II.61})$$

L'algorithme itératif pour l'approximation de $\left(\frac{1}{s}\right)^{1/n}$ est formulé comme suit :

*Entrée des données

- L'ordre entier n
- Fréquence de coupure w_c

- 1) Initialiser $x_0 = 1$
- 2) Faire pour $i = 1, 2, \dots, I$

$$x_i = x_{i-1} \frac{(n-1)x_{i-1}^n s + (n+1)}{(n+1)x_{i-1}^n s + (n-1)}$$

I : nombre d'itérations

$$F_{app}(s) = x_I$$

Table II.4 : Algorithme d'approximation de Carlson

L'approximation de Carlson réalise l'approximation du transfert $\left(\frac{1}{s}\right)^{1/n}$, $n \in \mathbb{N}^*$, qui définit un dérivateur généralisé avec une fréquence de coupure $\omega_c = 1 \text{ rad/sec}$. Pour imposer une fréquence ω_c quelconque, il faut calculer les zéros et les pôles de la transmittance d'approximation obtenue au départ, les multiplier par la fréquence de croisement voulue puis reconstruire la transmittance avec ces nouveaux paramètres.

VII.2. Méthode d'approximation d'Oustaloup

L'approximation d'Oustaloup [2, 21, 27] d'un dérivateur généralisé, dont l'action différentielle couvre tout l'espace fréquentiel, repose sur une distribution récursive d'une infinité de pôles et de zéros réels négatifs (pour assurer le comportement à phase minimale). Soit la fonction de transfert du dérivateur généralisé :

$$D_{gen}(s) = \left(\frac{s}{\omega_c}\right)^\alpha, \alpha \in \mathbb{R} \quad (\text{II.62})$$

ω_c : est la fréquence de croisement.

Dans le cadre d'une synthèse réaliste fondée sur un nombre fini de zéros et de pôles, il convient de réduire le comportement différentiel généralisé sur un intervalle fréquentiel borné, choisi selon les besoins de l'application. Le transfert résultant dit "dérivateur généralisé borné en fréquences" est décrit par:

$$D_{borné}(s) = \left(C_0 \frac{1 + \frac{s}{\omega_b}}{1 + \frac{s}{\omega_h}}\right)^\alpha, \alpha \in \mathbb{R} \quad (\text{II.63})$$

Le dérivateur borné en fréquence, étant lui aussi un transfert fractionnaire, doit être approximé par une transmittance d'ordre entier, construite au moyen des zéros et de pôles distribués récursivement :

$$D_{borné}(s) = \lim_{N \rightarrow \infty} D_N(s) \quad (\text{II.64})$$

$$D_N(s) = \left(\frac{\omega_c}{\omega_h}\right)^\alpha \prod_{k=-N}^N \left(\frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}}\right) \quad (\text{II.65})$$

z_i et p_i sont respectivement le zéros et le pôle de rang i , ces singularités (paires zéro-pôle) au nombre total de $(2N + 1)$ sont distribuées récursivement comme suit :

$$\begin{cases} \frac{p_i}{z_i} = \delta > 0, & \frac{z_{i+1}}{p_i} = \eta > 0 \\ \frac{z_{i+1}}{z_i} = \frac{p_{i+1}}{p_i} = \delta\eta > 1 \end{cases} \quad \begin{matrix} \text{(II. 66)} \\ \text{(II. 67)} \end{matrix}$$

Les rapports δ et η définis par (II.66) et qui impliquent un rapport constant $\delta\eta$ entre deux zéros ou deux pôles consécutifs, sont appelés « facteurs récursifs ». Comme le montre la figure (II.2), l'approximation est réalisée par un lissage de diagramme asymptotique de Bode de $D_N(s)$ pour conduire à ceux de $D_{borné}(s)$. Pour le détail du calcul des singularités (paires zéro-pôle) et le rapport $\delta\eta$ voir l'annexe B.

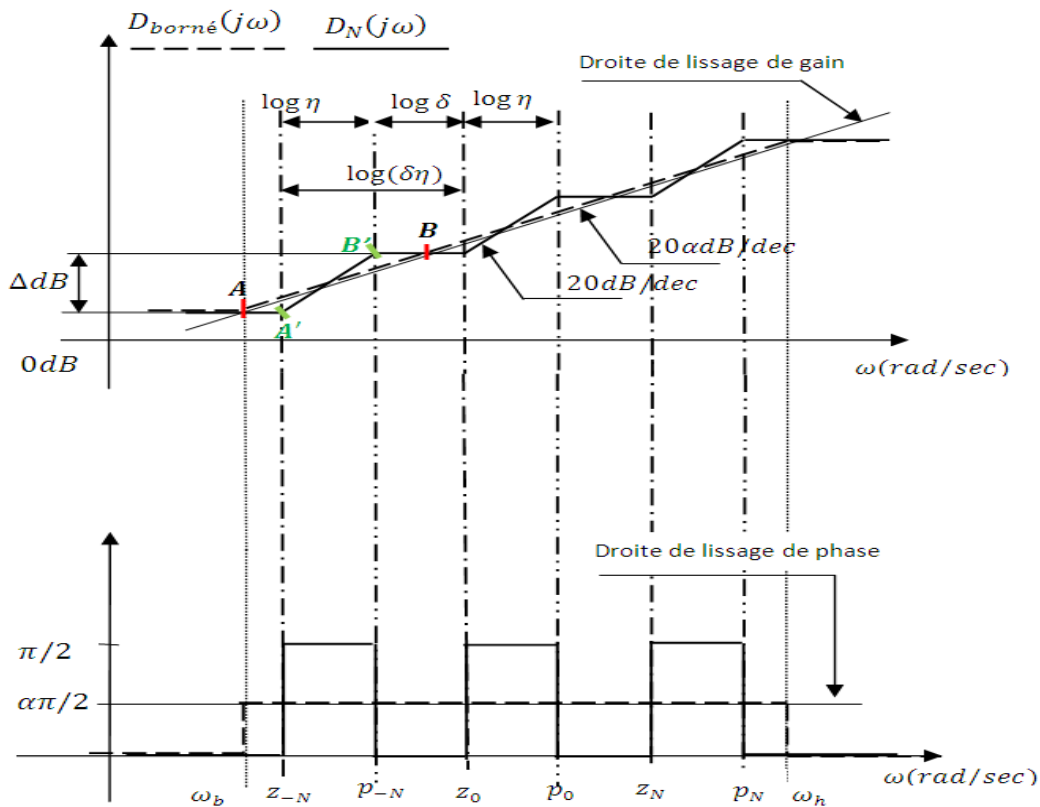


Figure II.2 : Diagramme asymptotique de Bode de $D_{borné}(s)$ et de $D_N(s)$

L'algorithme est donné dans la table (II.5)

<p>*Entrée des données</p> <p>- L'ordre non entier α,</p> <p>- Fréquence de croisement ω_c et la taille $\left(\frac{\omega_B}{\omega_A}\right)$ de la bande de fréquence.</p> <p>1. Calculer ω_B, ω_A</p> $\begin{cases} \omega_A = \frac{\omega_c}{(Taille)^{1/2}} \\ \omega_B = \omega_c (Taille)^{1/2} \end{cases}$ <p>2. calculer ω_b, ω_h</p> $\omega_b = \omega_A/10 \quad \text{et} \quad \omega_h = 10\omega_B$ <p>3. Calculer le nombre total des fréquences transitionnelles N</p> $\begin{cases} (\delta\eta)_{initial} = 5 \\ 2N + 1 = PE \left[\frac{\log(\omega_h/\omega_b)}{\log((\delta\eta)_{initial})} + 0.5 \right] \end{cases}$ <p>4. Calculer les pôles et les zéros</p> <p>Faire pour $i = -N, \dots, 0, 1, 2, \dots, N$</p> $\begin{cases} z_i = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{i+N+1/2-\alpha/2}{2N+1}} \\ p_i = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{i+N+1/2+\alpha/2}{2N+1}} \end{cases} \quad -N \leq i \leq N$ <p>5. Calculer C_0 et la transmittance $D_N(s)$</p> $C_0 = \frac{\omega_b}{\omega_c} = \frac{\omega_c}{\omega_h}$ <ul style="list-style-type: none"> Initialiser $D = 1$ <p>Faire pour $i = -N, \dots, 0, 1, 2, \dots, N$</p> $D = D \frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}}$ <ul style="list-style-type: none"> Calculer $D_N(s) = C_0^\alpha D$
--

Table II.5 : Algorithme d'approximation d'Oustaloup

VII.3 Méthode d'approximation de Charef [22, 2]

Un système fractal dit également pôle à puissance fractionnaire (FPP : Fractional Power Pole) est approximé par une fonction de singularité constituée d'une série de paires pôle-zéro dont le nombre et la distribution fréquentielle dépendent, notamment, d'une erreur d'approximation définie au préalable. Les systèmes fractals sont représentés par une transmittance de la forme :

$$H(s) = \frac{1}{s^\alpha} \quad , \quad \alpha \in \mathbb{R} \quad (\text{II.68})$$

Généralement, les systèmes fractals ont la propriété d'avoir une amplitude finie en basses fréquences, d'où l'on approprie un pôle à puissance fractionnaire :

$$H(s) = \frac{1}{\left(1 + \frac{s}{\omega_c}\right)^\alpha} \quad , \quad 0 < \alpha < 1 \quad (\text{II.69})$$

ω_c : est la fréquence de cassure.

Le lieu de Bode de la fonction de transfert non entière peut être approximé par un transfert d'ordre entier. En mettant en série plusieurs filtres passe bande, la transmittance équivalente est donnée par :

$$H(s) = \lim_{N \rightarrow \infty} \frac{\prod_{i=0}^{N-1} \left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^N \left(1 + \frac{s}{p_i}\right)} \quad (\text{II.70})$$

Pour que le transfert recherché soit de dimension finie, le nombre N doit être également fini, l'approximation de $H(s)$ est donnée par :

$$H(s) \approx \frac{\prod_{i=0}^{N-1} \left(1 + \frac{s}{z_i}\right)}{\prod_{i=0}^N \left(1 + \frac{s}{p_i}\right)} \quad (\text{II.71})$$

Charef utilise comme paramètre principal l'écart maximal, $\varepsilon > 0$ (en décibel) entre la ligne en zigzag et la droite de pente $-20\alpha \text{dB/décade}$, pour déterminer les $2N + 1$ pôles et zéros avec un simple calcul géométrique, voir (Annexe B).

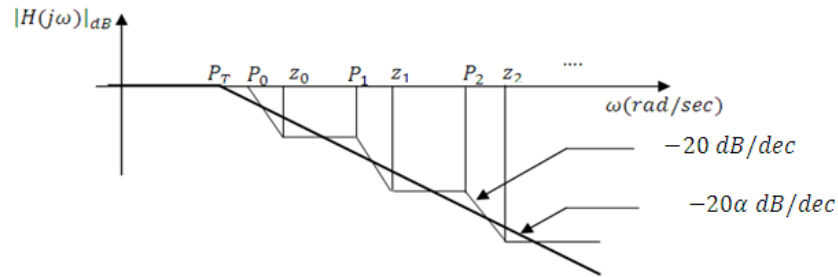


Figure II.3 : Diagramme asymptotique d’amplitude de Bode de et son approximation

Après quelques opérations algébriques simples (détaillées dans l’Annexe B), l’algorithme qui permet de générer les différentes singularités à partir du premier pôle est :

*Entrée des données :

- L’écart maximal et la taille du spectre de fréquences ;
- L’ordre non entier ,la fréquence de croisement

1. Calculer les constantes et b

$$= \frac{\quad}{\quad}$$

2. Calculer des pôles
 - Calculer
 - Calculer le nombre de singularité

$$\frac{\quad}{\quad} + 1$$

<p>Faire pour $i = 1, 2, \dots, N$</p> $p_i = (ab)^i p_0$ <p>3. Calculer les zéros</p> <p>Faire pour $i = 0, 1, 2, \dots, N - 1$</p> $z_i = (ab)^i a p_0$ <p>4. Calcul de la transmittance</p> <ul style="list-style-type: none"> • Initialiser $Num = 1$ <p>Faire pour $i = 0, 1, 2, \dots, N - 1$</p> $Num = Num \left(1 + \frac{s}{z_i} \right)$ <ul style="list-style-type: none"> • Initialiser $Den = 1$ <p>Faire pour $i = 0, 1, 2, \dots, N$</p> $Den = Den \left(1 + \frac{s}{p_i} \right)$ <p>Calculer $H(s) = \frac{Num}{Den}$</p>
--

Table II.6 : Algorithme d'approximation de Charef

Remarque :

Il est à signaler que l'algorithme, tel que développé, ne permet que l'approximation de l'intégrateur fractionnaire borné en fréquences, pour l'approximation du dérivateur fractionnaire borné en fréquences, nous utilisons le même algorithme, puis on remplace les pôles par des zéros et les zéros par des pôles.

VII.4 Méthode d'Oustaloup modifiée

Afin d'améliorer la méthode d'Oustaloup [35], on extrait un filtre à partir de l'expression du dérivateur généralisé borné en fréquences, avant d'approximer ce dernier au moyen de l'algorithme d'Oustaloup.

Dans cette méthode, l'opérateur de dérivation généralisé s^α ($\alpha \in \mathbb{R}$) est mis sous la forme bornée en fréquences :

$$D(s) = \left(\frac{1 + \frac{s}{d}}{\frac{b\omega b}{1 + \frac{s}{d\omega h}}} \right)^\alpha \quad (\text{II.72})$$

Où $s = j\omega$, $b = 10$, $d = 9$, la transmittance peut s'écrire sous la forme :

$$D(s) = \left(\frac{bs}{d\omega b} \right)^\alpha \left(1 + \frac{-ds^2 + d}{ds^2 + b\omega h s} \right)^\alpha \quad (\text{II.73})$$

Le développement en série de Taylor de la partie droite de l'expression donne :

$$D(s) = \left(\frac{bs}{d\omega b} \right)^\alpha \left(1 + \alpha p(s) + \frac{\alpha(\alpha-1)}{2} p^2(s) + \dots \right) \quad (\text{II.74})$$

Où
$$p(s) = \frac{-ds^2 + d}{ds^2 + b\omega h s} \quad (\text{II.75})$$

$$s^\alpha = \frac{(d\omega b)^\alpha}{b^\alpha \left(1 + \alpha p(s) + \frac{\alpha(\alpha-1)}{2} p^2(s) + \dots \right)} \left(\frac{1 + \frac{s}{d}}{\frac{b\omega b}{1 + \frac{s}{d\omega h}}} \right)^\alpha \quad (\text{II.76})$$

En réduisant les série de Taylor dans l'expression à l'ordre 1, on obtient :

$$s^\alpha \approx \left(\frac{1}{1 + \alpha p(s)} \right) \left(\frac{d\omega b}{b} \right)^\alpha \left(\frac{1 + \frac{s}{d}}{\frac{b\omega b}{1 + \frac{s}{d\omega h}}} \right)^\alpha \quad (\text{II.77})$$

En substituant $p(s)$ par son expression, on aura la formule suivante :

$$s^\alpha \approx \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha} \right) \left(\frac{d\omega_b}{b} \right)^\alpha \left(\frac{1 + \frac{s}{\frac{d}{b}\omega_b}}{1 + \frac{s}{\frac{b}{d}\omega_h}} \right)^\alpha \quad (\text{II.78})$$

En développant la partie non entière de l'expression (II.78) par l'algorithme d'Oustaloup :

$$D(s) = \lim_{N \rightarrow \infty} D_N(s) = \lim_{N \rightarrow \infty} \prod_{i=-N}^N \frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}} \quad (\text{II.79})$$

Ce qui donne les relations suivantes :

$$\begin{cases} z_i = \left(\frac{d\omega_b}{b} \right)^{\frac{\alpha-2i}{2N+1}} \\ p_i = \left(\frac{b\omega_h}{d} \right)^{\frac{\alpha+2i}{2N+1}} \end{cases} \quad -N \leq i \leq N \quad (\text{II.80})$$

Finalement,

$$s^\alpha \approx K \left(\frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha} \right) \prod_{i=-N}^N \frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}} \quad (\text{II.81})$$

K est calculé de telle façon à avoir un gain unité à la fréquence $\omega_c = 1 \text{ rad/sec}$, fréquence de cassure de l'opérateur s^α :

$$K = \left(\frac{d\omega_b}{b} \right)^\alpha \prod_{i=-N}^N \frac{z_i}{p_i} \quad (\text{II.82})$$

Il est indispensable de vérifier la stabilité de la transmittance. Comme les pôles et les zéros calculés au moyen de l'algorithme d'Oustaloup sont strictement négatifs (fréquences transitionnelles comprises entre ω_b et ω_h avec $\omega_b > 0$ et $\omega_h > 0$), il reste de vérifier la stabilité du filtre additionnel défini par :

$$D'(s) = \frac{ds^2 + b\omega_h s}{d(1-\alpha)s^2 + b\omega_h s + d\alpha} \quad (\text{II.83})$$

Cette transmittance étant de second ordre, elle est stable à condition que tous les coefficients de son dénominateur soient positifs, ce qui est vérifié si $\alpha < 1$.

L'algorithme est donné la table(II.7)

* Entrée des données	
- L'ordre non entier α	
-Fréquence de croisement ω_c et les fréquences transitionnelles ω_b et ω_h	
1. Calculer la transmittance $D(s)$	
$b = 10, d = 9 ; P = 1 ; K=1$	
Faire pour $i = -N, ..0,1,2, \dots, N$	
1)	$z_i = \left(\frac{d\omega_b}{b}\right)^{\frac{\alpha-2i}{2N+1}}$
2)	$p_i = \left(\frac{b\omega_h}{d}\right)^{\frac{\alpha+2i}{2N+1}}$
3)	$K = K \left(\frac{d\omega_b}{b}\right)^\alpha \frac{z_i}{p_i}$
4)	$P = P \left(\frac{1 + \frac{s}{z_i}}{1 + \frac{s}{p_i}}\right)$
Fin pour	
$F = \frac{ds^2 + b\omega_h s}{d(1 - \alpha)s^2 + b\omega_h s + d\alpha}$	
$D(s) = K \times F \times P$	

Table II.7 : Algorithme d'approximation d'Oustaloup modifiée

VIII. Conclusion

Dans ce chapitre, les algorithmes de calcul fractionnaire appliqués à l'analyse des systèmes d'ordre non entier ont été élaborés.

