



Projet de Fin d'Etudes d'Ingénieur ENSEA

CONSTRUCTION DE FAMILLE D'INSTRUMENTS

A VENT VIRTUELS

Étudiant

Thomas Hézard
ENSEA STC 2009

Encadrants IRCAM
équipe Analyse/Synthèse

Rémi Mignot
Thomas Hélie

Professeur ENSEA

Antoine Tauvel

Version 1.2 - mars 2011

Table des matières

I	Structures simulables et simulation temps-réel	9
1	Décomposition d'un instrument en cellules simples	10
1.1	Représentation en cellules MIMO simples	10
1.2	Quadripôles de passage	10
1.3	Découpage des tronçons	11
1.4	Embouchure	14
1.5	Rayonnement	15
1.6	Jonction à N portes	15
1.7	Modèle de switch continu	16
1.8	Modèle de diaphragme et de piston	16
2	Représentation d'état en temps continu	18
2.1	Découpage des tronçons	18
2.2	Embouchure	20
2.3	Rayonnement	20
2.4	Jonction à N portes	21
2.5	Modèle de switch continu	21
2.6	Modèle de diaphragme et de piston	21
2.7	Concaténation de deux cellules	21
2.8	Diagonalisation de la matrice A	22
2.9	Autres modifications des représentation d'état	23
3	Structure simulable	23
3.1	Discrétisation	23
3.2	Débouclage partiel	24
3.3	Fonctionnement des différentes cellules variables	25
3.3.1	Switch continu	25
3.3.2	Piston	26
4	Implémentation Matlab	26
5	Simulation temps-réel	27
5.1	Excitation : modèle de lèvres	27
5.2	Principe de la simulation	29
5.3	Implémentation C++	29
II	Optimisation de profil de tube	30

6	Présentation du problème	31
6.1	Notations	31
6.2	Modèle	31
6.2.1	Tronçon à courbure constante	31
6.2.2	Profil C^1	31
7	Optimisation géométrique	32
7.1	Tronçon caténoïdal sur $x \in [-1, 1]$	32
7.2	Changements de variables	33
7.2.1	Profil cible	34
7.2.2	Profil modèle	34
7.3	Résolution analytique des contraintes C^1	35
7.4	Présentation du problème d'optimisation	35
7.5	Critère	36
7.5.1	Définition du critère	36
7.5.2	Etude du critère lorsque $\Upsilon_n \rightarrow 0$	37
7.6	Contraintes supplémentaires : critère pénalisé, lagrangien, lagrangien augmenté	38
7.6.1	Critère pénalisé	38
7.6.2	Lagrangien	38
7.6.3	Lagrangien augmenté	38
7.7	Implémentation Matlab de l'optimisation géométrique	39
7.7.1	Fonctions non définies en 0	39
7.7.2	Point de départ de l'optimisation et choix de N	39
7.7.3	Problème de minima locaux	39
7.7.4	Résultats	40
7.8	Applications	42
8	Optimisation en impédance	42
8.1	Calcul de l'impédance à partir du modèle	42
8.2	Choix du critère	43
9	Développements d'outils de "LAO"	43
III	Validation du modèle : mesures et simulation	45
10	Problème du sinus glissant	46
11	Campagnes de mesures effectuées à l'IRCAM	47
11.1	Trompette	47
11.2	Cor	48
12	Prototype de pavillon sur-mesure	48

A	Structure simulable	54
A.1	Standardisation des représentations d'état	54
A.2	Débouclage partiel	57
B	Optimisation géométrique	58
B.1	Formules de passage	58
B.1.1	Variables dimensionnées	58
B.1.2	Variables adimensionnées	59
B.2	Détails des calculs de la récurrence sur \mathbf{U}_n	60

Table des figures

1	Quadripôle	10
2	Quadripôles de passage $\mathcal{C}_{(p,PU)}$, $\mathcal{C}_{(UP,p)}$, $\mathcal{C}_{(p,UP)}$ and $\mathcal{C}_{(PU,p)}$	11
3	Décomposition du tube en 7 quadripôles	13
4	A gauche : schéma d'une embouchure. A droite : circuit électrique équivalent.	14
5	Quadripôle de l'embouchure.	14
6	Quadripôle de rayonnement	15
7	Switch continu en position intermédiaire.	16
8	Schéma d'un diaphragme dans un tube	17
9	Quadripôle du diaphragme	17
10	Décomposition d'un piston	18
11	Concatenation de deux cellules en représentation d'état	22
12	Débouclage	25
13	Concaténation algébrique du switch continu	26
14	Modèle de lèvre	28
15	Première série de résultats d'optimisation.	40
16	Deuxième série de résultats d'optimisation.	41
17	Concaténation de 2 quadripôles	43
18	Mesure d'impédance de la trompette	47
19	Différents doigtés de la trompette	48
20	Mesure d'impédance du cor	49
21	Main à l'intérieur du pavillon du cor	49
22	Mesures sur le cor. En haut : sans boucher le pavillon. En bas : en bouchant le pavillon	50
23	Pièce fabriquée sur mesure. A gauche : modèle numérique CATIA. A droite : pièce réalisée	51
24	Courbe d'impédance du prototype	52

