



UNIVERSITÉ BORDEAUX 1 - ENSEIRB

UMR 5131 CNRS

AUTOMATIQUE, PRODUCTIQUE, SIGNAL & IMAGE

The CRONE Toolbox : module identification par modèle non entier

O. Cois, M. Aoun, P. Melchior, P. Lanusse, F. Dancla et A. Oustaloup



Olivier Cois

Action thématique SDNE

29 mars 2002

Équation différentielle non entière

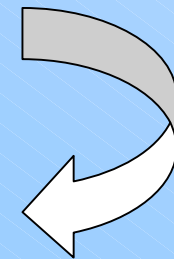
$$\alpha_1 D^{(n_{\alpha_1})} y(t) + \dots + \alpha_I D^{(n_{\alpha_I})} y(t) = \beta_1 D^{(n_{\beta_1})} u(t) + \dots + \beta_J D^{(n_{\beta_J})} u(t)$$

- Les ordres de dérivation sont tous **réels** entiers ou non entiers
- Objectif : **estimer** le meilleur jeu de paramètres qui permet une **reproduction fidèle** du comportement dynamique observé

$$\hat{\theta} = [\hat{\alpha}_1, \hat{n}_{\alpha_1}, \dots, \hat{\alpha}_I, \hat{n}_{\alpha_I}, \hat{\beta}_1, \hat{n}_{\beta_1}, \dots, \hat{\beta}_J, \hat{n}_{\beta_J}]$$



Problème d'optimisation
vis-à-vis d'un critère



Modèles non entiers :

- **linéarité** vis-à-vis des **coefficients** de l'équation différentielle
- **non linéarité** vis-à-vis des **ordres** de dérivation
- ↳ deux types de modèles non entiers

$$\alpha_1 D^{(n_{\alpha_1})} y(t) + \dots + \alpha_I D^{(n_{\alpha_I})} y(t) = \beta_1 D^{(n_{\beta_1})} u(t) + \dots + \beta_J D^{(n_{\beta_J})} u(t)$$

Seuls les coefficients sont estimés

- Les ordres de dérivation sont fixés par l'utilisateur
- Modèle à erreur d'équation
- Optimisation linéaire
- ↳ technique des moindres carrés (LS)

Les ordres sont estimés au même titre que les coefficients

- Modèle à erreur de sortie
- Optimisation non linéaire
- ↳ technique non linéaire (Marquardt)

Modèle à erreur d'équation

Introduction

$$\alpha_1 D^{(n_{\alpha_1})} y(t) + \dots + \alpha_I D^{(n_{\alpha_I})} y(t) = \beta_1 D^{(n_{\beta_1})} u(t) + \dots + \beta_J D^{(n_{\beta_J})} u(t) + e(t)$$

- Seuls les coefficients des opérateurs sont estimés
- Les ordres sont fixés par l'utilisateur
- Deux techniques d'estimation paramétrique
 - par discrétisation de l'équation différentielle non entière
 - par construction d'une matrice de régression continue

1. Discrétisation de l'équation différentielle

Approximation de Grünwald de l'opérateur de dérivation non entière :

$$D^n f(t) = \frac{1}{h^n} \sum_{k=0}^K (-1)^k \binom{n}{k} f((K-k)h)$$



$$\alpha_1 D^{(n_{\alpha_1})} y(t) + \dots + \alpha_I D^{(n_{\alpha_I})} y(t) = \beta_1 D^{(n_{\beta_1})} u(t) + \dots + \beta_J D^{(n_{\beta_J})} u(t) + e(t)$$

$$\sum_{i=1}^I \frac{\alpha_i}{h^{n_{\alpha_i}}} \sum_{k=0}^K (-1)^k \binom{n_{\alpha_i}}{k} y((K-k)h) = \sum_{j=1}^J \frac{\beta_j}{h^{n_{\beta_j}}} \sum_{k=0}^K (-1)^k \binom{n_{\beta_j}}{k} u((K-k)h) + e(Kh)$$



Équation récurrente de dimension croissante

1. Discrétisation de l'équation différentielle

➤ Prédiction de la sortie du modèle à un pas

$$y(Kh) = \frac{\sum_{i=1}^I \frac{\alpha_i}{h^{n\alpha_i}} \sum_{k=1}^K (-1)^k \binom{n\alpha_i}{k} y((K-k)h) + \sum_{j=1}^J \frac{\beta_j}{h^{n\beta_j}} \sum_{k=0}^K (-1)^k \binom{n\beta_j}{k} u((K-k)h)}{\sum_{i=1}^I \frac{\alpha_i}{h^{n\alpha_i}}}$$

➤ Non linéaire vis-à-vis des paramètres à estimer

↳ Changement de variables :

$$y(Kh) = -\sum_{i=1}^I a_i Y_i(Kh) + \sum_{j=1}^J b_j U_j(Kh) + \tilde{e}(Kh)$$

$$a_i = \frac{\frac{\alpha_i}{h^{n\alpha_i}}}{\sum_{j=1}^I \frac{\alpha_j}{h^{n\alpha_j}}}, 1 \leq i \leq I \quad \text{et} \quad b_j = \frac{\frac{\beta_j}{h^{n\beta_j}}}{\sum_{i=1}^I \frac{\alpha_i}{h^{n\alpha_i}}}, 1 \leq j \leq J \quad \text{et} \quad \sum_{i=1}^I a_i = 1$$

$$Y_i(t) = \sum_{k=1}^K (-1)^k \binom{n\alpha_i}{k} y((K-k)h) \quad \text{et} \quad U_j(t) = \sum_{k=0}^K (-1)^k \binom{n\beta_j}{k} u((K-k)h)$$

1. Discrétisation de l'équation différentielle

➤ Optimisation sous contrainte

➤ $a_1 = 1 - \sum_{i=2}^I a_i$ ⇒ Estimation de $\theta_r = (a_2, \dots, a_I, b_1, \dots, b_J)^T$