L'algorithme de calcul de la réponse d'un modèle d'état d'ordre non entier n'a pu être réalisé en raison de la complexité du produit de convolution, faisant intervenir la fonction de Mittag-Leffler.

En contrepartie, les réponses d'un système fractionnaire peuvent être simulées sur la base de la résolution des équations différentielles. D'autre part les différentes approximations nécessaires à l'implémentation des filtres et des régulateurs fractionnaires ont été étudiées et les algorithmes correspondants ont été élaborés.

Les différents algorithmes présentés vont nous permettre de construire le logiciel sous Matlab dédié à l'analyse et la simulation des systèmes fractionnaires.

Chapitre III :

Réalisation du logiciel et application

I. Introduction

Dans ce chapitre nous exposons brièvement l'utilitaire « GUIDE » de Matlab qui permet de créer des interfaces graphiques, puis nous allons présenter notre logiciel appelé « logiciel dédié à l'analyse et la simulation des systèmes dynamiques d'ordre fractionnaire » réalisé sous Matlab version 7.7.0(R2008b).

L'objectif de ce logiciel est de faciliter le calcul fractionnaire et la simulation des systèmes dynamiques d'ordre fractionnaire, nous proposons de l'exploiter à travers ses différentes parties en illustrant avec des exemples d'application pour chaque interface. Nous avons consulté les ouvrages suivants [17, 6, 15].

II. Création d'une interface graphique sous Matlab

Les **IHM** (Interfaces Homme Machine), sont appelées GUI (Graphical User Interface) dans Matlab. Elles permettent de créer des interfaces où le créateur choisit plusieurs types d'objets (*boutons, editbox, listbox, ...*) appelés *handles*, puis réalise la programmation pour obtenir l'interaction souhaitée entre ces différents objets.

On peut procéder de deux manières pour créer une interface graphique avec Matlab soit :

✓ **par programmation**

Les objets graphiques constituant l'interface sont créés et modifiés par programmation.

✓ **en utilisant l'outil GUIDE**

Le logiciel Matlab offre à l'utilisateur la possibilité de créer ses propres interfaces (communément dites simulateurs) en utilisant un ensemble de commandes, et des fonctions destinées à cela, ceci est possible notamment, grâce à son outil « GUIDE ».

Du fait du nombre important d'objets et surtout du nombre encore plus élevé de paramètres associés, leur programmation "à la main" déroutent généralement le débutant. Depuis la version 5.0 (1997), Matlab possède un outil IDE dédié à la création des

interfaces graphiques. Cet outil, appelé **GUIDE** (**G**raphical **U**ser **I**nterface **D**evelopment **E**nvironment), permet de concevoir intuitivement ces interfaces graphiques.

III. Eléments d'une interface graphique

Les différents éléments qui constituent une interface graphique sont:

III.1. Les contrôles

Ce sont des objets graphiques qui exécutent des actions, il en existe plusieurs types :

- Boutons poussoirs : pour l'exécution de l'action ;
- Boutons radio (d'option) : permettent un choix exclusif ;
- Case à cocher : pour un choix non exclusif ;
- Textes statiques : pour l'affichage des textes, des chaînes de caractères, non modifiables par l'utilisateur ;
- Textes éditables : pour la récupération des données et l'affichage des résultats ;
- Cadres (frame) : bordure pour la formation des groupes logiques ;
- Axes : zones graphiques, dans lesquelles l'utilisateur peut réaliser des tracés et des images ;
- Barres de défilement : objets qui permettent de choisir une valeur dans un intervalle donné ;

✓ **Propriétés des contrôles**

Tous les objets graphiques ont des propriétés qui contrôlent leurs affichages sur la figure, certaines d'entre elles ne sont valables que pour quelques objets. Les plus utilisées sont (style, position, callback, string, value,...).

String : elle représente :

- Le texte qui apparaît sur le contrôle dans le cas des boutons poussoirs ;
- Le texte à côté des boutons radio et des cases à cocher ;
- La valeur par défaut dans le cas d'un texte éditable ;
- La liste des choix pour les menus ;

- Le texte apparaît sur le contrôle dans le texte statique ;

Value : représente la valeur courante du contrôle, elle prend la valeur 1 pour un contrôle actif et la valeur 0 pour un contrôle inactif.

Style : elle permet d'identifier le type de contrôle.

Position : permet de positionner l'objet et de limiter sa taille, cette propriété est donnée par un vecteur ligne de quatre éléments.

Position = [gauche, bas, largeur, hauteur]

Callback : propriété réservée aux contrôles exécutant des actions, qui est une chaîne de caractères pouvant être une commande de Matlab, le nom d'une variable ou **d'un fichier.m**.

III.2. Les menus

Ce sont les objets qui s'ajoutent à la barre des menus d'une figure et d'une interface.

La création des menus est devenue aisée avec l'éditeur du «GUIDE», car il suffit d'un simple clic sur "Menu Editor" puis un autre clic sur "New Menu" pour créer un nouveau menu et ses options, puis il faut spécifier ses propriétés.

✓ Les propriétés d'un menu

Label : permet de spécifier le nom du menu ou de l'option.

Tag : c'est le nom du menu ou d'une option, utilisé pour la récupération des handles.

Callback : c'est le code événement.

IV. Exemple de création d'une interface

La version 7.7.0 de MATLAB possède un outil nommé GUIDE (Graphical User Interface Development Environment) facilitant la génération des scripts MATLAB nécessaires à la création d'une interface utilisateur.

Tapez "guide" dans l'espace "Command Windows" de Matlab, la fenêtre suivante s'ouvre :

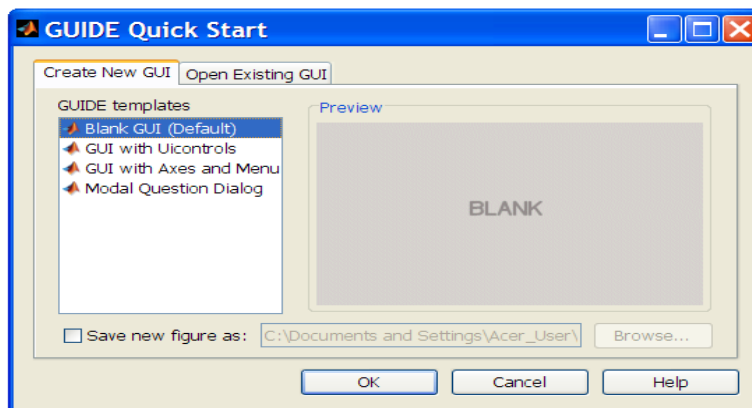


Figure III.1 : Configuration principale du GUIDE de Matlab

Cliquez sur "OK" pour créer une nouvelle interface. La fenêtre d'édition d'interface ci dessous s'ouvre :

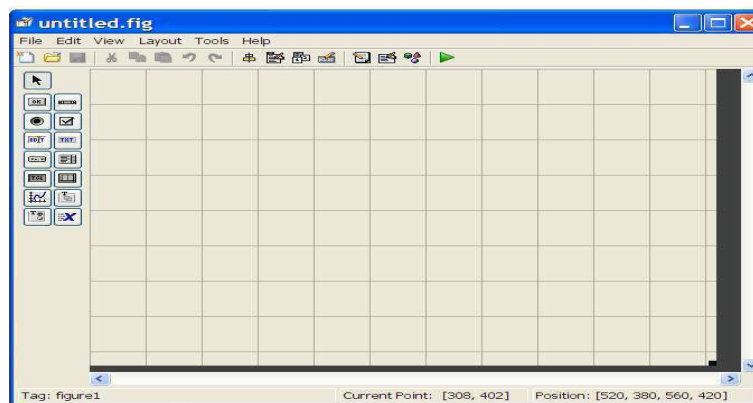


Figure III.2: Création d'une nouvelle interface graphique sur le GUIDE

Cette fenêtre permet l'ajout, l'édition et la suppression des différents éléments de l'interface.

Pour éditer les propriétés de la figure, double-cliquez dessus, la fenêtre d'édition des propriétés s'ouvre :

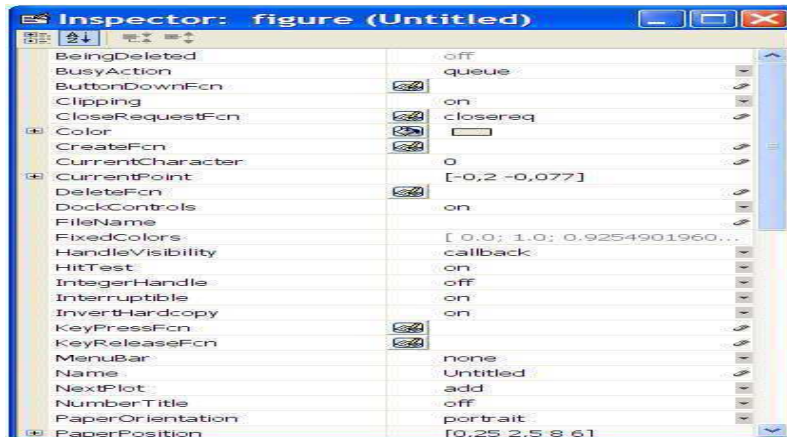


Figure III.3 : Propriétés d'une interface graphique

Pour former une interface souhaitée, choisir les handles disponibles dans la fenêtre de création d'interface, on présente quelques éléments :



: Pour l'axe, il faut préciser les propriétés Units, Position et Tag (axes1).



: Pour le menu popup, il faut préciser les propriétés Units, Position, String (les noms des différents items), et Tag (popup1).



: Pour le bouton on-off, il faut préciser les propriétés Units (pixels), Position, Min (0), Max (1), Value (0), String (Pointillé), et Tag (slider1).



: Pour l'ascenseur, il faut préciser les propriétés Units (pixels), Position, Min (1), Max (5), Value (5), et Tag (slider1).



: Pour le label, il faut préciser les propriétés Units (pixels), Position, String, et Tag (text1).



: Pour le bouton, il faut préciser les propriétés Units (pixels), Position, String (Quitte), et Tag (pushbutton1).

Exemple de création

On présente un exemple de la création d'une interface graphique pour la simulation d'une fonction donnée, pour sa réalisation on a choisit les objets suivant :

1. Un cadre pour la bordure.
2. Un axe, pour l'affichage de la courbe.
3. Deux textes éditables, pour l'introduction les paramètres.
4. Deux textes statiques, pour indiquer les paramètres à introduire et le résultat obtenu.
5. Un bouton poussoir, pour l'exécution de l'action « simuler ».

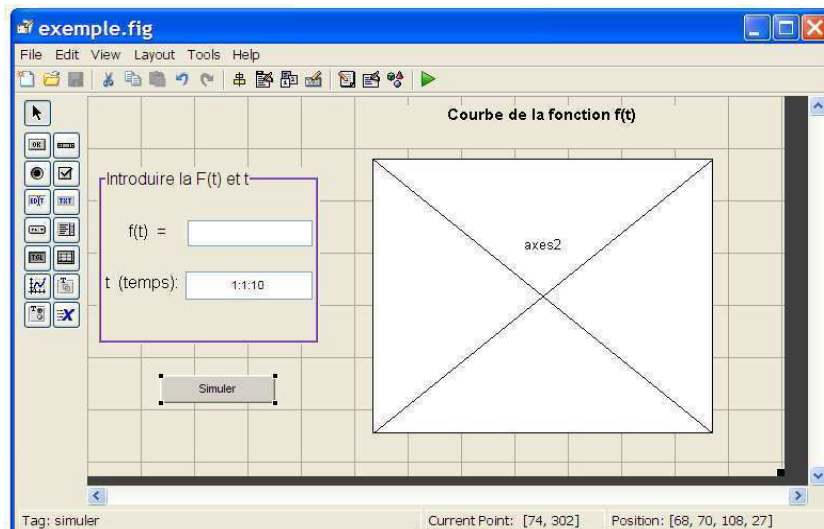


Figure III.4: Réalisation d'une interface graphique

Sauvegarder l'interface ainsi créée "File Save" en la nommant « exemple.fig »
Le fichier.fig est un fichier Matlab contenant l'ensemble de l'interface graphique,
à ce fichier.fig est associé un fichier.m contenant le code Matlab correspondant
qu'il va falloir compléter pour ajouter les différentes actions associées à l'interface.

On associe à la propriété "callback" du pushbutton que nous avons appelé "Simuler" dans le fichier « exemple.m », le programme qui permet de simuler une fonction donnée $f(t)$ (Figure III.5). Le fichier « exemple.m » est complet et peut être exécuté (Figure III.6)

```

% --- Executes on button press in simuler.
function simuler_Callback(hObject, eventdata, handles)
% hObject    handle to simuler (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes2);
cla;
f=get(handles.f,'string');
t=get(handles.t,'string');
F=num2str(f);
t=str2num(t);
F=str2mat(F);
h=t(2)-t(1);
N=(t(end)-t(1))/h;
for i=1:N
    t(i)=(i-1)*h;
    r(i)=subs(F,t(i));
    plot(r)
end
xlabel('Temps(sec)');
ylabel('f(t)');
grid
h = legend('f(t)',2);
set(h,'Interpreter','none');
g=['la courbe de f(t)= ', f];
set(handles.rr,'string',g);

```

Figure III.5 : Programme ajouté au fichier « exemple.m »

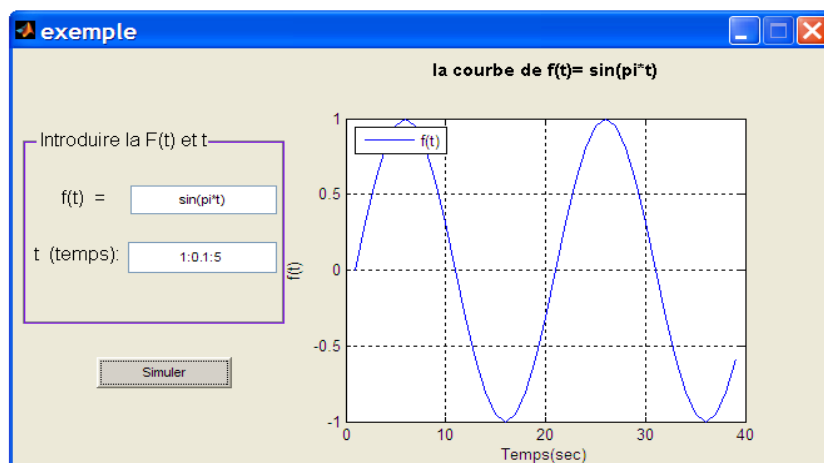


Figure III.6 : Interface qui simule une fonction $f(t)$

V. Présentation générale du logiciel

L'objectif de ce logiciel est de faciliter le calcul fractionnaire et la simulation des systèmes dynamiques d'ordre fractionnaire, pour en tirer profit du logiciel, il est nécessaire de l'utiliser de façon correcte. Pour cela, nous proposons de l'exploiter à travers ses différentes parties en illustrant avec des exemples les interfaces graphiques construites sous Matlab, qui sont dédiées exclusivement à l'analyse et la simulation des systèmes dynamiques d'ordre non entier.

Le premier contact avec notre logiciel est une fenêtre d'accueil (Figure III.7) qui permet en cliquant sur le bouton « Entrer » l'accès à la fenêtre du menu principal du logiciel (Figure III.8).