Remerciements

Je tiens à remercier en premier lieu *Hugues Vinet*, directeur scientifique de l'IRCAM pour m'avoir accueilli dans son laboratoire. Je tiens à remercier ensuite *Xavier Rodet*, responsable de l'équipe Analyse & Synthèse des Sons dans laquelle j'ai travaillé pendant ces six mois.

Je tiens à remercier *Sylvain Reynal*, professeur à l'ENSEA, qui m'a aidé à entrer en contact avec l'IRCAM.

Je remercie tout particulièrement *Thomas Hélie* et *Rémi Mignot* qui m'ont encadré et qui ont su être présents et disponibles tout au long de mon stage pour répondre aux nombreuses questions que j'ai pu leur poser. Merci à *Rémi Mignot* qui m'a aidé pour la mise en forme et la relecture du présent document.

Je remercie chaleureusement toutes les personnes de l'IRCAM qui m'ont aidé de près ou de loin pendant mon stage, notamment *René Caussé*, *Gérard Bertrand*, *Allain Terrier* et *Frédéric Cornu* qui n'ont jamais été avares de services rendus.

Je remercie enfin très chaleureusement *Jérôme Joubert* qui nous a permis de faire une série de mesures très complète sur son cor et qui a participé activement à ces mesures.

Ce travail a été soutenu par le projet CONSONNES - ANR-05-BLAN-0097-01.

Introduction

Etat de l'art des modélisations physiques de propagation acoustique dans les tubes

La modélisation physique pour la simulation temps-réel des instruments intéresse physiciens et programmeurs depuis de nombreuses années. L'avancée exponentielle des technologies de l'informatique et du traitement du signal permet de concevoir des modèles de plus en plus complexes afin de modéliser au plus près l'acoustique d'un instrument réel.

Mon travail durant ce stage se situe aux premières implémentations de nouveaux modèles de propagation pour la simulation temps-réel des cuivres. Les deux premières hypothèses sous lesquelles les différents travaux de modélisation ont été effectués sont les suivantes :

- L'influence de l'enroulement de l'instrument sur lui-même est négligée.
- Les instruments, une fois déroulés, sont considérés comme axisymétriques.

Ainsi, un instrument est entièrement décrit par le relevé de sa perce, c'est à dire le relevé du rayon du tube le long de son axe de symétrie. C'est cette courbe qui est appelée "profil de tube" dans la suite de ce document.

Les premiers modèles de *Markel* (voir [MG73]), datant de 1976, proposent de modéliser un tube par une concaténation de tubes cylindriques à sections plus ou moins grandes, son profil modèle est alors une fonction constante par morceaux. *Välimäki* (voir [VV95]), en 1994, introduit la prise en compte de portions coniques dans le modèle. Ces deux modèles nécessitent des hypothèses fortes puisque qu'ils impliquent la sphéricité des ondes à l'intérieur des tubes.

En 2002, dans le cadre de sa thèse, *Thomas Hélie* introduit la prise en compte de tubes courbes et des pertes visco-thermiques dans les modèles de propagation pour le temps-réel (voir [TH02]). Ce modèle permet de considérablement réduire les hypothèses : d'une hypothèse de sphéricité des ondes dans le tube, nous passons à une hypothèse locale de quasi-sphéricité des ondes au contact avec la paroi. Un changement de variable judicieux permet de redresser la carte isobare et, du fait de la quasi-sphéricité des ondes à la paroi, de réduire le problème à une seule dimension : l'abscisse curviligne ℓ le long de la paroi interne de l'instrument. L'intégralité des travaux présentés ici se fait sous ces hypothèses et après le changement de variable spatiale.