➤ Régression linéaire

$$\hat{\mathbf{Y}}(\theta_r) + \mathbf{Y}_1 = \mathbf{H} \theta_r$$

$$\hat{\mathbf{Y}} = [\hat{y}(kh), \dots, \hat{y}((k+N)h)]^T$$

$$\mathbf{Y}_1 = [Y_1(kh), \dots, Y_1((k+N)h)]^T$$

$$\mathbf{H} = \begin{bmatrix} -Y_2(kh) + Y_1(kh) & \dots & -Y_I(kh) + Y_1(kh) & U_1(kh) & \dots & U_J(kh) \\ -Y_2((k+1)h) + Y_1((k+1)h) & \dots & -Y_I((k+1)h) + Y_1((k+1)h) & U_1((k+1)h) & \dots & U_J((k+1)h) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -Y_2((k+N)h) + Y_1((k+N)h) & \dots & -Y_I((k+N)h) + Y_1((k+N)h) & U_1((k+N)h) & \dots & U_J((k+N)h) \end{bmatrix}$$

1. Discrétisation de l'équation différentielle

➤ Estimation des paramètres du modèle récurrent

➤ Minimisation de la norme de l'erreur de prédiction $\hat{\theta}_r = \arg \min(\mathbf{E}^T \mathbf{E})$

$$\rightarrow \hat{\theta}_r = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (\mathbf{Y} + \mathbf{Y}_1)$$

➤ Retour aux paramètres du modèle continu

$$\begin{bmatrix} (a_1 - 1)h^{-n\alpha_1} & a_1 h^{-n\alpha_2} & \dots & a_1 h^{-n\alpha_{I-1}} \\ a_2 h^{-n\alpha_1} & (a_2 - 1)h^{-n\alpha_2} & \dots & a_2 h^{-n\alpha_{I-1}} \\ \vdots & \vdots & & \vdots \\ a_{I-1} h^{-n\alpha_1} & a_{I-1} h^{-n\alpha_2} & \dots & (a_{I-1} - 1)h^{-n\alpha_{I-1}} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{I-1} \end{bmatrix} = \begin{bmatrix} -a_1 h^{-n\alpha_I} \\ -a_2 h^{-n\alpha_I} \\ \vdots \\ -a_{I-1} h^{-n\alpha_I} \end{bmatrix}$$

$$\beta_j = b_j \sum_{i=1}^I \alpha_i h^{n\beta_j - n\alpha_i}$$

2. Construction d'une matrice de régression continue

➤ Méthode directe

➤ Régression linéaire directe

$$D^{(n_{\alpha_0})}y(t) = -\alpha_1 D^{(n_{\alpha_1})}y(t) - \dots - \alpha_I D^{(n_{\alpha_I})}y(t) + \beta_1 D^{(n_{\beta_1})}u(t) + \dots + \beta_J D^{(n_{\beta_J})}u(t) + e(t)$$

➤ Forme matricielle

$$D^{(n_{\alpha_0})}\mathbf{Y} = \mathbf{H}\boldsymbol{\theta} + \mathbf{E}$$

$$\boldsymbol{\theta} = [\alpha_1 \quad \dots \quad \alpha_I \quad \beta_1 \quad \dots \quad \beta_J]^T$$

$$\mathbf{Y} = [y(0) \quad y(h) \quad \dots \quad y(Nh)]^T \quad \mathbf{E} = [e(0) \quad e(h) \quad \dots \quad e(Nh)]^T$$

$$H = \begin{bmatrix} -D^{(n_{\alpha_1})}y(0) & \dots & -D^{(n_{\alpha_I})}y(0) & D^{(n_{\beta_1})}u(0) & \dots & D^{(n_{\beta_J})}u(0) \\ -D^{(n_{\alpha_1})}y(h) & \dots & -D^{(n_{\alpha_I})}y(h) & D^{(n_{\beta_1})}u(h) & \dots & D^{(n_{\beta_J})}u(h) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -D^{(n_{\alpha_1})}y(Nh) & \dots & -D^{(n_{\alpha_I})}y(Nh) & D^{(n_{\beta_1})}u(Nh) & \dots & D^{(n_{\beta_J})}u(Nh) \end{bmatrix}$$

2. Construction d'une matrice de régression continue

➤ Méthode directe

➤ Estimation paramétrique

⇒ Méthode des moindres carrés linéaires

$$\boldsymbol{\theta}_{\text{opt}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (D^{n_{a0}} \mathbf{Y})$$

⇒ Les dérivées continues sont remplacées par l'approximation

$$D^n f(t) = \frac{1}{h^n} \sum_{k=0}^K (-1)^k \binom{n}{k} f((K-k)h)$$

↳ problèmes d'amplification du bruit

2. Construction d'une matrice de régression continue

➤ Méthode par intégration

- Intégration préalable de l'équation différentielle non entière à l'ordre le plus élevé du modèle (CI nulles)

$$\alpha_0 y(t) + \dots + \alpha_I D^{(n_{\alpha_I} - n_{\alpha_0})} y(t) = \beta_1 D^{(n_{\beta_1} - n_{\alpha_0})} u(t) + \dots + \beta_J D^{(n_{\beta_J} - n_{\alpha_0})} u(t) + I^{n_{\alpha_0}} e(t)$$

- Régression linéaire (sous forme matricielle) : **Y = Hθ + E**

$$\theta = [\alpha_1 \quad \dots \quad \alpha_I \quad \beta_1 \quad \dots \quad \beta_J]^T$$

$$Y = [y(0) \quad y(h) \quad \dots \quad y(Nh)]^T \quad E = \begin{bmatrix} I^{n_{\alpha_0}} e(0) & I^{n_{\alpha_0}} e(h) & \dots & I^{n_{\alpha_0}} e(Nh) \end{bmatrix}^T$$

$$H = \begin{bmatrix} -I^{(n_{\alpha_0} - n_{\alpha_1})} y(0) & \dots & -I^{(n_{\alpha_0} - n_{\alpha_I})} y(0) & I^{(n_{\alpha_0} - n_{\beta_1})} u(0) & \dots & I^{(n_{\alpha_0} - n_{\beta_J})} u(0) \\ -I^{(n_{\alpha_0} - n_{\alpha_1})} y(h) & \dots & -I^{(n_{\alpha_0} - n_{\alpha_I})} y(h) & I^{(n_{\alpha_0} - n_{\beta_1})} u(h) & \dots & I^{(n_{\alpha_0} - n_{\beta_J})} u(h) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -I^{(n_{\alpha_0} - n_{\alpha_1})} y(Nh) & \dots & -I^{(n_{\alpha_0} - n_{\alpha_I})} y(Nh) & I^{(n_{\alpha_0} - n_{\beta_1})} u(Nh) & \dots & I^{(n_{\alpha_0} - n_{\beta_J})} u(Nh) \end{bmatrix}$$