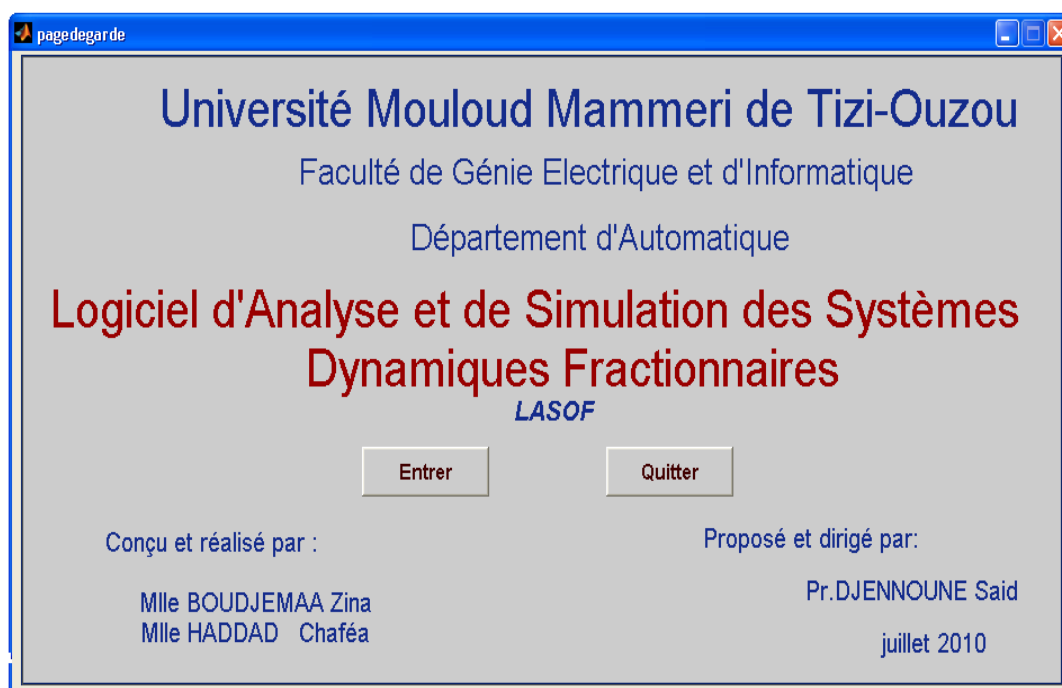


Figure III.7 : Interface d'accueil du logiciel

Le logiciel permet de traiter quatre parties qui sont :

1. Calcul fractionnaire
2. Modèle de représentation
3. Résolution d'un système d'ordre fractionnaire
4. Approximation du dérivateur généralisé

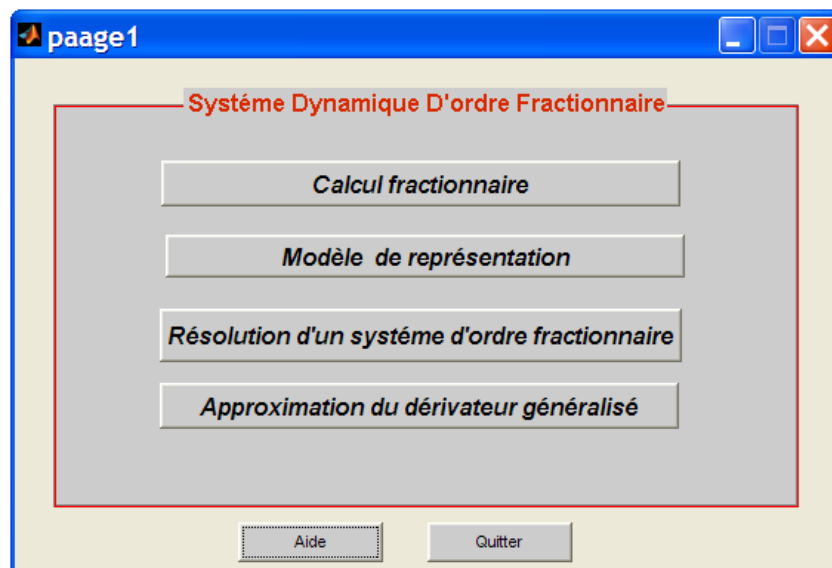


Figure III.8 : Interface du menu principal

Chaque partie fait appel à une interface de sélection (Figures III.9, III.15, III.22, III.28) dont le bouton " Ok " permet d'accéder à d'autres interfaces d'opérations qui contiennent des zones d'édition, des zones d'affichage, des radios boutons, des listes de sélection et des boutons poussoirs pour les actions.

VI. Application

Nous exposons dans ce qui suit les différentes parties du logiciel avec quelques exemples d'application :

VI.1. Calcul fractionnaire

Cette interface permet la simulation de la dérivée et de l'intégrale non entières, ainsi que le calcul de la transformée de Laplace fractionnaire et les racines d'un polynôme d'ordre fractionnaire. Sélectionner un choix puis cliquer sur le bouton "OK".

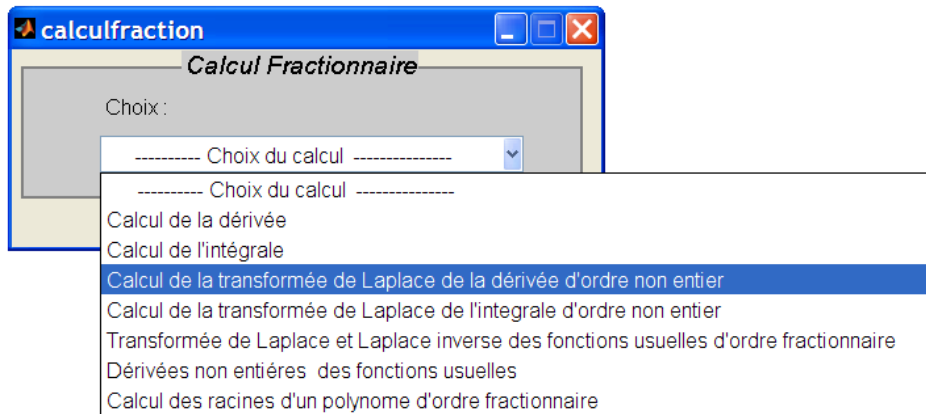


Figure III.9 : Interface de sélection pour le calcul fractionnaire

VI.1.1. Dérivation/Intégration fractionnaire

Pour simuler La dérivée d'ordre fractionnaire d'une fonction $f(t)$, nous devons introduire l'expression analytique de la fonction, l'ordre de dérivation et l'intervalle du temps tout en indiquant le pas d'échantillonnage.

La figure(III.10) montre un exemple de simulation de la dérivée d'ordre 0.5 d'un échelon unitaire sur un intervalle de temps $t = [0 : 0.1 : 10]$.

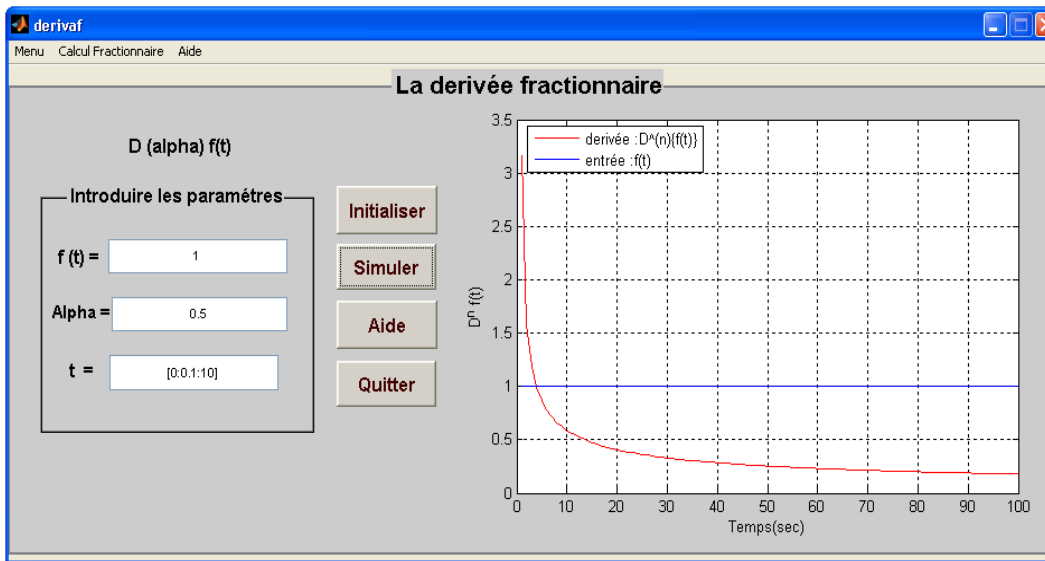


Figure III.10 : L'interface de la dérivée fractionnaire

Cette interface montre la courbe représentative de la dérivée d'ordre 0.5 de la fonction $f(t) = 1$, la courbe résultante met en évidence la propriété importante des systèmes fractionnaires qui est la lenteur d'établissement. En effet, la dérivée à l'ordre 0.5 de l'échelon unitaire possède l'allure d'un pic de Dirac mais avec une forme plus étalée sur l'axe des temps, la dérivée fractionnaire met ainsi beaucoup de temps à s'annuler.

Pour la visualisation de l'intégration fractionnaire nous procédons de la même manière que pour la dérivée fractionnaire.

La figure (III.11) représente l'interface de simulation de l'intégration fractionnaire d'ordre 0.5 pour un échelon unitaire.

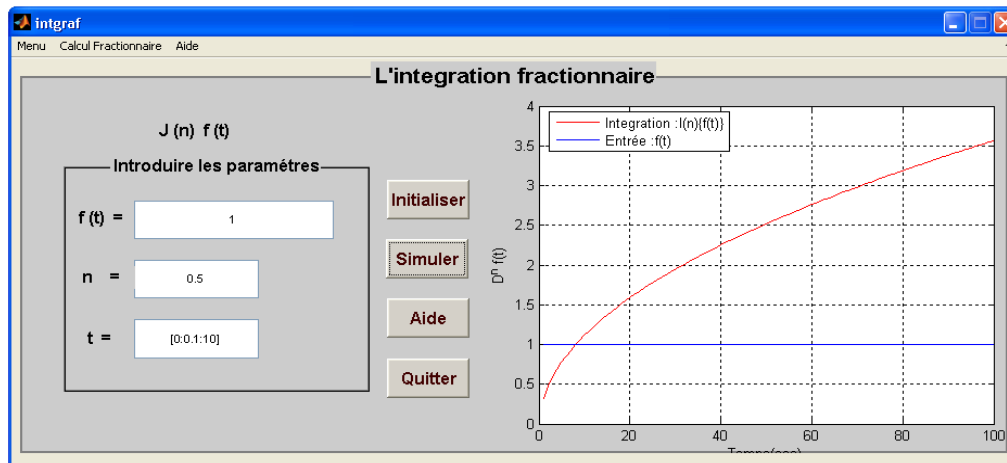


Figure III.11: L'interface de l'intégrale d'ordre fractionnaire

A titre d'illustration la figure (III.12) représente les courbes des intégrales d'ordre non entier 0.5, 0.99, 1.5 d'un échelon unitaire et la courbe de l'échelon unitaire.

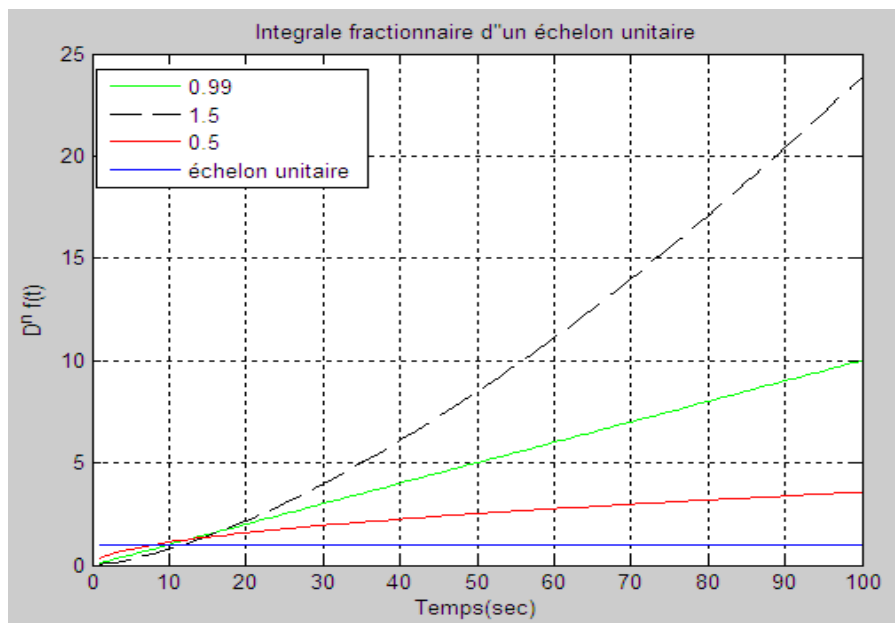


Figure III.12: Courbes d'intégrale pour différents ordres fractionnaires

VI.1.2. La transformée de Laplace (TL) de la dérivée /intégrale fractionnaire

Dans cette interface, nous allons introduire l'expression analytique de la fonction $f(t)$, ainsi que l'ordre non entier n , puis on choisit l'opération à faire : TL de l'intégrale ou TL de la dérivée.

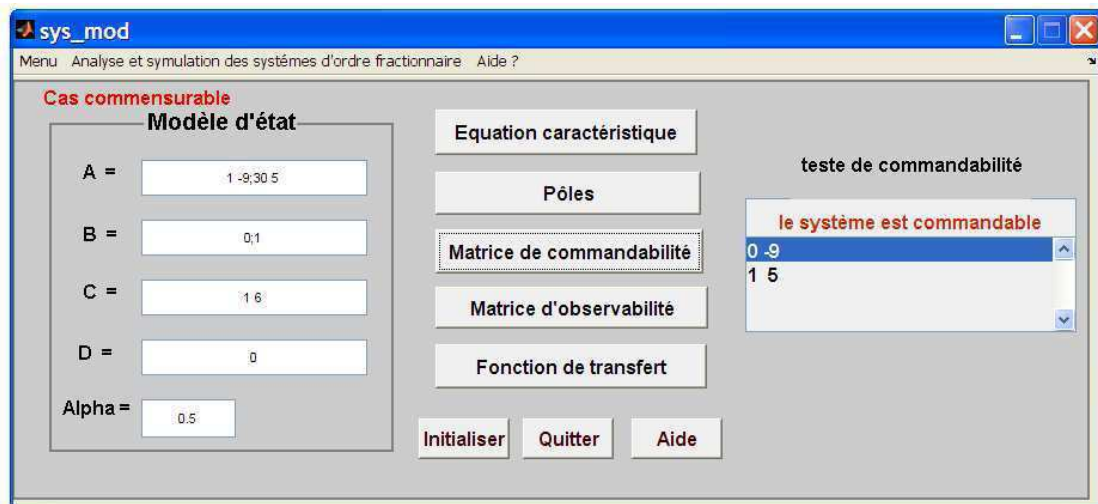


Figure III.13 : L'interface de la TL de la dérivée/intégrale fractionnaire

VI.1.3. Calcul des racines d'un polynôme d'ordre fractionnaire

Soit à résoudre le polynôme à puissances non entières réelles suivant :

$$P(s) = 1s^{0.8} - 3s^{0.2} + 2 \tag{III.1}$$

$$\begin{cases} 0.8 = \frac{4}{5} \\ 0.2 = \frac{1}{5} \end{cases} \tag{III.2}$$

$$P(s) = 1s^{4/5} - 3s^{1/5} + 2 \tag{III.3}$$

Le changement de variable $p = s^{1/5}$ permet de résoudre un polynôme d'ordre entier en fonction de p :

$$P(p) = p^4 - 3p^1 + 2 \quad (\text{III.4})$$

Calculer les racines du polynôme $P(p)$:

$$\begin{cases} p_1 = -0.9053 + 1.2837j \approx 1.5708e^{\frac{9}{13}\pi j} \\ p_2 = -0.9053 - 1.2837j \approx 1.5708e^{-\frac{9}{13}\pi j} \\ p_3 = -0.8105 = 0.8105 e^{0j} \\ p_4 = 1 = 1 e^{0j} \end{cases} \quad (\text{III.5})$$

La condition d'existence des racines du polynôme fractionnaire s'exprime dans ce cas par :

$$\begin{cases} \text{Pour } p_1: & -\frac{29}{65} < k < -\frac{16}{65} \Rightarrow k \text{ n'existe pas} \\ \text{Pour } p_2: & -\frac{16}{65} < k < -\frac{29}{65} \Rightarrow k \text{ n'existe pas} \\ \text{Pour } p_3 \text{ et } p_4: & -\frac{1}{10} < k < +\frac{1}{10} \Rightarrow k = 0 \end{cases} \quad (\text{III.6})$$

Donc seules les racines p_3 et p_4 du polynôme entier engendrent des racines au polynôme fractionnaire correspondant :

$$s_3 = 0.3498 \quad , \quad s_4 = 1 \quad (\text{III.7})$$

En remplaçant ces solutions dans le polynôme $P(s)$, on obtient :

$$P(s_3) = 1.3306 * 10^{-5} \quad , \quad P(s_4) = 0 \quad (\text{III.8})$$

Remarque :

Les racines calculées d'un polynôme d'ordre fractionnaire ne sont que des racines approximées.

Le résultat du calcul est montré dans l'interface suivante :

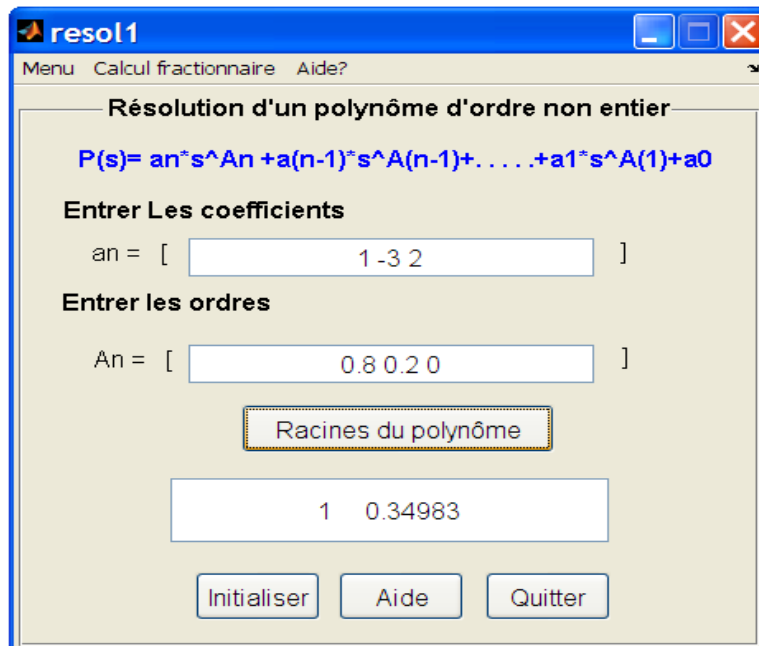


Figure III.14 : L'interface de résolution d'un polynôme fractionnaire

VI.2. Modèles de représentation

Un système est dit fractionnaire lorsqu'il est décrit par une équation différentielle impliquant des dérivées fractionnaires. Un système peut être représenté soit par une fonction de transfert (domaine fréquentiel), soit par un modèle d'état (domaine temporel), dans cette partie nous allons présenter les différents passages entre ces modèles.