2. Construction d'une matrice de régression continue

➤ Méthode par intégration

➤ Estimation paramétrique

⇒ Méthode des moindres carrés linéaires

$$\boldsymbol{\theta}_{\text{opt}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Y}$$

⇒ Les intégrales continues sont remplacées par l'approximation

$$I^n f(t) = \frac{1}{h^{-n}} \sum_{k=0}^K (-1)^k \binom{-n}{k} f((K-k)h)$$

2. Construction d'une matrice de régression continue


➤ Méthode par filtres de variables d'état non entiers

Objectif : estimer les coefficients $\boldsymbol{\theta} = [a_1 \ \dots \ a_M \ b_0 \ \dots \ b_N]^T$

$$T(t) + a_1 D^{0.5} T(t) + a_2 D^1 T(t) + \dots + a_M D^{0.5M} T(t) = b_0 \phi(t) + b_1 D^{0.5} \phi(t) + \dots + b_N D^{0.5N} \phi(t)$$

$$\begin{aligned} \Leftrightarrow T(t) &= -a_1 D^{0.5} T(t) - \dots - a_M D^{0.5M} T(t) + b_0 \phi(t) + \dots + b_N D^{0.5N} \phi(t) \\ &= \left[-D^{0.5} T(t) \ \dots \ -D^{0.5M} T(t) \ \phi(t) \ \dots \ D^{0.5N} \phi(t) \right] \boldsymbol{\theta} \\ &= \boldsymbol{\varphi}(t) \boldsymbol{\theta} \end{aligned}$$

$$\begin{aligned} T_f(t) &= -a_1 D^{0.5} T_f(t) - \dots - a_M D^{0.5M} T_f(t) + b_0 \phi_f(t) + \dots + b_N D^{0.5N} \phi_f(t) \\ &= \left[-D^{0.5} T_f(t) \ \dots \ -D^{0.5M} T_f(t) \ \phi_f(t) \ \dots \ D^{0.5N} \phi_f(t) \right] \boldsymbol{\theta} \\ &= \boldsymbol{\varphi}_f(t) \boldsymbol{\theta} \end{aligned}$$



Filtrage passe-bas

2. Construction d'une matrice de régression continue

➤ Estimation de la matrice de régression Φ_f^*

→ Évaluation numérique des dérivées successives des signaux température et flux filtrés

→ Utilisation d'un filtre de variables d'état non entier

$$\mathbf{x} = \left[D^{(N_f-1) \times n} z_f(t), D^{(N_f-2) \times n} z_f(t), \dots, D^n z_f(t), z_f(t) \right]^T$$

→ Filtre de Poisson non entier

$$F(s) = \frac{1}{\left(\left(\frac{s}{\omega_f} \right)^n + 1 \right)^{N_f}} = \frac{\omega_f^{n \times N_f}}{s^{n \times N_f} + C_1^{N_f} \omega_f^n s^{n \times (N_f-1)} + \dots + C_{N_f-1}^{N_f} \omega_f^{n \times (N_f-1)} s^n + \omega_f^{n \times N_f}}$$

$$\Downarrow \quad D^n \mathbf{x}(t) = \begin{bmatrix} -C_1^{N_f} \omega_f^n & -C_2^{N_f} \omega_f^{2 \times n} & \dots & C_{N_f-1}^{N_f} \omega_f^{n \times (N_f-1)} & \omega_f^{n \times N_f} \\ 1 & 0 & \dots & \dots & 0 \\ 0 & \ddots & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \omega_f^{n \times N_f} & 0 & \dots & \dots & 0 \end{bmatrix}^T z(t)$$

Introduction

- Modèle issu de la forme modale développée d'un système non entier (forme diagonale de la représentation d'état non entière)

$$\begin{cases} D^n \mathbf{x}_\Delta(t) = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix} \mathbf{x}_\Delta(t) + \mathbf{B}_\Delta u(t) \\ y(t) = \mathbf{C}_\Delta \mathbf{x}_\Delta(t) + e(t) \end{cases} \iff y(t) = \sum_{k=1}^N \mathcal{L}^{-1} \left(\frac{A_k}{p^n - \lambda_k} \right) \otimes u(t) + e(t)$$

- L'objectif de l'identification est d'estimer les paramètres A_k , λ_k et n .

$$\iff \begin{cases} z(t) + \alpha_1 D^{(1*n)} z(t) + \dots + \alpha_N D^{(N*n)} z(t) = \beta_0 u(t) + \dots + \beta_{N-1} D^{((N-1)*n)} u(t) \\ y(t) = z(t) + e(t) \end{cases}$$

Estimation paramétrique

► Algorithme à erreur de sortie

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \left(\sum_{k=0}^K (y_{\text{mesuré}}(kh) - y_{\text{modèle}}(kh, \boldsymbol{\theta}))^2 \right)$$

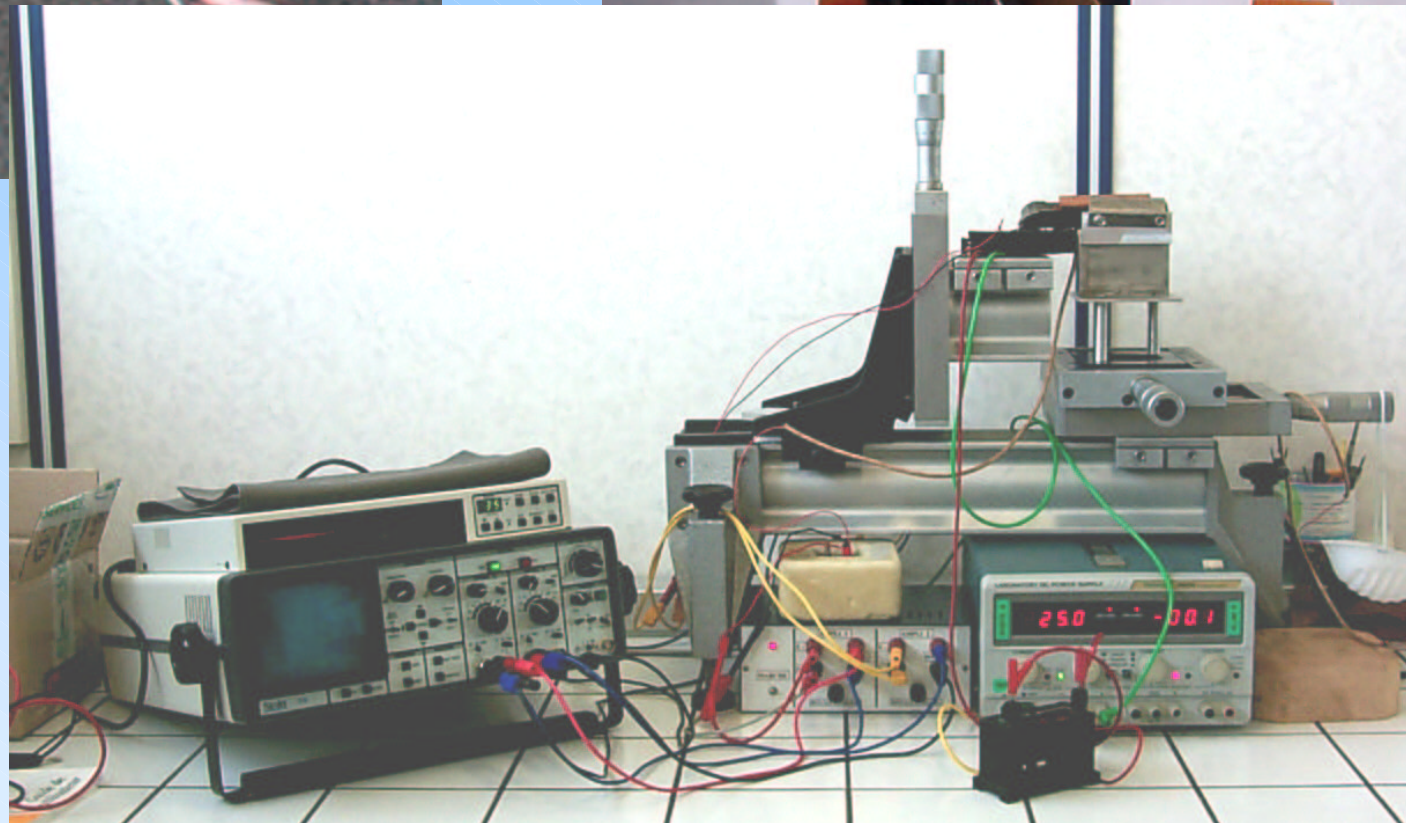
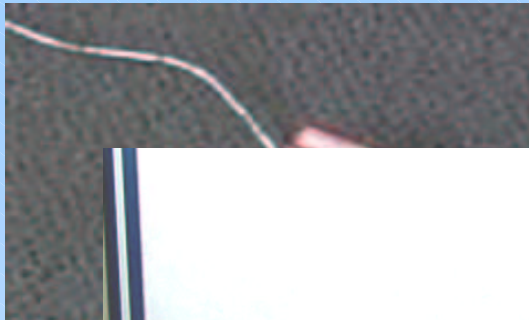
⇒ La sortie du modèle est non linéaire vis-à-vis des paramètres à estimer

↳ Algorithme d'optimisation non linéaire (Marquardt)

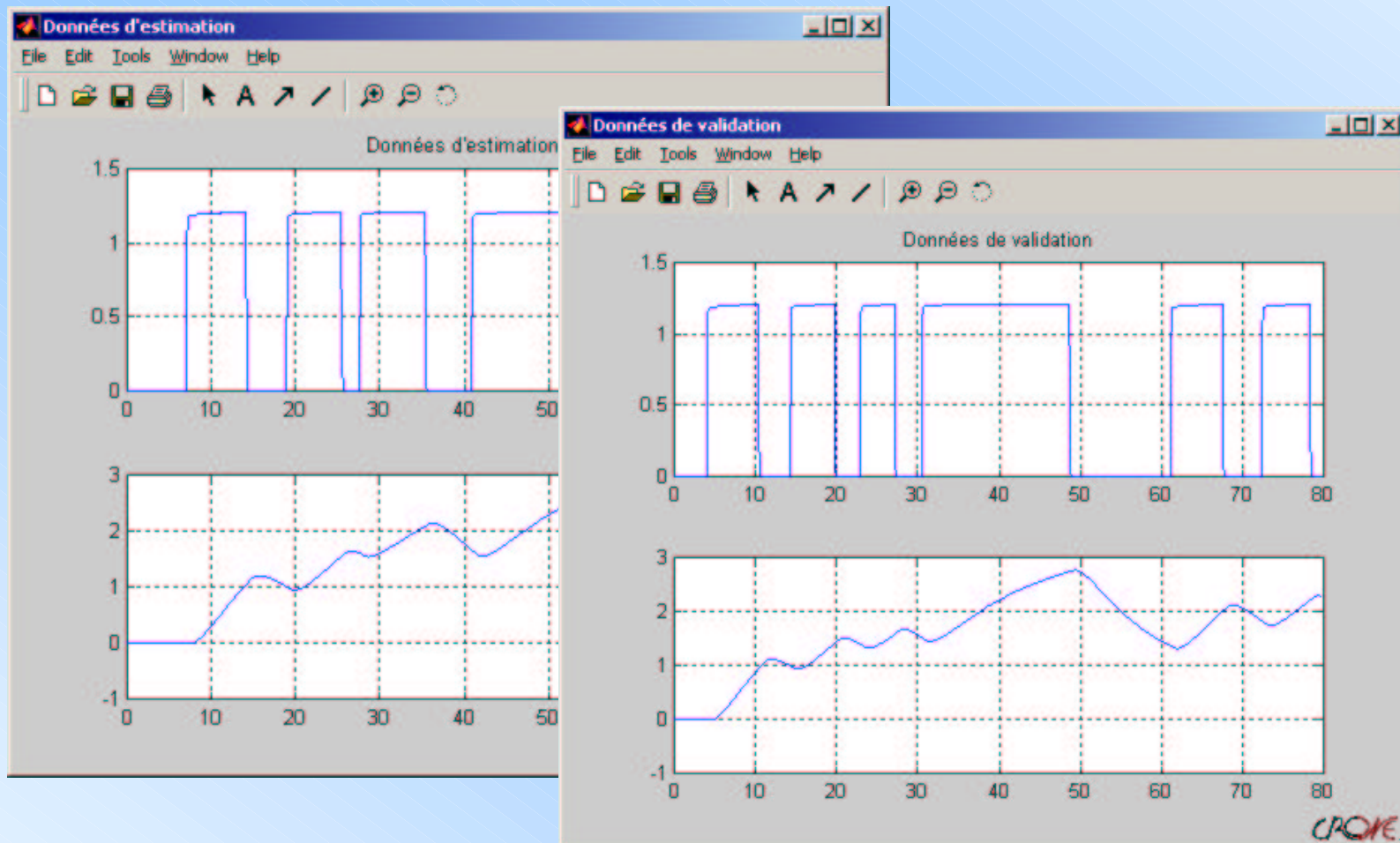
$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \left\{ \left[\mathbf{J}_{\boldsymbol{\theta}\boldsymbol{\theta}}'' + \xi \mathbf{I} \right]^{-1} \mathbf{J}_{\boldsymbol{\theta}}' \right\}_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}$$