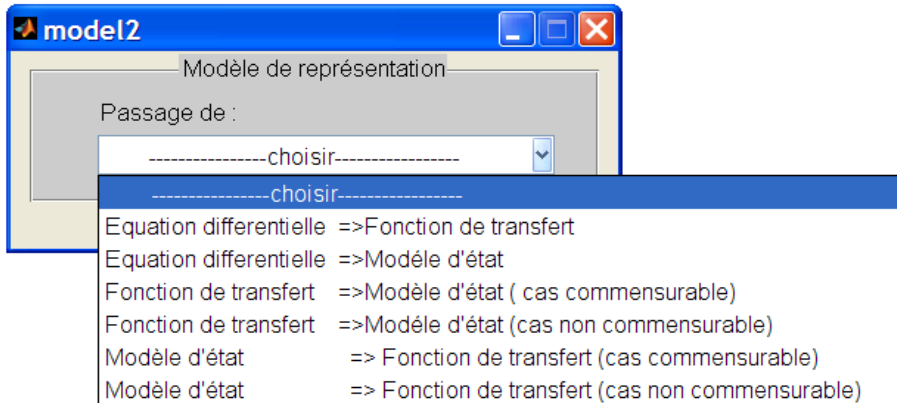


Figure III.15 : Interface de sélection des modèles de représentation

VI.2.1 Passage d’une équation différentielle d’ordre fractionnaire vers une fonction de transfert

Soit l’équation différentielle linéaire d’ordre fractionnaire à coefficients constants:

$$D y(t) + 0.2D^{0.8}y(t)+9 D^{0.2}y(t) + 3y(t)=3D^{0.6}u(t) + D^{0.3}u(t)+2u(t) \quad (III.9)$$

L’application de la transformée de Laplace à l’équation (III.9), en considérant les conditions initiales nulles, permet d’obtenir :

$$s Y(s) + 0.2s^{0.8}Y(s)+9 s^{0.2}Y(s) + 3Y(s)=3s^{0.6}U(s) + s^{0.3}U(s)+ 2U(s) \quad (III.10)$$

La fonction de transfert s’écrit donc :

$$\frac{Y(s)}{U(s)} = \frac{(3s^{0.6}+s^{0.3}+2)}{(s+0.2s^{0.8}+9s^{0.2}+3)} \quad (III.11)$$

Le résultat est affiché dans figure (III.16)

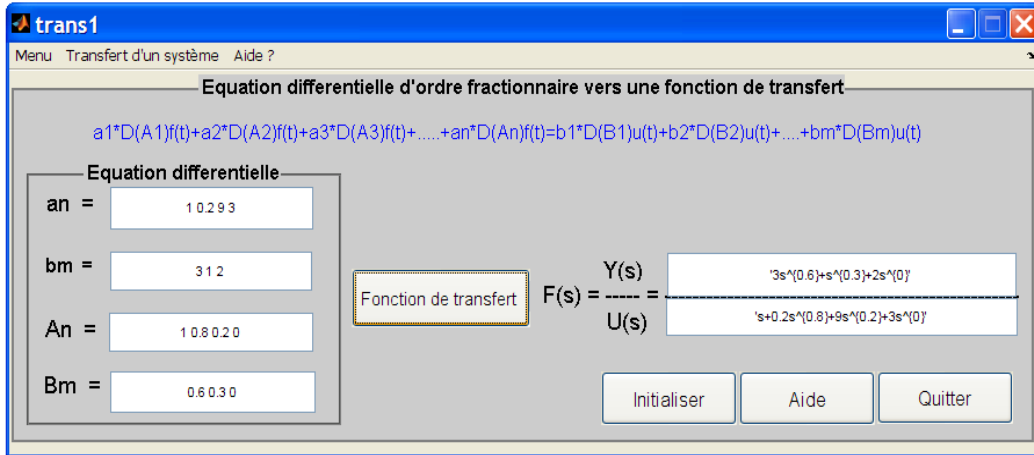


Figure III.16 : L'interface du passage d'une équation différentielle vers une fonction de transfert

VI.2.2. Passage de la représentation transfert non entière à la représentation d'état non entière

❖ **Cas des systèmes commensurables**

Soit un système non entier monovariante linéaire représenté par son modèle de transfert $G(s)$:

$$G(s) = \frac{3s^{0.3}}{2s^{0.4} + 9s^{0.1} + 3} \tag{III.12}$$

A l'aide du changement de variable

$$p = s^\alpha \quad \text{avec } \alpha = 0.1 \tag{III.13}$$

On obtient:

$$G(p) = \frac{3p^3}{2p^4 + 9p^{1+3}} \tag{III.14}$$

Le modèle d'état correspondant au modèle transfert commensurable $G(p)$ est obtenu par la méthode classique ou par la commande «tf2ss» de Matlab:

>> [A,B,C,D]=tf2ss([3 0 0 0],[2 0 0 9 3])

$$A = \begin{bmatrix} 0 & 0 & -4.5 & -1.5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; C = [1.5 \ 0 \ 0 \ 0]; D = 0$$

Nous allons introduire les coefficients de la fonction de transfert entière (le dénominateur et le numérateur) et l'ordre non entier α , le modèle d'état non entier s'affiche en cliquant sur le bouton « **modèle d'état** », l'interface suivante montre le résultat :

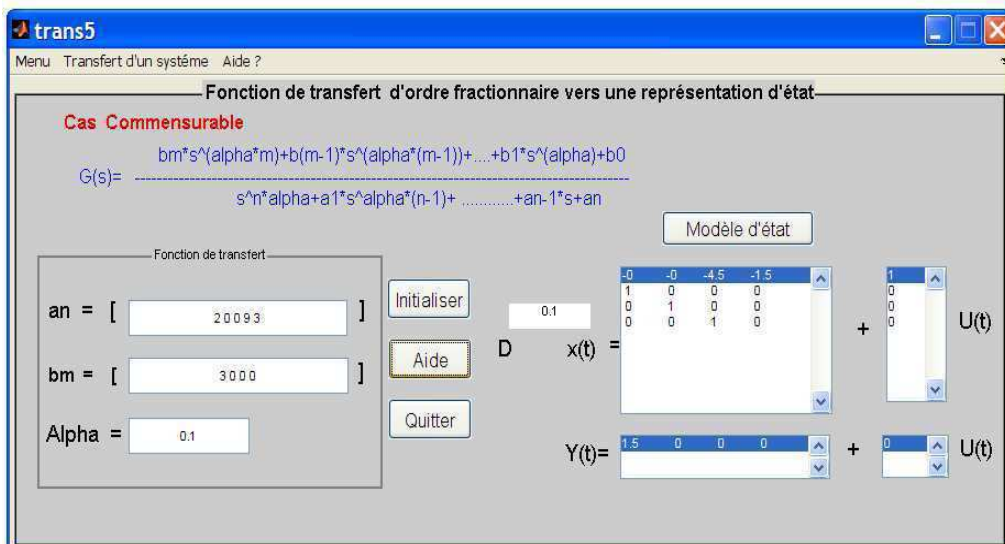


Figure III.17: L'interface du passage d'une fonction de transfert vers une représentation d'état (cas commensurable)

❖ Cas des systèmes non commensurables

Considérons le modèle de transfert :

$$G(s) = \frac{3s^{0.6} + s^{0.3} + 2}{s + 0.2s^{0.8} + 0.9s^{0.2} + 3} \tag{III.14}$$

$$\begin{cases} \beta_2 = 0.6 ; b_2 = 3 ; \beta_1 = 0.6 ; b_1 = 1 ; b_0 = 2 \\ \alpha_3 = 1 ; \alpha_2 = 0.8 ; a_1 = 0.2 ; \alpha_1 = 0.2 ; a_2 = 0.9 ; a_3 = 3 \end{cases} \quad (\text{III.15})$$

$$\text{Le vecteur } \tilde{\alpha} \text{ est : } \tilde{\alpha} = [1 \ 0.8 \ 0.6 \ 0.3 \ 0.2] \quad (\text{III.16})$$

En utilisant la procédure de sélection (définie par l'équation (II.49) voire chapitre 2), on obtient :

$$\begin{cases} \tilde{\alpha}_5 = a_3 = 3 & \tilde{c}_1 = b_0 = 2 \\ \tilde{\alpha}_1 = \alpha_1 & \Rightarrow \tilde{c}_2 = 0 \quad \text{et} \quad \tilde{\alpha}_4 = a_2 = 0.9 \\ \tilde{\alpha}_2 = \beta_1 & \Rightarrow \tilde{c}_3 = 1 \quad \text{et} \quad \tilde{\alpha}_3 = 0 \\ \tilde{\alpha}_3 = \beta_2 & \Rightarrow \tilde{c}_4 = 3 \quad \text{et} \quad \tilde{\alpha}_2 = 0 \\ \tilde{\alpha}_4 = \alpha_2 & \Rightarrow \tilde{c}_5 = 0 \quad \text{et} \quad \tilde{\alpha}_1 = a_1 = 0.2 \end{cases} \quad (\text{III.17})$$

Le vecteur d'état $x(t)$ est donné par :

$$x = [y \ D^{0.2}y \ D^{0.3}y \ D^{0.6}y \ D^{0.8}y]^T \quad (\text{III.18})$$

La dérivée d'ordre non entier du vecteur d'état est :

$$D^{(\tilde{\alpha})}(x) = [D^{0.2}x_1 \ D^{0.1}x_2 \ D^{0.3}x_3 \ D^{0.2}x_4 \ D^{0.2}x_5]^T \quad (\text{III.19})$$

Le modèle d'état correspondant au modèle transfert (III.14) est donné par :

$$\begin{cases} D^{(\tilde{\alpha})}(x(t)) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -3 & -0.9 & 0 & 0 & -0.2 \end{bmatrix} (x(t)) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = [2 \ 0 \ 1 \ 3 \ 0](x(t)) + 0u(t) \end{cases} \quad (\text{III.20})$$

Dans cette interface nous introduisons les coefficients et les ordres non entiers du numérateur et du dénominateur d'une fonction de transfert, il suffit de cliquer sur le bouton "Modèle d'état" et nous obtenons le modèle d'état commandable correspondant.

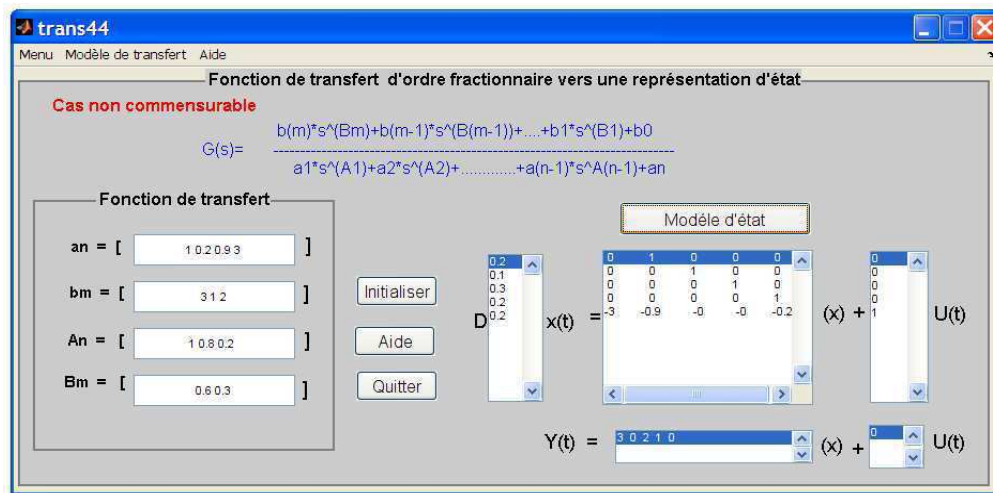


Figure III.18.a :L'interface du passage d'une fonction de transfert vers une représentation d'état (cas non commensurable)

Remarque :

Dans le cas où il existe des ordres du numérateur et ceux du dénominateur qui sont égaux cette interface effectue la simplification.

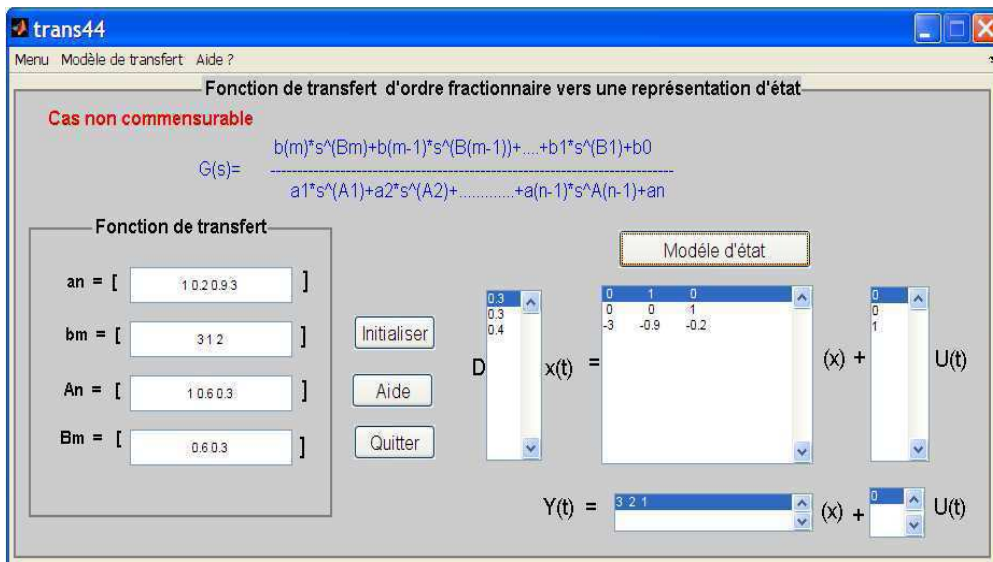


Figure III.18.b :L'interface du passage d'une fonction de transfert vers une représentation d'état (cas non commensurable)

VI.2.3. Passage d'une équation différentielle non entière vers une représentation d'état

Cette interface permet le passage direct d'une équation différentielle linéaire d'ordre fractionnaire à coefficients constants vers un modèle d'état non entier.

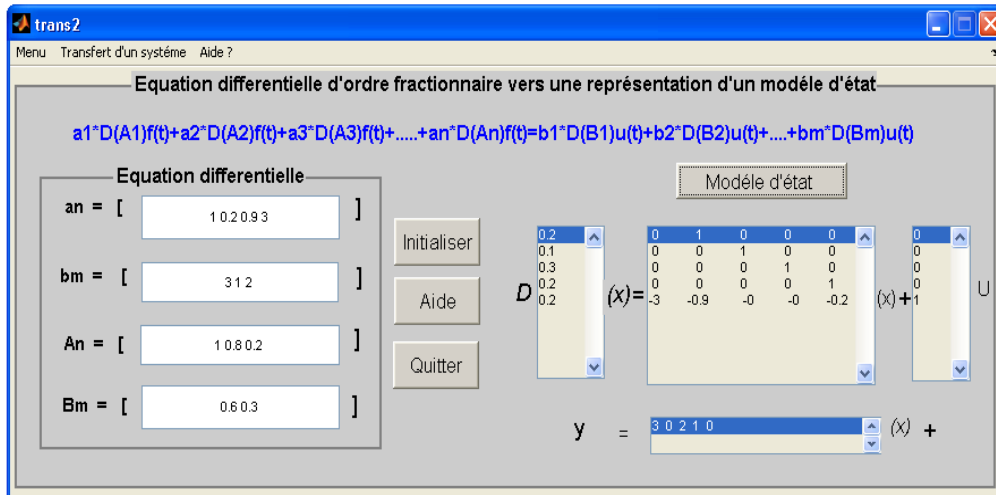


Figure III.19 : L'interface de passage d'une équation différentielle d'ordre fractionnaire vers une représentation d'état

VI.2.4. Passage d'un modèle d'état d'ordre non entier vers une fonction de transfert

❖ **Cas commensurable**

Nous introduisons les matrices et l'ordre non entier commensurable du modèle d'état suivant :

$$\begin{cases} D^{0.1}(x(t)) = \begin{bmatrix} 0 & 0 & -0.45 & -1.5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} (x(t)) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u(t) \\ y(t) = [1.5 \ 0 \ 0 \ 0] (x(t)) + 0u(t) \end{cases} \quad (III.21)$$

La fonction de transfert obtenue est donnée dans la figure (III.20):

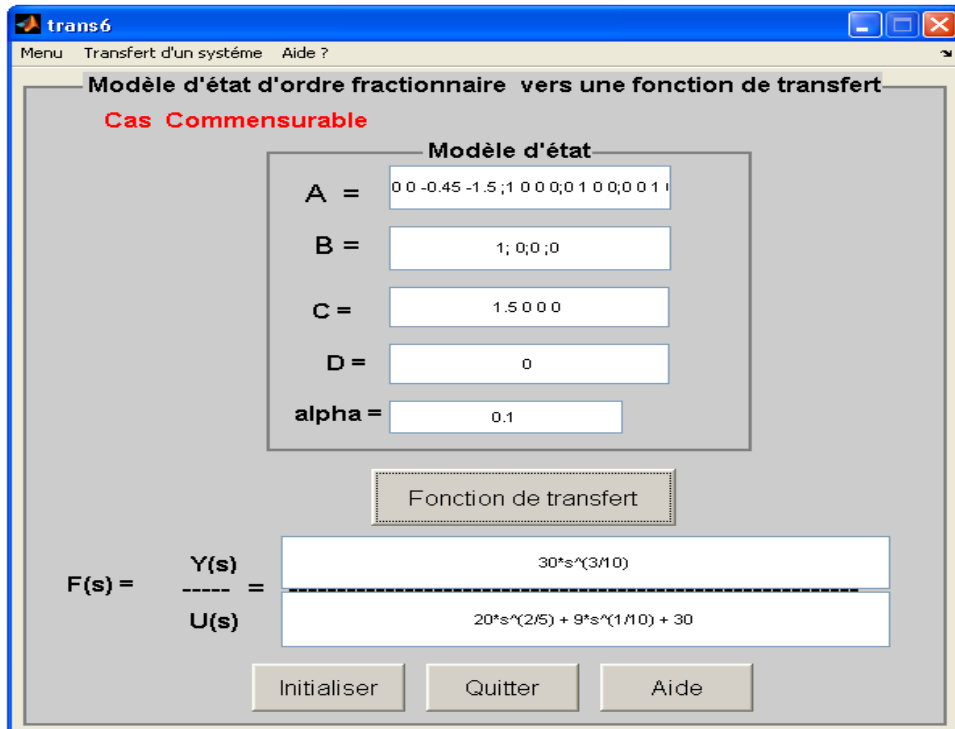


Figure III.20 : L’interface du passage d’une représentation d’état vers une fonction de transfert (cas commensurable)