$$\left\{ \begin{array}{l} \mathbf{J}_{\boldsymbol{\theta}}' = -2 \sum_{k=0}^{K-1} \varepsilon(kh) \mathbf{S}(kh, \boldsymbol{\theta}) : \text{gradient} \\ \mathbf{J}_{\boldsymbol{\theta}\boldsymbol{\theta}}'' \approx 2 \sum_{k=0}^{K-1} \mathbf{S}(kh, \boldsymbol{\theta}) \mathbf{S}^T(kh, \boldsymbol{\theta}) : \text{hessien} \\ \mathbf{S}(kh, \boldsymbol{\theta}) = \frac{\partial y(kh, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} : \text{fonction de sensibilité de la sortie} \\ \xi : \text{paramètre de Marquardt} \end{array} \right.$$

- Le flux est appliqué à l'aide d'une micro résistance

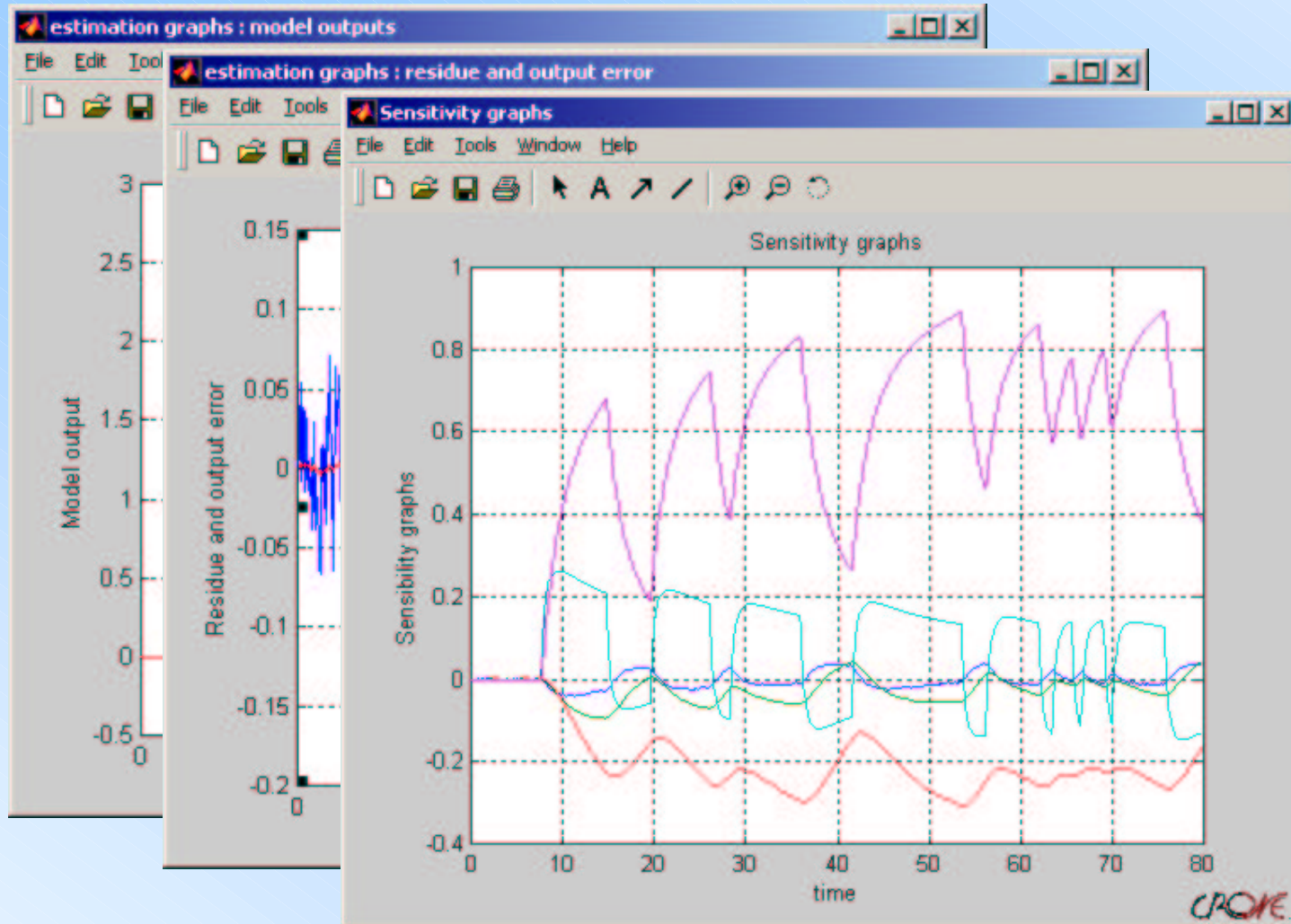


- 2 enregistrements : estimation et validation ($T_s=0.1s$)