❖ **Cas non commensurable**

Nous introduisons les matrices du modèle d’état et les ordres du vecteur d’état suivant :

$$\begin{cases}
 \begin{bmatrix} D^{0.2}x_1(t) \\ D^{0.1}x_2(t) \\ D^{0.3}x_3(t) \\ D^{0.2}x_4(t) \\ D^{0.2}x_5(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -3 & -0.9 & 0 & 0 & -0.2 \end{bmatrix} (x(t)) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(t) \\
 y(t) = [2 \ 0 \ 1 \ 3 \ 0](x(t)) + 0u(t)
 \end{cases} \tag{III.22}$$

La fonction de transfert obtenue est donnée dans (Fig.III.21):

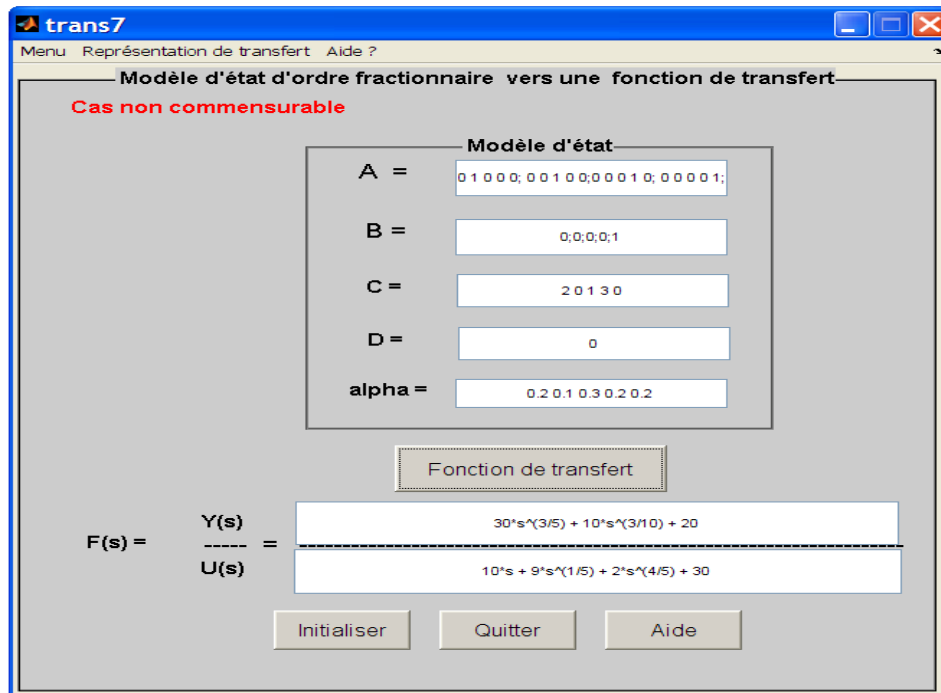


Figure III.21 : L'interface du passage d'une représentation d'état vers une fonction de transfert (cas non commensurable)

VI.3. Analyse et simulation des systèmes fractionnaires

Dans cette partie nous allons entamer l'analyse fréquentielle et temporelle des systèmes d'ordre fractionnaire.

La figure (III.22) représente l'interface de sélection pour l'analyse et la simulation des systèmes fractionnaire.

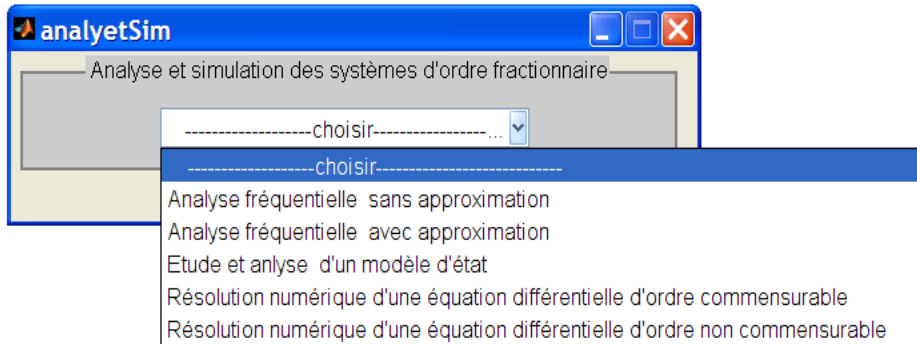


Figure III.22 : L'interface de sélection pour l'analyse et la simulation des systèmes d'ordre fractionnaire

VI.3.1. Analyse fréquentielle sans approximation

La réponse fréquentielle s'obtient par l'évaluation directe de la fonction de transfert, toutefois, on peut :

- Tracer le diagramme de Bode, Nyquist et Nichols.
- Etudier la stabilité des systèmes fractionnaires.
- Calculer les pôles et les zéros.
- Tracer la réponse indicielle et la réponse à une entrée donnée.

A titre d'illustration on introduit les coefficients et les ordres de la fonction de transfert fractionnaire (III.23), une bande de fréquences de $\omega = \text{logspace}(-4,4)$ pour le tracer les diagrammes, et l'intervalle de temps $[0 : 1 : 10]$ pour tracer les réponses.

$$G(s) = \frac{s^{0.4} + 1}{s^{1.4} + 1s^{0.4} + 1} \quad (\text{III.23})$$

Pour la fonction $G(s)$, on calcule les pôles et on trace le diagramme de Bode, les résultats obtenus sont donnés dans (Figure III.23)

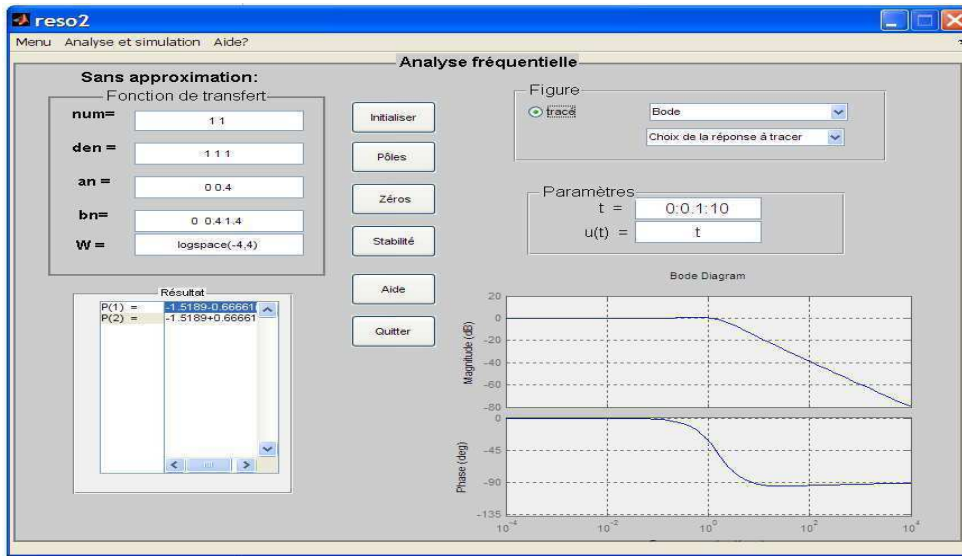


Figure III.23 : L'interface de l'analyse fréquentielle (Sans approximation)

VI.3.2 Analyse fréquentielle avec approximation

Pour synthétiser une transmittance entière à partir d'une transmittance non entière, on substitue l'opérateur de dérivation fractionnaire s^n par une des approximations développées dans le chapitre II. Dans notre cas on a utilisé l'approximation d'Oustaloup modifiée.

A titre d'illustration on introduit les coefficients et les ordres de la fonction de transfert fractionnaire (III.23), la taille de la bande de fréquences $le = \text{logspace}(-4,4)$, fréquence de croisement $\omega_c = 1 \text{ rad/sec}$, et l'intervalle de temps $[0 : 0.1 : 10]$ pour tracer les réponses.

$$G(s) = \frac{1}{s^{0.5} + 40} \tag{III.24}$$

L'analyse de la transmittance entière approximée se fait d'une manière classique. Les résultats de l'approximation sont donnés dans La figure (III.24).

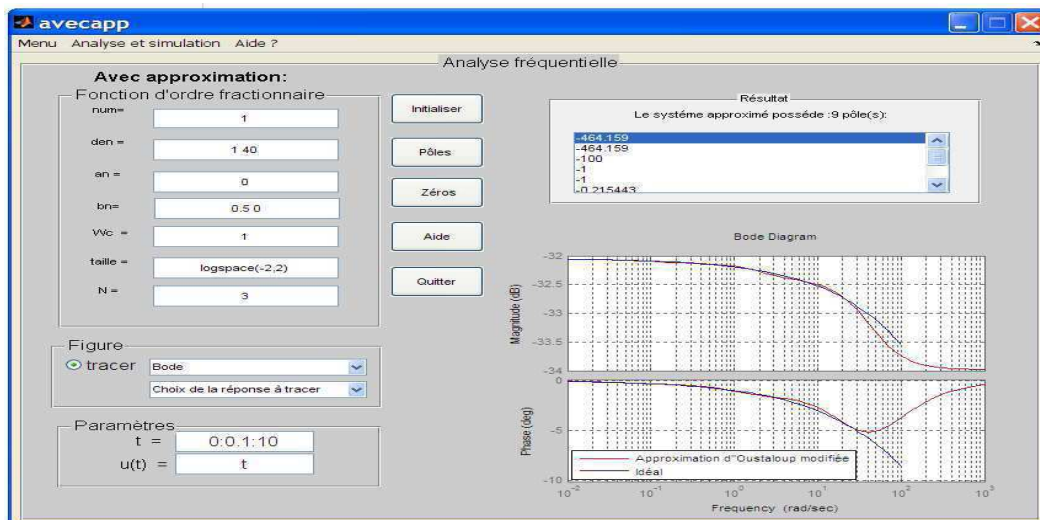


Figure.III.24 : L'interface d'analyse fréquentielle (avec approximation)

VI.3.3. Etude et analyse d'un modèle d'état fractionnaire

L'interface de la figure (III.25) permet l'analyse d'un modèle d'état fractionnaire commensurable en :

- Testant l'observabilité et la commandabilité
- Calculant l'équation caractéristique et ses pôles.
- Calculant la fonction de transfert.

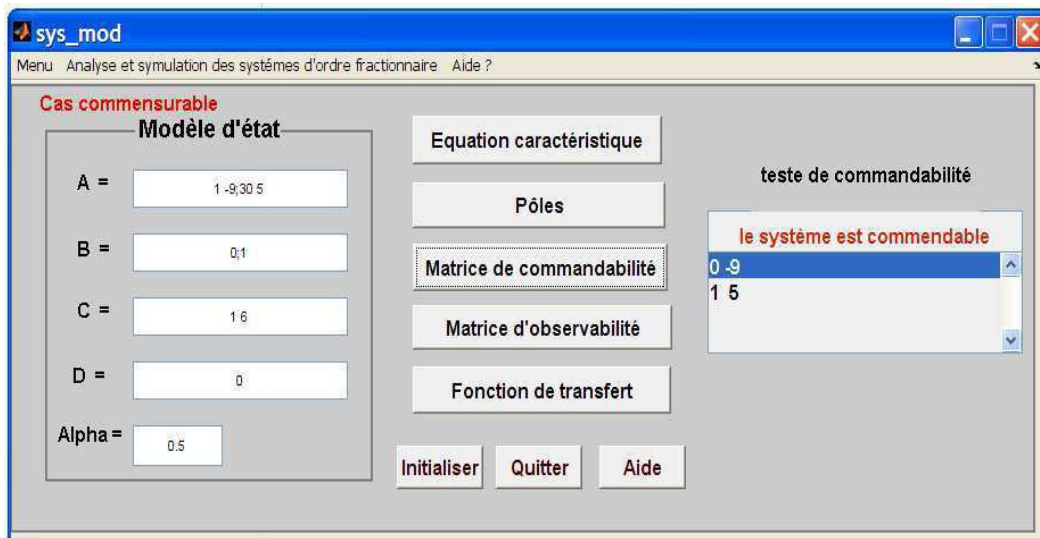


Figure.III.25 : L'interface de l'analyse d'un modèle d'état fractionnaire

VI.3.4. Résolution numérique d'une équation différentielle

La réponse temporelle du système linéaire fractionnaire, s'obtient par la résolution numérique de l'équation différentielle fractionnaire.

A titre d'illustration on simule la réponse temporelle de (III.25) pour une entrée échelon unitaire sur un intervalle de temps $t = [0 : 0.01 : 0.5]$

Soit l'équation différentielle :

$$D^{1/2}y(t) + 40y(t) = u(t) \quad , \quad (III.25)$$

❖ Cas commensurable

Pour la simulation de la sortie $y(t)$: introduire les coefficients, l'ordre commensurable, l'entrée $u(t)$ et l'intervalle du temps (pour l'ordre commensurable $\alpha = 1/q$ donné, introduire q).

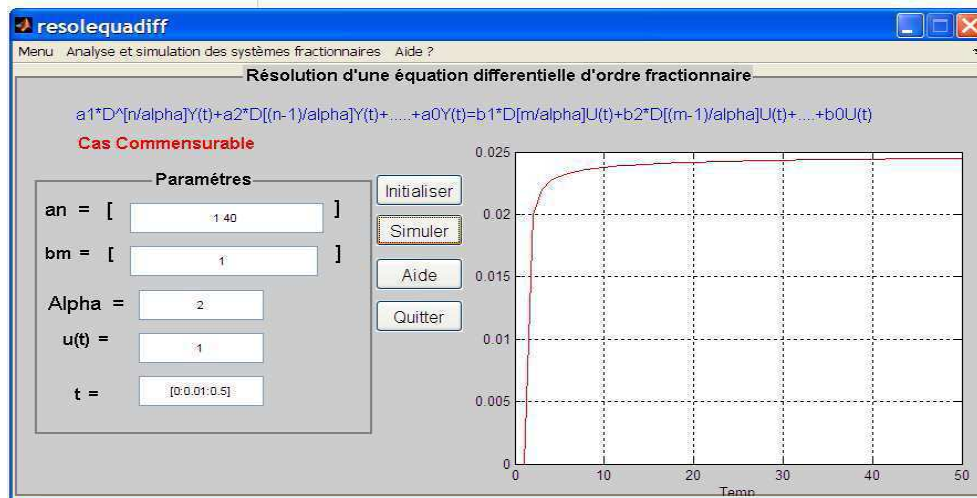


Figure III.26 : L'interface résolution d'une équation différentielle d'ordre commensurable

❖ Cas non commensurable

Pour la simulation de la sortie $y(t)$: introduire les coefficients, les ordres non entiers correspondants, l'entrée $u(t)$ et l'intervalle du temps

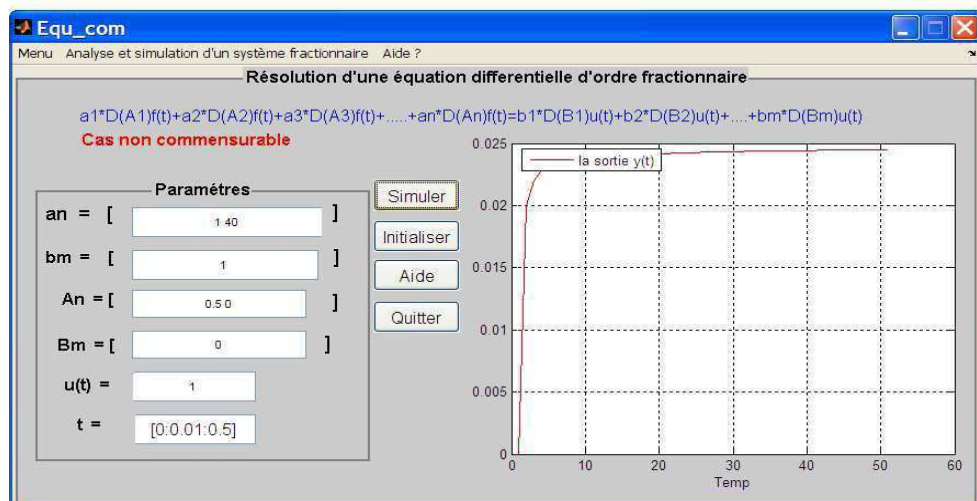


Figure III.27 : L'interface résolution d'une équation différentielle d'ordre non commensurable

VI.4. Méthodes d'approximation d'un dérivateur généralisé

Il existe plusieurs méthodes d'approximation, dans cette interface nous présentons quatre méthodes illustrées ci-dessous :

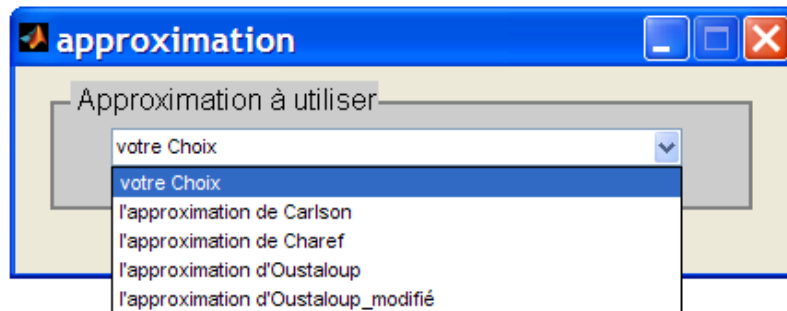


Figure.III.28 : L'interface de sélection pour les méthodes d'approximation d'un dérivateur généralisé

VI.4.1. Méthode de Carlson

La méthode de Carlson à été initialement développée pour approximer l'expression $\left(\frac{1}{s}\right)^{1/n}$ qui décrit le condensateur fractionnaire général pour tout entier $n > 1$, par une fonction de transfert d'ordre entier tout en spécifiant n , fréquence de croisement ω_c et le nombre d'itération N .

L'exemple présenté approxime la transmittance $H(s) = \left(\frac{1}{s}\right)^{1/3}$, pour $N = 2$ et $\omega_c = 1$,

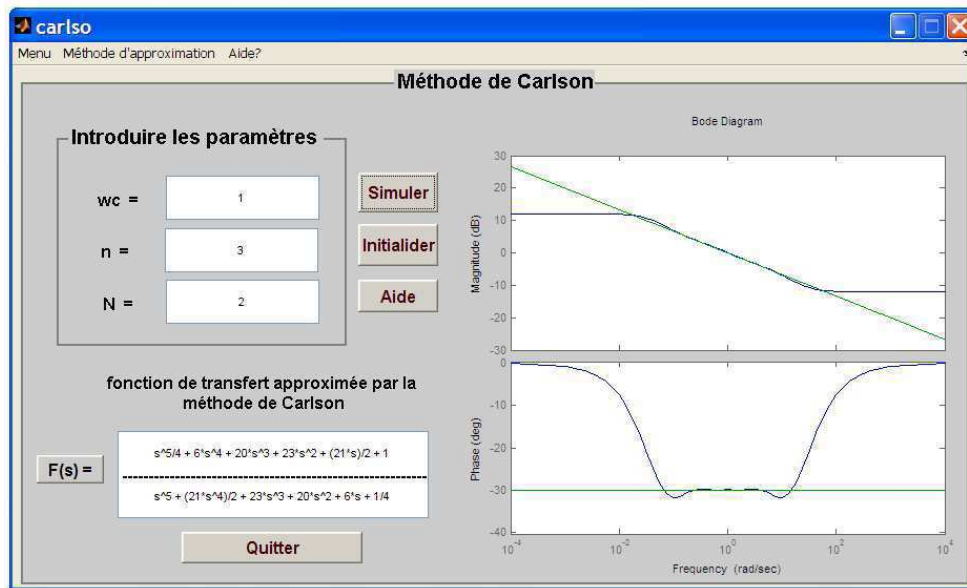


Figure III.29 : L'interface de l'approximation de Carlson

On constate que les tracés des diagrammes d'amplitude et de phase approximatés sont identiques au diagramme de Bode exacte de $H(s)$ sur un intervalle fréquentielle borné, avec une précision plus ou moins importante.

On a remarqué que pour un nombre d'itération supérieur à 2, pour $n = 10$, et supérieur à 4, pour $n = 2$, le temps de simulation est très lent et les résultats obtenus sont inexploitable, en raison de l'ordre très élevé de la transmittance entière.

VI.4.2. Approximation d'Oustaloup

Cette méthode consiste à remplacer le dérivateur généralisé $\left(\frac{s}{\omega_c}\right)^\alpha$ où α l'ordre non entier et ω_c fréquence de croisement, par un dérivateur borné en fréquence que l'on approxime ensuite par une transmittance entière.

L'interface de la méthode d'Oustaloup permet :

- De tracer le diagramme de Bode
- L'affichage de la fonction de transfert approximée d'ordre entier

Pour un intervalle fréquentielle $[\omega_A \quad \omega_B]$, introduire les paramètres :

$$\omega_c, \text{ la taille} = \frac{\omega_B}{\omega_A}, \alpha \text{ et } N.$$

Exemple :

$$\text{taille} = \frac{\omega_B}{\omega_A} = \frac{7.0711}{0.1414}, \omega_c = 1(\text{rad/sec}), N = 3 \text{ et } \alpha = 0.7$$

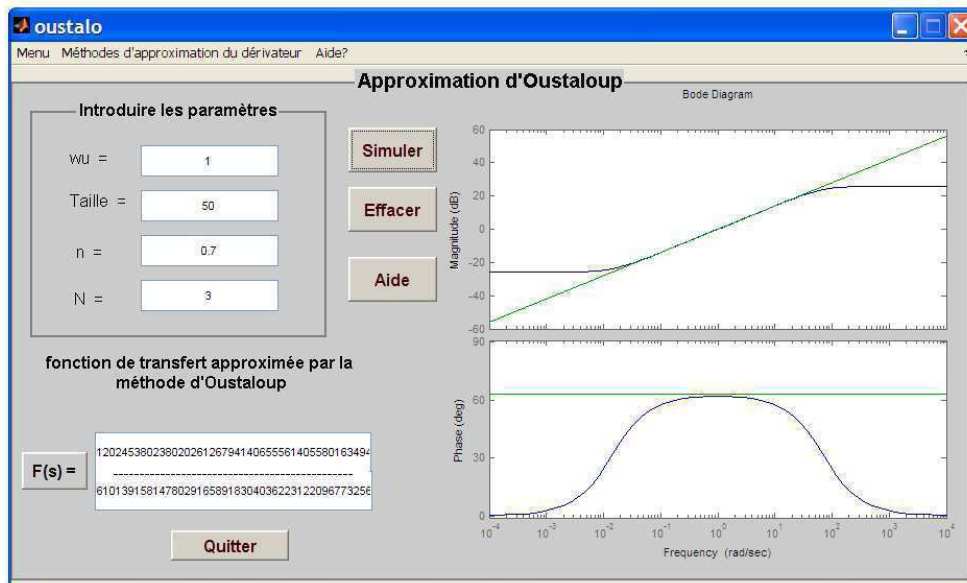


Figure III.30 : L'interface d'approximation d'Oustaloup

Remarque :

Pour plusieurs simulations de différentes valeurs de n on a remarqué que l'approximation est moins précise pour les valeurs de n proches de 0 et de 1

VI.4.3. Approximation de Charef

La fonction de transfert non entière $H(s) = \frac{1}{(1+\frac{s}{\omega_c})^n}$ peut être approximée par un transfert d'ordre entier en mettant en série plusieurs filtres passe bande. Charef utilise comme paramètre principal l'écart maximal, $\varepsilon > 0$ (en décibel)

Prenant un exemple pour un écart de $y = 0.1$ et un ordre non entier $n = 0.8$, une fréquence de coupure $\omega_u = 0.1$ et une bande de fréquences de $taille = 50$.

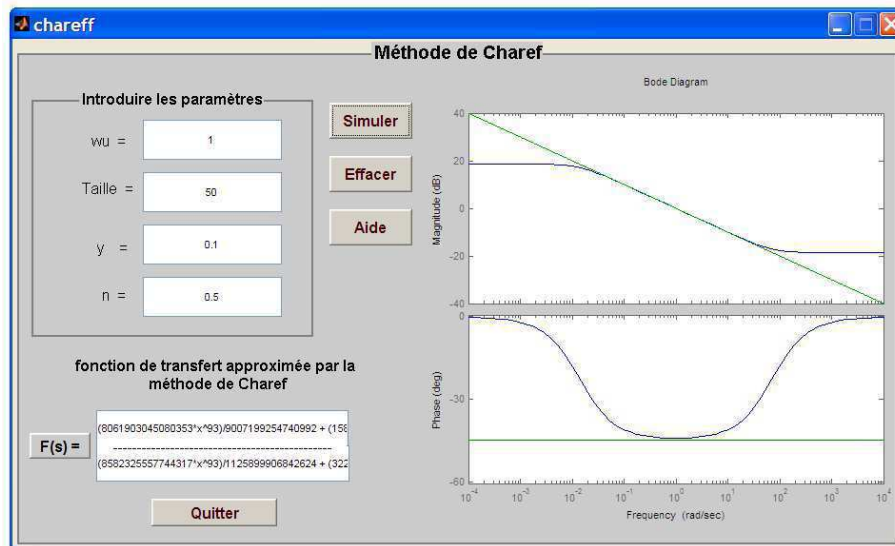


Figure III.31 : L'interface d'approximation de Charef

On constate que cette méthode est moins performante pour les valeurs n proche de 0.5.

VI.4.4. Approximation d'Oustaloup modifiée

Afin d'améliorer la méthode d'Oustaloup, il a été proposé d'extraire un filtre à partir de l'expression du dérivateur généralisé borné en fréquences, avant d'approximer ce dernier au moyen de l'algorithme d'Oustaloup.

Pour un intervalle fréquentielle $[\omega_A \quad \omega_B]$, introduire les paramètres:

$$\text{la taille} = \frac{\omega_B}{\omega_A}, \omega_c, \alpha \text{ et } N.$$

Nous illustrons cette méthode pour les paramètres suivants :

$$\text{taille} = \frac{\omega_B}{\omega_A} = 50, \omega_c = 1(\text{rad/sec}), N = 3 \text{ et } \alpha = 0.4$$

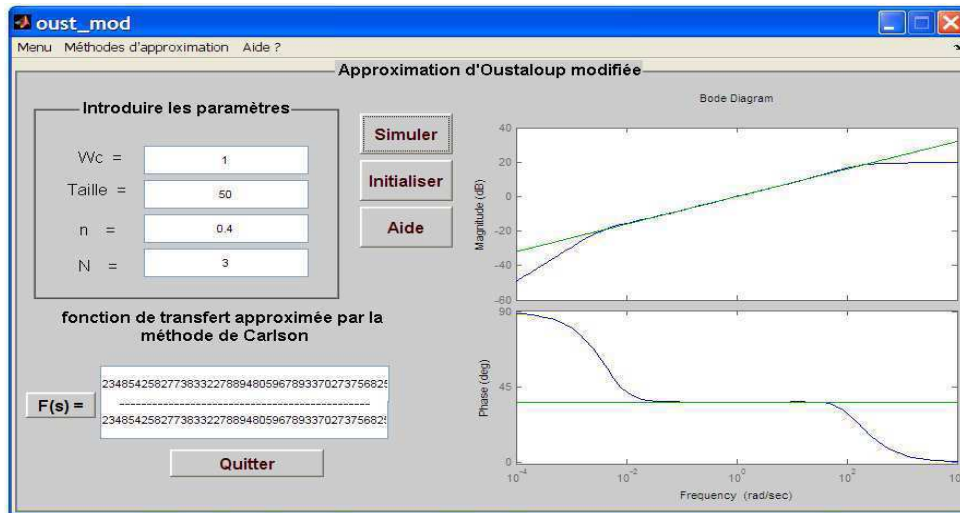


Figure III.32 : L'interface d'approximation d'Oustaloup modifiée

La méthode d'Oustaloup modifiée est précise sur un plus large intervalle d'approximation.

En fin, on remarque que les courbes d'approximation (figures III.29, 30, 31,32) et celles du modèle exact sont confondues autour de la fréquence de croisement $\omega_c = 1 \text{ rad/sec}$ mais divergent au fur et à mesure qu'on s'éloigne, en hautes fréquences en particulièrement.

VI.5. Présentation de l'aide

Dans l'intérêt de faciliter l'utilisation de notre logiciel, nous l'avons accompagné d'une aide disponible lors du contact avec l'interface principale. Cliquer sur « Aide » dans la figure (III.8) pour y accéder.

A propos du guide l'introduction des paramètres dans chacune des interfaces présentées, on dispose aussi d'un bouton « Aide » spécifié à l'interface qui le constitue. A titre d'illustration nous présentons la figure(III.33).

Le logiciel contient des messages d'erreurs (par exemple la figure (III.34)), qui apparaissent lorsqu'une erreur d'introduction des paramètres est commise.

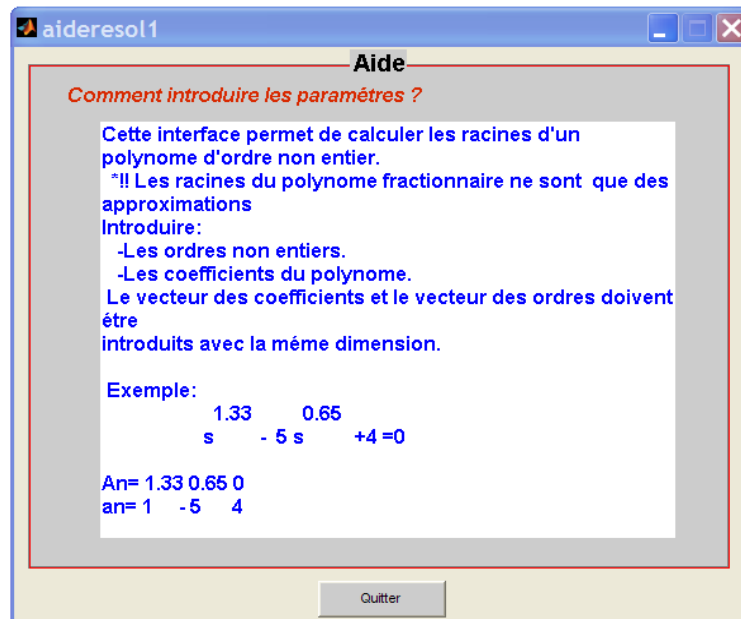


Figure III.33 : Exemple d'interface de l'aide



Figure III.34 : Exemple d'interface d'un message d'erreur

VII. Conclusion

Afin d'illustrer notre logiciel, ce chapitre a exposé d'une manière explicite et simple l'outil de création d'interface graphique « GUIDE » de Matlab, puis nous avons présenté les différentes parties du logiciel avec des exemples d'application.

En fin nous espérons que ce logiciel soit bénéfique pour l'analyse et la simulation des systèmes dynamiques d'ordre fractionnaire.

Conclusion
générale

La dérivation non entière est un outil de plus en plus utilisé pour la description mathématique des phénomènes physiques, en particulier ceux à comportement diffusif et à longue mémoire et aussi pour la conception des régulateurs robustes dans une chaîne d'asservissement.

L'objectif de notre travail a été de développer un logiciel sous Matlab d'aide à l'analyse et la simulation des systèmes d'ordre non entier décrits au moyen de la dérivation non entière.

Le logiciel que nous avons réalisé contient :

- Les opérations de dérivation et d'intégration fractionnaire.
- La résolution et la simulation d'une équation différentielle d'ordre fractionnaire.
- La résolution d'une équation caractéristique d'ordre fractionnaire.
- Les différents passages entre les modèles (modèle d'état, fonction de transfert, équation différentielle).
- Le problème d'approximation des modèles non entiers est pris en charge par le logiciel réalisé, plusieurs méthodes sont programmées.

Ces applications sont accompagnées par un support graphique basé sur les instructions de tracés déjà disponibles dans Matlab.

Le logiciel que nous avons réalisé permet donc de :

- ✓ Simuler un système d'ordre fractionnaire.
- ✓ Tester la stabilité d'une fonction de transfert.
- ✓ Tester la commandabilité et l'observabilité.
- ✓ Calculer les différentes approximations.

Certains problèmes ont été rencontrés et n'ont pas pu être résolus comme : le calcul de la réponse d'un modèle d'état, le calcul de la fonction de Mittag-Leffler (matrice de transition) et la stabilité étudiée est limitée au cas commensurable.

Le logiciel réalisé n'est qu'une version préliminaire qui nécessite des améliorations, il serait intéressant de pouvoir réaliser une Toolbox complète sous Matlab pour l'étude des systèmes d'ordre fractionnaire.

Vue la modularité de notre logiciel, celui-ci peut être amélioré, modifié et complété par des ajouts, comme perspectives, nous envisageons :

- L'extension aux systèmes multivariables.
- Développement des outils pour la construction des diagrammes fonction pour les systèmes fractionnaires sous Simulink.
- Insertion des méthodes d'identification et des régulateurs d'ordre fractionnaire.

Bibliographie

- [1] Abdennebi T Epouse Djamah. (2009), "*Identification des systèmes fractionnaires par des modèles d'ordre fractionnaire*", Mémoire de Doctorat, Université Mouloud Mammeri de Tizi-Ouzou.
- [2] Ait Messaoud L. (2007), "*Contribution à la commande des systèmes par des régulateurs d'ordre non entier. Application à la commande de la machine asynchrone*". Mémoire de Magister, Université Mouloud Mammeri de Tizi-Ouzou.
- [3] Canat. Sylvain. (2005), "*Contribution à la modélisation dynamique d'ordre non entier de la machine asynchrone à cage*", Mémoire de Doctorat, l'institut national polytechnique de Toulouse.
- [4] Carlson G.E et Halijak C. "Approximation of Fractional Capacitors $(1/s)^{1/n}$ by a Regular Newton Process". *IEEE Transactions on circuits and systems*, Volume **11**, No 2. 1964 .pp. 210-213
- [5] Chyi Hwang, Jeng-Fan Leu, et Sun-Yuan Tsay., "Technical Notes and Correspondence: A Note on Time-Domain Simulation of Feedback Fractional-Order Systems", *IEEE Transactions on Automatic Control*, vol. **47**, No. 4, Avril 2002. pp. 625-631.
- [6] Cois. Olivier. (2002), "*Systèmes linéaires non entiers et Identification par modèle non entier : Application en thermique*", Mémoire de Doctorat, l'université Bordeaux 1.
- [7] Djouambi. A, Besançon. A. V et Charef. A. "*Identification Réursive des Systèmes à Dérivée Non Entière*" .Journées Identification et Modélisation Expérimentale JIME'2006 – 16 et 17 novembre – Poitiers <http://www.lag.ensieg.inpg.fr>
- [8] Diethelm. K et Freed. Alan D., "The FracPECE Subroutine for the Numerical Solution of Differential Equations of Fractional Order". In *Forschung und wissenschaftliches Rechnen* 1998, S. Heinzl and T. Plessner, eds, no 52 in GWDG-Berichte, Gottingen, 1999, Gesellschaft fur wissenschaftliches. pp. 57-71

- [9] Diethelm K., "An algorithm for the numerical solution of differential equations of fractional order". *Elec. Transact. Numer. Anal.* Volume **5**, 1997. pp. 1–6. Kent State University. ISSN 1068-9613.
- [10] Diethelm K., Walz G. (1997). " Numerical solution of fractional order differential equations by extrapolation". *Numer. Algorithms* **16**, 231–253.
- [11] Diethelm K., Freed A. D. "Viscoelastic/viscoplastic material modeling using the Fractional calculus". *Submitted for publication*. 1998.
- [12] Diethelm K., Neville J.Ford. " Numerical solution of fractional linear and non linear fractional differential equations involving fractional derivatives of several order ". *ISSN 1360-1725 Numerical Analysis Report No.379*. February 21,2001.
- [13] Dominik Sierociul., "Fractional Order Discrete State-Space System Simulink Toolkit User Guide".
- [14] François Dubois, Ana-Cristina Galucio et Nelly Point., " Introduction à la dérivation fractionnaire Théorie et applications". Avril 2009.
- [15] Frederick Dean K., "*Feedback control problems using systems Toolbox*", New York Publishing company. 2000.
- [16] Hartley T.T., Lorenzo C.F., "A solution to the Fundamental Linear Fractional Order Differential Equation", *NASA report*, 1998.
- [17] Henslman, Duane et littlefield Bruce., "Mastring Matlab 7". *London Prentice Hall*. 2005.
- [18] LADACI S., CHAREF A, LOISEAU J.J.," Robust Fractional Adaptive Control based on the strictly positive realness condition" *Int. J. Appl. Math. Comput. Sci.*, 2009, Vol. **19**, No. 1, pp. 69–76 DOI: 10.2478/v10006-009-0006-6
- [19] Lapresté et J.H. "*Introduction à Matlab*". Paris. Ellipse. 1999.