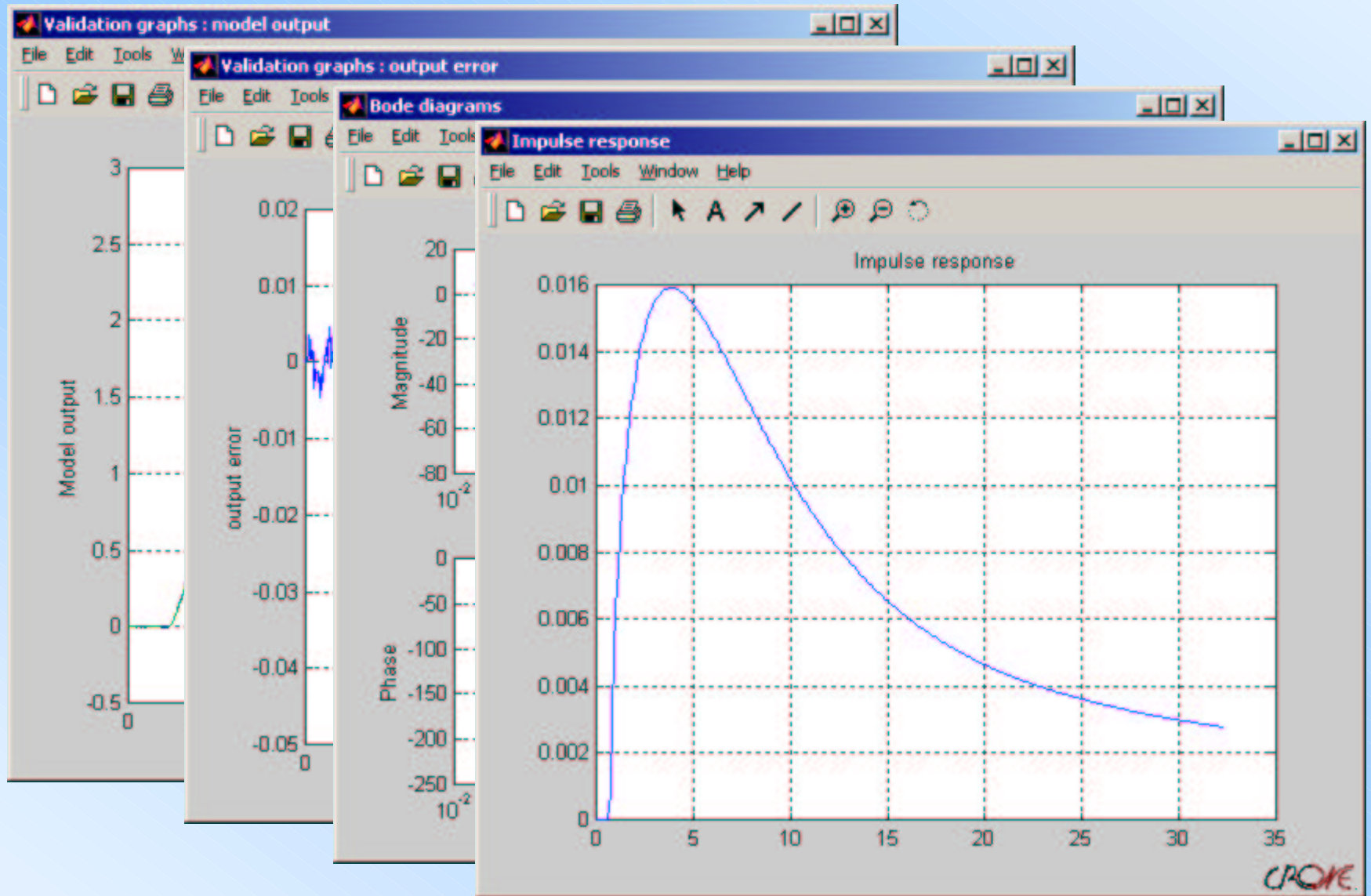


- Estimation paramétrique : $M=3, N=1$ (5 paramètres)

$$T(t) + 10.90 D^{0.5} T(t) - 1.04 D^1 T(t) + 39.81 D^{0.5 M} T(t) = 6.22 \phi(t) - 1.65 D^{0.5} \phi(t)$$



➤ Validation du modèle



Expression des modèles non entiers

→ Fonction de transfert rationnelle

$$H(s) = \frac{\beta_{N-1}s^{n\beta_{N-1}} + \dots + \beta_0}{s^{n\alpha_N} + \alpha_{N-1}s^{n\alpha_{N-1}} + \dots + \alpha_0}$$

→ Forme modale développée

$$H(s) = \sum_{l=1}^N \frac{A_l}{s^n - \lambda_l}$$

→ Forme modale factorisée

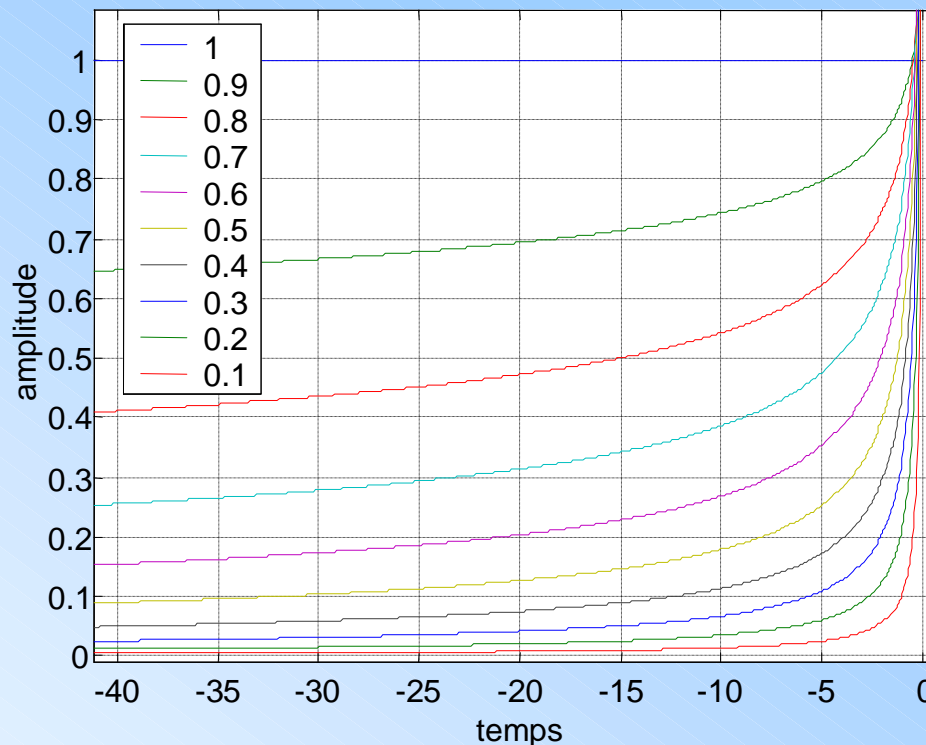
$$H(s) = \frac{A \prod_{l=1}^{N_z} (s^n - z_l)}{\prod_{l=1}^N (s^n - \lambda_l)}$$

Rappels sur la dérivation non entière

Dérivation non entière

➤ Intégration non entière

[Liouville, 1832]
$$I_{t_0}^n f(t) \stackrel{\Delta}{=} \frac{1}{\Gamma(n)} \underbrace{\int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1-n}} d\tau}_{\text{produit de convolution}} = \int_{t_0}^t \underbrace{\frac{1}{\Gamma(n)(t-\tau)^{1-n}}}_{\text{facteur d'oubli}} f(\tau) d\tau$$