- [20] Lixia.Yuan, Om P. Agrawal., "A Numerical Scheme for Dynamic Systems Containing Fractional Derivatives". ASME Design Engineering Technical Conferences. Proceedings of Detc'98. Septembre, 1998, Atlanta, Georgia.
- [21] Mainardi. F,et Gorenflo R,"Fractional diffusion processes: probability distributions and continuous time random walk" in: G. Rangarajan et M. Ding (Editors), *Processes with Long Range Correlations*, Springer-Verlag, Berlin (2003), pp. 148-166. [Lecture Notes in Physics, No. 621]
- [22] Mansouri R, (2008), "*Contribution à l'analyse et la synthèse des systèmes d'ordre fractionnaire par la représentation d'état*". Mémoire de Doctorat, Université Mouloud Mammeri de Tizi-Ouzou.
- [23] Miller K.S et B. Ross. "*An introduction to the fractional calculus and fractional differential equations*", A Wiley-Interscience Publication, 1993.
- [24] Nicola G, Genevière D, Laurent L. "*Approximation d'un modèle d'ordre non entier*".
- [25] Neville J. Ford et Charles Simpson. " Numerical and analytical treatment of differential equations of fractional order ". *ISSN 1360-1725 Numerical Analysis Report* No.387. February 22, 2001.
- [26] Oustaloup .A. "*Systèmes Asservis linéaires d'ordre fractionnaire*", Editions MASSON, Paris, 1983.
- [27] Oustaloup. A. " *La dérivation Non Entière : Théorie, Synthèse et application*". Edition HERMES, Paris, 1995.
- [28] Oldham K.B. et Spanier. J. "*The fractional calculus*". Academic Press, New York and-London, 1974.
- [29] Podlubny. I. "*Fractional Differential Equations : An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution*". Academic Press, San Diego, 1999.

- [30] Podlubny I., Dorcak L., Kostial I. "On Fractional Derivatives, Fractional-order Dynamic systems and $PI^\lambda D^\mu$ -controllers", *IEEE; Proceeding of 36th Conference on Decision & Control*. pp. 4985–4990. San Diego, California USA. Decembre, 1997.
- [31] Poinot T, Trigeassou J-C. "A method for modelling and simulation of fractional systems". *Signal Processing*. Volume **83**, 2003. pp.2319-2333.
- [32] Rivoire M et Ferrier M."*Matlab, Simulink, Stateflow avec des exercices d'automatique resolu*". Paris. Techniq. 2001.
- [33] Sabatier. J., Agrawal. O. P et Tenreiro. J. A. Machado. "*Advances in Fractional Calculus Theoretical Developments and Applications in Physics and Engineering*" Springer, The Netherlands. Published by Springer, 2007.
- [34] Shantanu.Das, "*Functional Fractional Calculus for System Identification and Controls*" Springer, Verlag Berlin Heidelberg 2008
- [35] Valério. Duarte. (2005), "*Fractional Robust System Control*", mémoire de Doctorat, Universidade técnica de lisboa.
- [36] Xue. Dingyü et Chen. Yang. Quan, and Atherton, Derek P.," *Linear Feedback Control: Analysis and Design with MATLAB*", the Society for Industrial and Applied Mathematics. Philadelphia, 2007.
- [37] Yang Q C, Ivo Petras et Blas Vinagre. "A List of Laplace and Inverse Laplace Transforms Related to Fractional Order Calculus". November 2001.

Annexe A

ANNEXE A

Outils mathématiques de base

A.1. La fonction Gamma [29]

Une des fonctions de base du calcul fractionnaire est la fonction Gamma d'Euler, qui généralise la fonction factorielle $n!$ et permet à n de prendre également des valeurs réelles et même des valeurs complexes. Nous rappelons quelques résultats sur la fonction Gamma qui sont utiles dans le calcul fractionnaire.

A.1.1. Définition

La fonction Gamma est définie par :

$$\Gamma(\alpha) = \int_0^{\infty} e^{-t} t^{\alpha-1} dt \quad \alpha \in \mathbb{R} \quad (\text{A.1})$$

A.1.2. Propriétés

1. Une des propriétés de base de la fonction Gamma satisfait l'équation :

$$\Gamma(\alpha + 1) = \alpha \Gamma(\alpha) \quad (\text{A.2})$$

preuve:

$$\Gamma(\alpha + 1) = \int_0^{\infty} e^{-t} t^{\alpha} dt = [-e^{-t} t^{\alpha}]_0^{\infty} + \alpha \int_0^{\infty} e^{-t} t^{\alpha-1} dt = \alpha \Gamma(\alpha) \quad (\text{A.3})$$

2. $\Gamma(n + 1) = n! \quad n \in \mathbb{N} \quad (\text{A.4})$

preuve:

On déduit de (A.1) que : $\Gamma(1) = 1$

Et en utilisant l'expression (A.2), on aura pour $\alpha = 1, 2, 3, \dots, n + 1$

$$\begin{aligned}
\Gamma(2) &= 1, \Gamma(1) = 1 = 1! \\
\Gamma(3) &= 2, \Gamma(2) = 2.1! = 2! \\
\Gamma(4) &= 3, \Gamma(3) = 3.2! = 3! \\
&\vdots \\
\Gamma(n+1) &= n, \Gamma(n) = n.(n-1)! = n!
\end{aligned}
\tag{A.5}$$

A.2. Mittag-Leffler [29,23]

La fonction exponentielle e^t , joue un rôle important dans la théorie des équations différentielles d'ordre non entier et notée par :

$$E_\alpha(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + 1)} \tag{A.6}$$

La fonction Mittag-Leffler à deux paramètres est définie par la série exponentielle :

$$E_{(\alpha,\beta)}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + \beta)} \quad (\alpha > 0, \beta > 0) \tag{A.7}$$

Pour $\alpha = \beta = 1$ on a :

$$E_{1,1}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(k+1)} = \sum_{k=0}^{\infty} \frac{t^k}{k!} = e^t \tag{A.8}$$

Annexe B

ANNEXE B

Méthodes d'approximations du dérivateur généralisé

B.1. Méthode de Carlson

Cette méthode permet d'approximer la fonction $H(s) = \left(\frac{1}{s}\right)^{1/n}$ pour tout entier $n > 1$, ce problème d'approximation a été formulé sous forme d'une équation ($f(x) = x^n - a$) non linéaire monovariante résolue par le processus itératif de Newton *telque* $f(x) = 0$

D'où résulte que :

$$x = a^{1/n} \quad (\text{B.1})$$

$$\text{Avec } \begin{cases} a = 1/s \\ s = j\omega \end{cases}$$

- Le principe de la méthode

Le processus de Newton est la méthode usuelle de la résolution numérique des équations non linéaire monovariante. Il consiste à mettre l'expression $f(x) = 0$ sous une forme $x = F(x)$, le processus itératif étant défini, pour une estimée initiale x_0 de la solution :

$$x_{i+1} = F(x_i) \quad (\text{B.2})$$

Jusqu'à ce que la séquence $\{x\}_{i=0}^{\infty}$ converge vers la solution exacte notée r ,

$$(f(r) = 0)$$

La fonction $F(x)$ est donné par :

$$F(x) = x - \frac{f}{f'} \quad (\text{B.3})$$

Et sa première dérivée donne :

$$F'(x) = \frac{ff''}{(f')^2} \quad (\text{B.4})$$

La convergence du processus est assuré par :

$$|x_{i+1} - x_i| < |x_i - x_{i-1}| \quad (\text{B.5})$$

Avec
$$x_{i+1} - x_i = F(x_i) - F(x_{i-1}) \quad (\text{B.6})$$

Ce qui veut dire que l'erreur à l'itération i est inférieure à celle à l'itération $(i - 1)$

Le développement en série de Taylor de (B.5) donne :

$$x_{i+1} - x_i = (x_i - x_{i-1})F'(\xi_i) \quad x_i < \xi_i < x_{i-1} \quad (\text{B.7})$$

-La condition nécessaire de convergence est assuré lorsque $|F'(x)| < 1$

-La régularité du processus est assuré si $F'(x)$ ou bien ff'' ne change pas de signe au voisinage de la solution et donc $a^{1/n}$ doit être un point d'inflexion ;

$$f(a^{1/n}) = f'(a^{1/n}) = 0 \quad (\text{B.8})$$

Cette condition ne pouvant pas être réaliser par la fonction $f(t)$ telle que définie par l'équation (B.1), il a été proposé de la transformé comme suit :

$$f(x) = \frac{x^n - a}{x^m}, \quad 0 < m < n - 1, m \in \mathbb{N} \quad (\text{B.9})$$

Cependant on montré que la condition (B.7) n'est valable que pour les ordres impaire de n , ($n = 2m + 1$) et l'équation devient de ce fait :

$$f(x) = \frac{x^{2m+1} - a}{x^m} \quad (\text{B.10})$$

Pour les n paires ($n = 2m$), les auteurs propose la formule :

$$f(x) = \frac{x^{2m} - a}{x^{m-1}} + \lambda \frac{x^{2m} - a}{x^m}, \quad (\text{B.11})$$

La racine λ n'étant considérée égale à x qu'à la fin du calcul de $F(x)$ qui est obtenu pour n impair et n pair, en remplaçant respectivement (B.10) et (B.11) dans (B.3), on aura la même formule pour les deux cas :

$$F(x) = x \frac{(n-1)x^n + (n+1)a}{(n+1)x^n + (n-1)a} \quad (\text{B.12})$$

En remplaçant dans (B.1) $F(x)$ et a par leurs expressions, on obtient :

$$x_{i+1} = x_i \frac{(n-1)x_i^n s + (n+1)}{(n+1)x_i^n s + (n-1)} \quad (\text{B.13})$$

Qui peut s'écrire aussi comme suit :

$$x_i = x_{i-1} \frac{(n-1)x_{i-1}^n s + (n+1)}{(n+1)x_{i-1}^n s + (n-1)} \quad (\text{B.14})$$

B.2. Méthode d'Oustaloup

L'approximation d'Oustaloup réduit le dérivateur généralisé pour la synthèse réaliste, à un nombre fini de pôles et de zéros sur un intervalle fréquentiel borné, choisi selon les besoins de l'application, ainsi que le transfert résultant est ensuite approximé par une transmittance d'ordre entier qui est donnée par :

$$D_{gen}(s) = \left(\frac{s}{\omega_c} \right)^\alpha, \alpha \in \mathbb{R} \quad (\text{B.15})$$

ω_c est la fréquence au gain unité ou fréquence de coupure, α est l'ordre non entier de dérivation. Une troncature à la fois du côté des basses fréquences et des hautes fréquences consiste à limiter sur l'intervalle fréquentiel $[\omega_A \omega_B]$, centré géométriquement sur ω_c , le comportement différentiel du transfert $\frac{s}{\omega_c}$. En fait, la troncature sera réellement effectuée, pour plus de précision, sur un intervalle de fréquence plus large $[\omega_b \omega_h]$ tel que :

$$\omega_b \ll \omega_A \text{ et } \omega_h \gg \omega_B \quad (\text{B.16})$$

On remplace alors le transfert fractionnaire par le transfert borné en fréquence donné par :

$$D_{borné}(s) = \left(C_0 \frac{1+\frac{s}{\omega_b}}{1+\frac{s}{\omega_h}} \right)^\alpha \quad (\text{B.17})$$

Avec :

$$C_0 = \frac{\omega_b}{\omega_c} = \frac{\omega_c}{\omega_h} \quad (\text{B.18})$$

Pour assurer un gain unitaire à la fréquence ω_c .

Le dérivateur borné en fréquences, étant lui aussi un transfert fractionnaire, doit être approximé par une transmittance d'ordre entier, construite au moyen des zéros et de pôles distribués récursivement :

$$D_{borné}(s) \rightarrow \lim_{N \rightarrow \infty} D_N(s) \quad (\text{B.19})$$

$$D_N(s) = \left(\frac{\omega_c}{\omega_h} \right)^\alpha \prod_{k=-N}^N \frac{1+\frac{s}{z_k}}{1+\frac{s}{p_k}} \quad (\text{B.20})$$

z_i et p_i ce sont respectivement le zéros et le pôle de rang i , ces singularités (paires zéro-pôle) au nombre total de $(2N + 1)$ sont distribuées récursivement comme suit :

$$\left\{ \begin{array}{l} \frac{p_i}{z_i} = \delta > 0, \quad \frac{z_{i+1}}{p_i} = \eta > 0 \end{array} \right. \quad (\text{B.21})$$

$$\left\{ \begin{array}{l} \frac{z_{i+1}}{z_i} = \frac{p_{i+1}}{p_i} = \delta\eta > 1 \end{array} \right. \quad (\text{B.22})$$

Les rapports δ et η qui impliquent un rapport constant $\delta\eta$ entre deux zéros ou deux pôles consécutifs, sont appelés « facteurs récursifs » l'approximation est réalisée par un lissage des diagrammes asymptotiques de Bode de $D_N(s)$ pour conduire à ceux de $D_{borné}(s)$

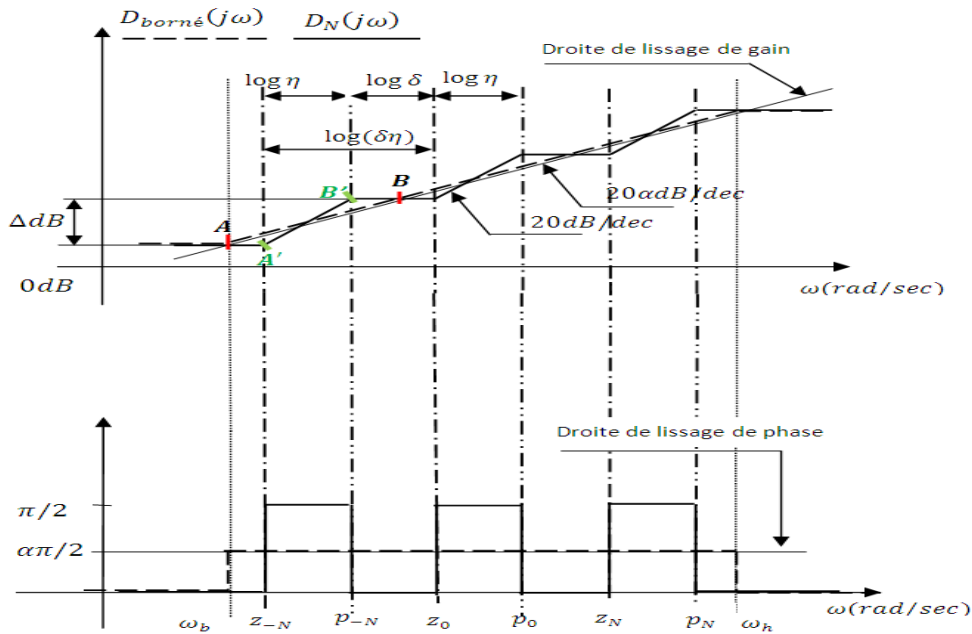


Figure B.1. Diagramme asymptotique de Bode de $D_{borné}(s)$ et de $D_N(s)$.

Le lissage des marches d'escalier qui constituent le diagramme asymptotique de gain de $D_N(s)$ entre ω_b et ω_h (figure B.1) qui peut être normalisé par une droite appelée «droite de lissage de gain», de pente $20\alpha \text{ dB/dec}$ avec α compris entre 0 et 1 ; cette droite supporte le diagramme de gain de $D_{borné}(s)$ et coupe le palier de longueur $\log(\alpha)$ en son milieu. En désignant par la hauteur de la marche d'escalier, les pentes de la droite.

$\frac{\alpha\pi}{2}$ conduit au même résultat donné par l'équation (B.25).

Comme l'asymptotique du gain de $D_{borné}(s)$ entre ω_b et ω_h coupe les marches d'escalier de $D_N(s)$ au milieu, les fréquences extrémales z_{-N} et p_N se situent respectivement à $1/2 \log(\eta)$ de ω_b et ω_h sur l'échelle logarithmique, soit alors :

$$\log(z_{-N}) - \log(\omega_b) = \frac{1}{2} \log(\eta) \Rightarrow z_{-N} = \eta^{1/2} \omega_b \quad (\text{B.26})$$

Et

$$\log(\omega_h) - \log(p_N) = \frac{1}{2} \log(\eta) \Rightarrow p_N = \eta^{-1/2} \omega_h \quad (\text{B.27})$$

Donc le rapport entre les fréquences transitionnelles ω_b et ω_h peut être exprimé à partir de (1) et (0) comme suit :

$$\frac{\omega_h}{\omega_b} = (\delta\eta)^{2N+1} \quad (\text{B.28})$$

D'où l'on tire le produit des facteurs récursif δ et η :

$$\delta\eta = \left(\frac{\omega_h}{\omega_b}\right)^{1/2N+1} \quad (\text{B.29})$$

Etant donné :

$$z_i = (\delta\eta)z_{i+1} \quad (\text{B.30})$$

B.3. Méthode de Charef

La méthode d'approximation approxime les systèmes fractals qui sont représentés par une transmittance de la forme :

$$H(s) = \frac{1}{s^\alpha} \quad (\text{B.31})$$

Ou $s = j\omega$ est la fréquence complexe et $\alpha \in \mathbb{R}$, est la dimension fractale. Généralement les systèmes fractals présentent une amplitude finie en basse fréquences, une représentation par un pôle à puissance fractionnaire est de ce fait plus appropriée :

$$H(s) = \frac{1}{(1 + \frac{s}{\omega_c})^\alpha} \quad (\text{B.32})$$

ω_c est la fréquence de cassure $0 < \alpha < 1$. La figure (Figure B.2) montre le diagramme asymptotique d'amplitude de Bode de la réponse en fréquence de $H(s)$. La droite de

pende $-20\alpha \text{ dB/dec}$ est approximée par une ligne en zigzag de pentes 0 dB/dec et -20 dB/dec . En alternance, ces pentes représentent les contributions individuelles des singularités : pôles p_i et zéros z_i .

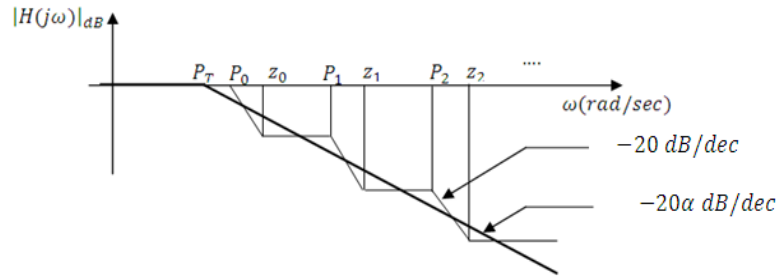


Figure B.2 : Diagramme asymptotique d'amplitude de Bode de $H(s)$ et son approximation.

L'approximation représentée dans la figure (B.2) se traduit par l'écriture suivante :

$$H(s) = \frac{1}{(1+\frac{s}{p_T})^\alpha} = \lim_{N \rightarrow \infty} \frac{\prod_{i=0}^{N-1} (1+\frac{s}{z_i})}{\prod_{i=0}^N (1+\frac{s}{p_i})}; \quad (\text{B.33})$$

Pour que le transfert recherché soit de dimension finie le nombre N doit être également fini, l'équation (B.33) prend ainsi la forme suivante :

$$H(s) = \frac{1}{(1+\frac{s}{p_T})^\alpha} \approx \frac{\prod_{i=0}^{N-1} (1+\frac{s}{z_i})}{\prod_{i=0}^N (1+\frac{s}{p_i})} \quad (\text{B.34})$$

Pour laquelle il faut déterminer les $2N+1$ pôles et zéros. Pour ce fait, Charef utilise comme paramètre principal l'écart maximal ε (en dicebel), entre la ligne d'approximation, (en zigzag) et la droite de pente $-20 \alpha \text{ dB/dec}$ (figure B.3.a), les pôles et les zéros seront ensuite obtenus par un simple calcul géométrique.

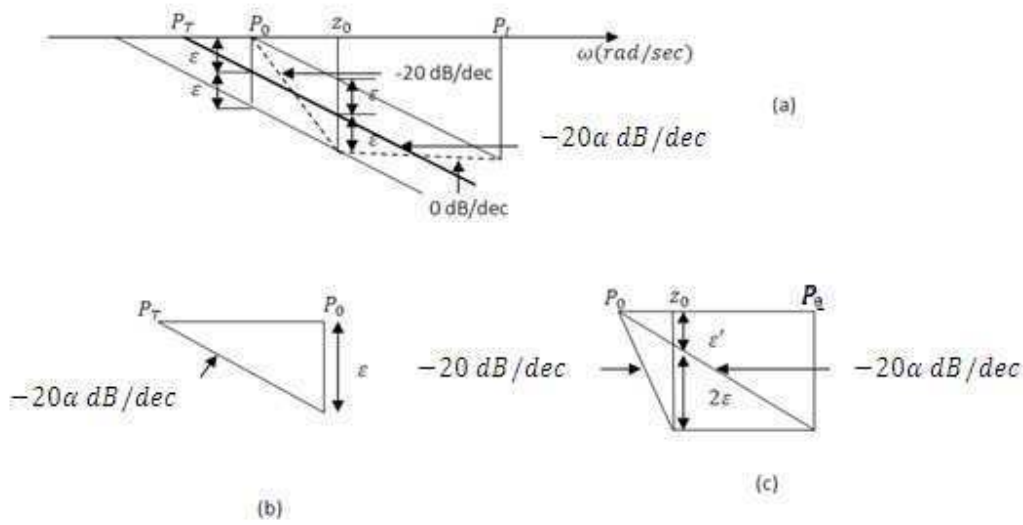


Figure B.3 : Principe du calcul des singularités

L'axe des abscisses ayant une échelle logarithmique, selon la figure (B.3.c) on a :

$$20m = \frac{\varepsilon}{\log(p_0) - \log(p_T)} \Rightarrow \varepsilon = 20m \log\left(\frac{p_0}{p_T}\right) \quad (\text{B.35})$$

Et selon la figure(B.3.c) on a :

$$20 = \frac{\varepsilon' + 2\varepsilon}{\log(p_0) - \log(p_T)} \quad (\text{B.36})$$

Sachant que :

$$20m = \frac{\varepsilon'}{\log(z_0) - \log(p_0)} \Rightarrow \varepsilon' = 20m \log\left(\frac{z_0}{p_0}\right) \quad (\text{B.37})$$

En remplaçant l'équation (B.37) dans (B.36) on obtient après quelques transformations :

$$z_0 = p_0 10^{\varepsilon/10(1-m)} \quad (\text{B.38})$$

Toujours selon la figure on a :

$$20m = \frac{2\varepsilon}{\log(p_1) - \log(z_0)} \Rightarrow \varepsilon = 10 \log \left(\frac{p_1}{z_0} \right) \quad (\text{B.39})$$

$$p_1 = z_0 10^{\varepsilon/10m} \quad (\text{B.40})$$

La généralisation des expressions (B.39) et (B.40) donne les relations suivantes :

$$\begin{cases} p_{i+1} = z_i 10^{\varepsilon/10m} \\ z_i = p_i 10^{\varepsilon/10(1-m)} \end{cases} \quad (\text{B.41})$$

P_0 étant obtenu au moyen de l'équation (B.40)

Afin d'obtenir un algorithme récursif, les expressions (B.40) peuvent encore être transformées, pour éviter quelques calculs redondances, pour ce faire considérant les deux constantes a et b suivantes :

$$\begin{cases} b = 10^{\frac{\varepsilon}{10m}} \\ a = 10^{\frac{\varepsilon}{10(1-m)}} \end{cases} \quad (\text{B.42})$$

$$ab = 10^{\varepsilon/10m(1-m)} \quad (\text{B.43})$$

Il est aisé d'écrire d'après l'équation (B.40) que :

$$\begin{cases} \frac{z_0}{p_0} = \frac{z_1}{p_1} = \dots = \frac{z_{N-1}}{p_{N-1}} = a \\ \frac{p_1}{z_0} = \frac{p_2}{z_1} = \dots = \frac{p_N}{p_{N-1}} = b \end{cases} \quad (\text{B.44})$$

Où encore

$$\begin{cases} \frac{z_1}{z_0} = \frac{z_2}{z_1} = \dots = \frac{z_{N-1}}{z_{N-2}} = ab \\ \frac{p_2}{p_1} = \frac{p_3}{p_2} = \dots = \frac{p_N}{p_{N-1}} = ab \end{cases} \quad (\text{B.45})$$

Annexe C

ANNEXE C

Table des dérivées d'ordre fractionnaire de quelques fonctions usuelles

Fonction $f(t)$	la dérivée ${}_{-\infty}D_t^\alpha f(t)$
1	$\frac{t^{-\alpha}}{\Gamma(1-\alpha)}$
$(t)^\lambda$	$\frac{\Gamma(\alpha+1)}{\Gamma(\lambda-\alpha+1)}(t)^{\lambda-\alpha}, \lambda > -1$
$(a-t)^{-1/2}$, pour $\alpha = \frac{1}{2}$	$\sqrt{\frac{a}{\pi t}} \frac{1}{a-t} \quad a > t$
$(a-t)^{-1/2}$, pour $\alpha = -\frac{1}{2}$	$\frac{1}{\sqrt{\pi}} \ln \frac{\sqrt{a} + \sqrt{t}}{\sqrt{a} - \sqrt{t}}$
e^{at}	$E_t(-v, a)$
$t e^{at}$	$tE_t(-\alpha, a) + \alpha E_t(1-\alpha, a)$
$E_t(\mu, a), \quad \mu > -1$	$E_t(\mu - \alpha, a)$
$tE_t(\mu, a), \quad \mu > -2$	$tE_t(\mu - \alpha, a) + \alpha E_t(\mu - \alpha + 1, a)$
$\sin(\lambda t)$	$\lambda^\alpha \sin(\lambda t + \frac{\pi\alpha}{2})$
$\cos(\lambda t)$	$\lambda^\alpha \cos(\lambda t + \frac{\pi\alpha}{2})$

Annexe D

Table des transformées de Laplace et Laplace inverse

La transformée de Laplace	Fonction
$\frac{s^{\alpha-1}}{s^{\alpha} \mp \lambda}, \Re(s) > \lambda ^{1/\alpha}$	$E_{\alpha,1}(\pm \lambda t^{\alpha})$
$\frac{k! s^{\alpha-\beta}}{(s^{\alpha} \mp \lambda)^{k+1}}, \Re(s) > \lambda ^{1/\alpha}$	$t^{\alpha k + \beta - 1} E_{\alpha,\beta}^{(k)}(\pm \lambda, t^{\alpha})$
$\frac{k!}{(\sqrt{s} \mp \lambda)^{k+1}}, \Re(s) > \lambda^2$	$\frac{k-1}{t} \frac{E_{1,1}^{(k)}(\pm \lambda \sqrt{t})}{\sqrt{2}}$
$\frac{1}{s^{\alpha}}$	$\frac{t^{\alpha-1}}{\Gamma(\alpha)}$
$\arctan \frac{k}{s}$	$\frac{1}{t} \sin(kt)$
$\log \frac{s^2 - a^2}{s^2}$	$\frac{2}{t} (1 - \cosh at)$
$\log \frac{s^2 + a^2}{s^2}$	$\frac{2}{t} (1 - \cos at)$
$\log \frac{s-a}{s-b}$	$\frac{1}{t} (e^{bt} - e^{at})$
$\frac{e^{-k\sqrt{s}}}{\sqrt{s}(a + \sqrt{s})}, (k \geq 0)$	$e^{ak} e^{a^2 t} \operatorname{erfc}\left(a\sqrt{t} + \frac{k}{2\sqrt{t}}\right)$
$\frac{ae^{-k\sqrt{s}}}{s(a + \sqrt{s})}, (k \geq 0)$	$\operatorname{erfc}\left(\frac{k}{2\sqrt{t}}\right) - e^{ak} e^{a^2 t} \operatorname{erfc}\left(a\sqrt{t} + \frac{k}{2\sqrt{t}}\right)$
$\frac{1}{s\sqrt{s}} e^{-k\sqrt{s}}, (k \geq 0)$	$2\sqrt{\frac{t}{\pi}} e^{-\frac{k^2}{4t}} - k \operatorname{erfc}\left(\frac{k}{2\sqrt{t}}\right)$
$\frac{1}{\sqrt{s}} e^{-k\sqrt{s}}, (k \geq 0)$	$\frac{1}{\sqrt{\pi t}} e^{-\frac{k^2}{4t}}$
$\frac{1}{s} e^{-k\sqrt{s}}, (k \geq 0)$	$\operatorname{erfc}\left(\frac{k}{2\sqrt{t}}\right)$
$e^{-k\sqrt{s}}, (k \geq 0)$	$\frac{k}{2\sqrt{\pi t^3}} e^{-\frac{k^2}{4t}}$
$\frac{1}{s^v} e^{k/s}, (v > 0)$	$\left(\frac{t}{k}\right)^{(v-1)/2} I_{v-1}(2\sqrt{kt})$

$$\frac{1}{s^v} e^{-k/s}, (v > 0)$$

$$\frac{1}{s\sqrt{s}} e^{k/s}$$

$$\frac{1}{s\sqrt{s}} e^{-k/s}$$

$$\frac{1}{\sqrt{s}} e^{k/s}$$

$$\frac{1}{\sqrt{s}} e^{-k/s}$$

$$\frac{1}{s} e^{-k/s}$$

$$\frac{k}{s^2 + k^2} \coth \frac{\pi s}{2k}$$

$$\frac{1}{\sqrt{s}}$$

$$\frac{1}{s\sqrt{s}}$$

$$\frac{1}{s^n \sqrt{s}}, (n = 1, 2, \dots)$$

$$\frac{s}{(s-a)^{3/2}}$$

$$\sqrt{s-a} - \sqrt{s-b}$$

$$\frac{1}{\sqrt{s+a}}$$

$$\frac{\sqrt{s}}{s-a^2}$$

$$\left(\frac{t}{k}\right)^{(v-1)/2} J_{v-1}(2\sqrt{kt})$$

$$\frac{1}{\sqrt{\pi k}} \sinh 2\sqrt{kt}$$

$$\frac{1}{\sqrt{\pi k}} \sin 2\sqrt{kt}$$

$$\frac{1}{\sqrt{\pi t}} \cosh 2\sqrt{kt}$$

$$\frac{1}{\sqrt{\pi t}} \cos 2\sqrt{kt}$$

$$J_0(2\sqrt{kt})$$

$$|\sin kt|$$

$$\frac{1}{\sqrt{\pi t}}$$

$$2\sqrt{\frac{t}{\pi}}$$

$$\frac{2^n t^{n-(1/2)}}{1.3.5 \dots (2n-1)\sqrt{\pi}}$$

$$\frac{1}{\sqrt{\pi t}} e^{at} (1 + 2at)$$

$$\frac{1}{2\sqrt{\pi t^3}} (e^{bt} - e^{at})$$

$$\frac{1}{\sqrt{\pi t}} - ae^{a^2 t} \operatorname{erfc}(a\sqrt{t})$$

$$\frac{1}{\sqrt{\pi t}} + ae^{a^2 t} \operatorname{erf}(a\sqrt{t})$$

Mittag-Leffler	$E_q(at^q) = \sum_{n=0}^{\infty} \frac{a^n t^{nq}}{\Gamma(nq + 1)}$	$\frac{s^q}{s(s^q - a)}$
Agarwal	$E_{\alpha,\beta}(t^\alpha) = \sum_{m=0}^{\infty} \frac{t^{\left(m + \frac{\beta-1}{\alpha}\right)\alpha}}{\Gamma(\alpha m + \beta)}$	$\frac{s^{\alpha-\beta}}{s^\alpha - 1}$
Erdelyi	$E_{\alpha,\beta}(t) = \sum_{m=0}^{\infty} \frac{t^m}{\Gamma(\alpha m + \beta)}$	$\sum_{m=0}^{\infty} \frac{\Gamma(m+1)}{\Gamma(\alpha m + \beta) s^{m+1}}$
Robotnov-Hartley	$F_q(a, t) = \sum_{n=0}^{\infty} \frac{a^n t^{(n+1)q-1}}{\Gamma(\{n+1\}q)}$	$\frac{1}{s^q - a}$
Miller-Ross	$E_t(v, a) = \sum_{k=0}^{\infty} \frac{a^k t^{k+v}}{\Gamma(v+k+1)}$	$\frac{s^{-v}}{s - a}$
Generalized R	$R_{q,v}(a, t) = \sum_{n=0}^{\infty} \frac{a^n t^{(n+1)q-1-v}}{\Gamma(\{n+1\}q - v)}$	$\frac{s^v}{s^q - a}$
Generalized G	$G_{q,v,r}(at) = \sum_{i=0}^{\infty} \frac{\{(-r)(-1-r)\dots(1-j-r)\}(-a)^j t^{(r+j)q-v-1}}{\Gamma(1+j)\Gamma(\{r+j\}q-v)}$	$\frac{s^v}{(s^q - a)^r}$

Résumé

Dans le cadre de ce mémoire, nous avons pour objectif la conception et la réalisation d'un logiciel sous Matlab dédié à l'analyse et la simulation des systèmes d'ordre fractionnaire.

Le travail que nous avons présenté se résume en trois chapitres, dans le premier chapitre nous présentons quelques notions fondamentales des systèmes d'ordre fractionnaires, les différents modèles de représentation des systèmes dynamiques fractionnaires et leurs propriétés.

Le deuxième chapitre est consacré au développement des algorithmes numériques nécessaires à la simulation des systèmes d'ordre non entier, ainsi que nous présentons les passages entre les différentes représentations des systèmes fractionnaires et quelques méthodes d'approximation des systèmes fractionnaires.

Le dernier chapitre présente notre logiciel qui est dédié à l'analyse et la simulation des systèmes dynamiques d'ordre non entier.

Le logiciel que nous avons réalisé permet donc de :

- ✓ Simuler un système d'ordre fractionnaire.
- ✓ Tester la stabilité d'une fonction de transfert.
- ✓ Tester la commandabilité et l'observabilité.
- ✓ Calculer les différentes approximations.