Dérivation non entière

➤ Intégration non entière

[Liouville, 1832]
$$I_{t_0}^n f(t) \stackrel{\Delta}{=} \frac{1}{\Gamma(n)} \underbrace{\int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1-n}} d\tau}_{\text{produit de convolution}} = \int_{t_0}^t \underbrace{\frac{1}{\Gamma(n)(t-\tau)^{1-n}}}_{\text{facteur d'oubli}} f(\tau) d\tau$$

➤ Transformée de Laplace d'une intégrale d'ordre non entier

$$\mathcal{L}\{I_0^n f(t)\} = \int_0^\infty e^{-st} \left(\frac{1}{\Gamma(n)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-n}} d\tau \right) dt = \frac{1}{s^n} F(s)$$

Dérivation non entière

➤ Dérivation non entière

$$D_{t_0}^n f(t) \stackrel{\Delta}{=} \left(\frac{d}{dt} \right)^{\lfloor n \rfloor + 1} \left(I_{t_0}^{1 - (n - \lfloor n \rfloor)} f(t) \right)$$

$$D_{t_0}^n f(t) \stackrel{\Delta}{=} \frac{1}{\Gamma(m-n)} \left(\frac{d}{dt} \right)^m \left(\int_{t_0}^t \frac{f(\tau)}{(t-\tau)^{1-(m-n)}} d\tau \right) \quad \text{où} \quad \begin{cases} t > t_0 \\ n > 0 \\ m = \lfloor n \rfloor + 1 \end{cases}$$

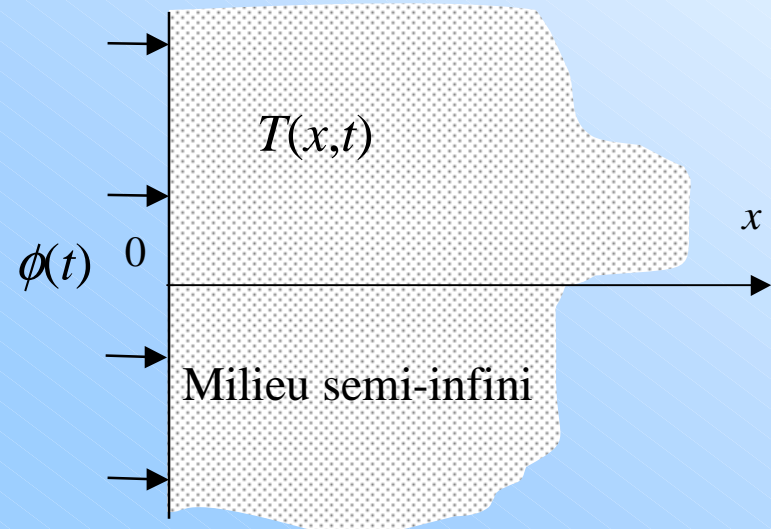
➤ Transformée de Laplace d'une dérivée d'ordre non entier

$$\mathcal{L}\{D_0^n f(t)\} = s^n F(s) - \sum_{k=0}^{m-1} s^{m-1-k} D^k \left(I_0^{m-n} f(t) \right) \Big|_{t=0^+} \quad \text{avec} \quad \begin{cases} n \in \mathbb{R}^{+*} \\ m = \lceil n \rceil \end{cases}$$

Dérivation non entière et systèmes thermiques

➤ Milieu semi-infini mono-dimensionnel

$$\left\{ \begin{array}{l} \frac{\partial T(x,t)}{\partial t} = \alpha \frac{\partial^2 T(x,t)}{\partial x^2}, \quad 0 < x < \infty, t > 0 \\ -\lambda \frac{\partial T(x,t)}{\partial x} = \phi(t), \quad x = 0, t > 0 \\ T(x,t) = 0, \quad 0 \leq x < \infty, t = 0 \end{array} \right.$$



$$H(x,s) = \frac{\bar{T}(x,s)}{\bar{\phi}(s)} = \frac{1}{\sqrt{s} \sqrt{\lambda \rho C_p}} e^{-x \sqrt{s/\alpha}}$$

Dérivation non entière et systèmes thermiques

➤ Milieu semi-infini mono-dimensionnel $H(x,s) = \frac{\bar{T}(x,s)}{\bar{\varphi}(s)} = \frac{1}{\sqrt{s} \sqrt{\lambda \rho C_p}} e^{-x\sqrt{s/\alpha}}$

→ $x = 0$ $H(0,s) = \frac{\bar{T}(0,s)}{\bar{\varphi}(s)} = \frac{1}{\sqrt{s} \sqrt{\lambda \rho C_p}}$ $T(0,t) = \frac{1}{\sqrt{\lambda \rho C_p}} I^{0.5} \varphi(t)$

→ $x > 0$: développements en série

- *développement naturel*

$$\tilde{H}_{1,K}(x,s) = \frac{1}{\sqrt{\lambda \rho C_p}} \sum_{k=0}^K a_k s^{(k-1)/2}$$

$$a_k = (-1)^k \frac{x^k}{\alpha^{k/2} k!}$$

- *développement issu de Padé*

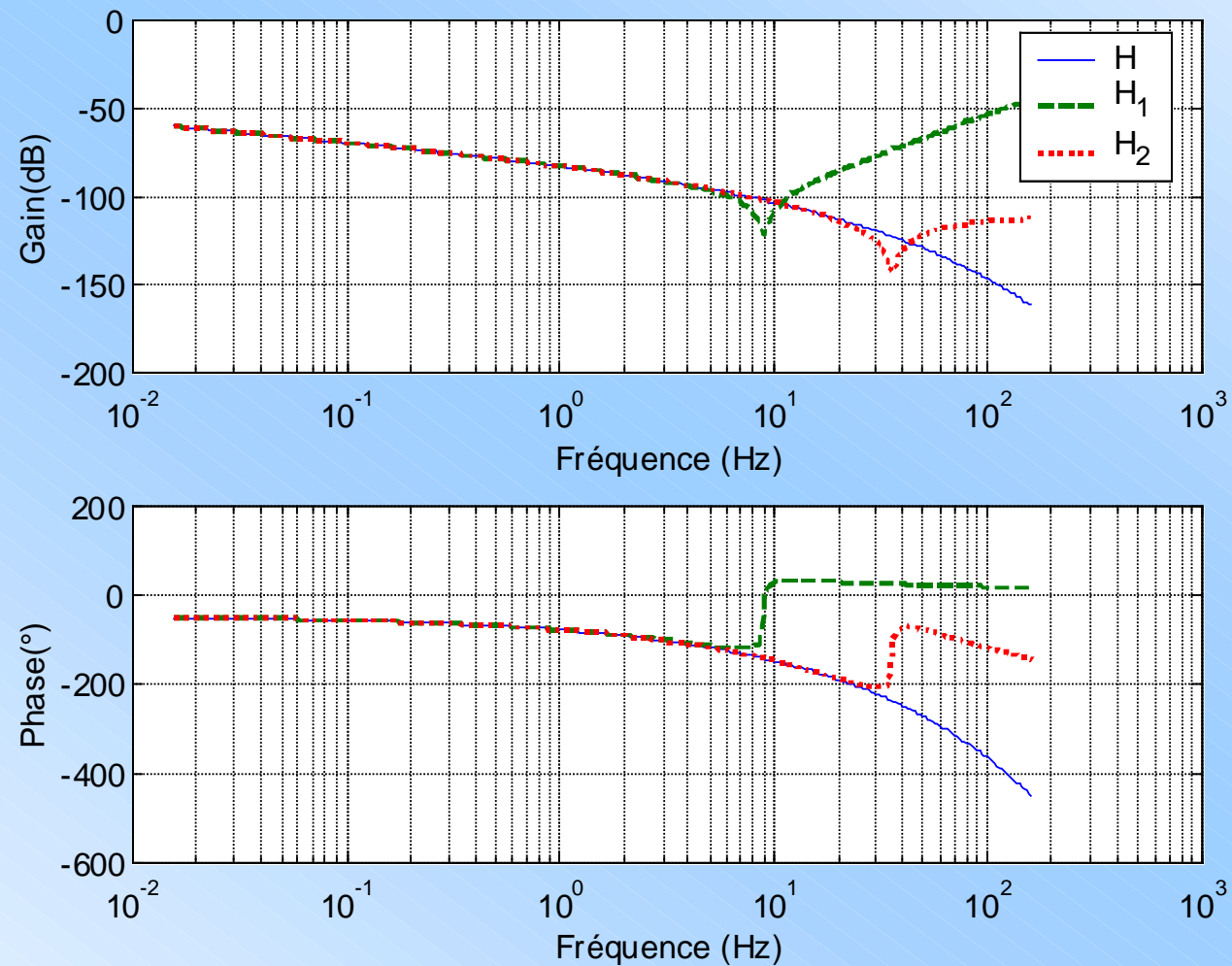
$$H(x,s) = \frac{1}{\sqrt{s} \sqrt{\lambda \rho C_p}} \frac{e^{-\frac{x}{2}\sqrt{s/\alpha}}}{e^{\frac{x}{2}\sqrt{s/\alpha}}}$$

$$\tilde{H}_{2,K}(x,s) = \frac{1}{\sqrt{\lambda \rho C_p}} \frac{\sum_{k=0}^K a'_k s^{k/2}}{\sum_{k=0}^K |a_k| s^{(k+1)/2}}$$

$$a'_k = (-1)^k \frac{(x/2)^k}{\alpha^{k/2} k!}$$

Dérivation non entière et systèmes thermiques

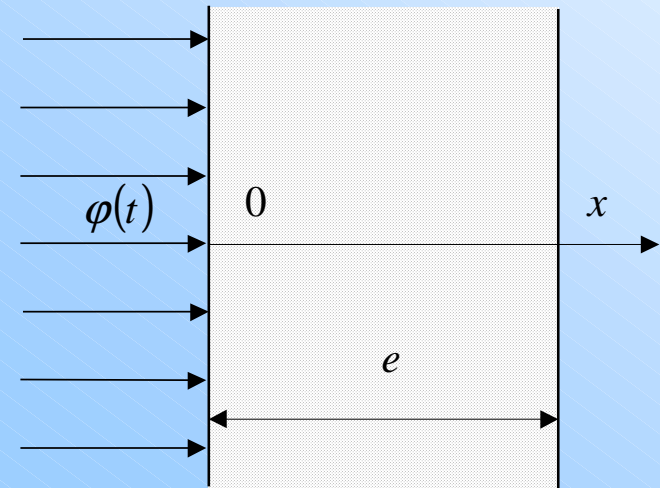
➤ Milieu semi-infini mono-dimensionnel $K = 5$ $x = 1\text{mm}$



Dérivation non entière et systèmes thermiques

➤ Milieu fini mono-dimensionnel

$$H(x, s) = \frac{\bar{T}(x, s)}{\bar{\varphi}(s)} = \frac{\cosh\left((e-x)\sqrt{\frac{s}{\alpha}}\right)}{\lambda\sqrt{\frac{s}{\alpha}} \sinh\left(e\sqrt{\frac{s}{\alpha}}\right)}$$



→ $x = 0$: impédance thermique

- *développement naturel : transfert entier*

$$\tilde{H}_{1,K}(0, s) = \frac{\sum_{k=0}^K a_k s^k}{\lambda \sum_{k=0}^K \frac{e}{\alpha(2k+1)} a_k s^{k+1}}$$

$$a_k = \frac{e^{2k}}{(2k)! \alpha^k}$$

→ *comportement asymptotique différent du modèle analytique*

Dérivation non entière et systèmes thermiques

➤ Milieu fini mono-dimensionnel

- *développement spécifique : transfert non entier*

$$H(0, s) = \frac{\bar{T}(0, s)}{\bar{\varphi}(s)} = \frac{\exp\left(2e\sqrt{\frac{s}{\alpha}}\right) + 1}{\lambda\sqrt{\frac{s}{\alpha}}\left(\exp\left(2e\sqrt{\frac{s}{\alpha}}\right) - 1\right)}$$

$$\tilde{H}_{2,K}(0, s) = \frac{1 + \sum_{k=1}^K a'_k s^{k/2}}{\lambda \sum_{k=1}^K \frac{a'_k}{\sqrt{\alpha}} s^{(k+1)/2}} \quad a'_k = \frac{2^{k-1} e^k}{k! \alpha^{k/2}}$$

→ *même comportement asymptotique que le modèle analytique*

Dérivation non entière et systèmes thermiques

➤ Milieu fini mono-dimensionnel $K = 5$ $e = 1\text{cm}$