Les ondes acoustiques à l'intérieur du tube sous ces hypothèses sont régies par le

modèle de *Webster-Lokshin* (pour plus de détails sur ce résultat, voir [HMM07]) :

$$\left[\left(\left(\frac{s}{c_0} \right)^2 + 2\varepsilon(\ell) \left(\frac{s}{c_0} \right)^{\frac{3}{2}} + \Upsilon(\ell) \right) - \partial_\ell^2 \right] \{ r(\ell) P(\ell, s) \} = 0, \quad (1)$$

$$\rho_0 s \frac{U(\ell, s)}{S(\ell)} + \partial_\ell P(\ell, s) = 0, \quad (2)$$

où $s \in \mathbb{C}$ est la variable de *Laplace* ($\Im m(s) = \omega$ est la pulsation), $r(\ell)$ est le rayon du tube, $S(\ell) = \pi r(\ell)^2$ est la section du tube, $\varepsilon(\ell) = \kappa_0 \sqrt{1 - r'(\ell)^2} / r(\ell)$ quantifie les pertes visco-thermiques et $\Upsilon(\ell) = r''(\ell) / r(\ell)$ est la courbure de la paroi. $P(\ell, s)$ et $U(\ell, s)$ sont respectivement la pression et le débit acoustique. Eq. (1) est appelée équation de *Webster*, et eq. (2) est appelée équation d'*Euler*.

La résolution analytique de ces équations n'est possible que pour des courbures Υ constantes. Ceci nous amène à la définition du modèle de tube utilisé tout au long de mes travaux : une concaténation de tronçons à courbure constante. Selon le signe de Υ , le profil de chaque tronçon s'écrit comme une combinaison linéaire de cosinus et sinus ou de cosinus hyperboliques et sinus hyperboliques :

$$\Upsilon < 0 \quad : \quad r(\ell) = A \cos(\sqrt{-\Upsilon} \ell) + B \sin(\sqrt{-\Upsilon} \ell) \quad \text{avec } (A, B) \in \mathbb{R}^2 \quad (3)$$

$$\Upsilon = 0 \quad : \quad r(\ell) = A + B\ell \quad \text{avec } (A, B) \in \mathbb{R}^2 \quad (4)$$

$$\Upsilon > 0 \quad : \quad r(\ell) = A \cosh(\sqrt{\Upsilon} \ell) + B \sinh(\sqrt{\Upsilon} \ell) \quad \text{avec } (A, B) \in \mathbb{R}^2 \quad (5)$$

Objectifs du stage

Mes objectifs durant ce stage étaient multiples. Premièrement, j'ai continué le travail de *Pierre-Damien Dekoninck* (cf [PDD08]) dans l'implémentation d'un constructeur de structures simulables (en Matlab) à partir des travaux de *Rémi Mignot*, basés sur le modèle décrit ci-dessus, et d'un programme de simulation temps-réel de ces structures (en C++). Deuxièmement, j'ai travaillé sur le développement d'outils d'optimisation de profils d'instruments, dont un des but, à terme, est d'apporter de nouveaux outils informatiques aux luthiers. Enfin, j'ai mis en place plusieurs séries de mesures d'impédance sur des instruments réels et sur un pavillon construit sur-mesure à partir d'un profil théorique dans le but de valider les modèles utilisés. Le présent document s'organise dans cet ordre.

Première partie

Structures simulables et simulation temps-réel

Le sujet principal de ce stage est la simulation temps-réel de cuivres. Cette partie décrit comment construire une structure simulable à partir de la description d'un instrument puis comment simuler cette structure en temps-réel. Les travaux de *Rémi Mignot* et *Pierre-Damien Dekoninck* sont résumés ici sans détailler les calculs qui ont amené les différents résultats. Le lecteur intéressé pourra se reporter à [RM06] et [PDD08] pour plus de détails sur les méthodes ayant conduit à ces résultats. Les travaux que j'ai effectués pendant mon stage sont, en revanche, plus détaillés, ici ou en annexe.

1 Décomposition d'un instrument en cellules simples

1.1 Représentation en cellules MIMO simples

A partir du modèle décrit plus haut, nous pouvons décomposer d'un point de vue acoustique un instrument en plusieurs éléments simples qui sont raccordés les uns aux autres. Nous pouvons regrouper ces éléments en deux grandes familles : les statiques et les variables.

Les éléments statiques sont les suivants :

- Tronçon de tube à courbure constante (droit, conique ou courbé)
- Embouchure
- Rayonnement
- Jonction de N tubes

Les éléments variables sont les suivants :

- Modèle de switch continu
- Modèle de diaphragme
- Modèle de piston

Les travaux de *Rémi Mignot* de ces dernières années permettent de décomposer chacun de ces éléments en une ou plusieurs cellule(s) MIMO (quadripôles pour la plupart). Chaque cellule contient une matrice de fonctions de transfert reliant ses différentes entrées et sorties. Nous représentons donc une cellule quadripolaire par quatre fonctions de transfert, deux transmissions et deux réflexions : $T^+(s), T^-(s), R^l(s), R^r(s)$ (cf figure 1). Les entrées et les sorties des cellules sont des états acoustiques, pouvant être par exemple une pression, un débit ou une fonction d'onde positive (propagation gauche-droite) ou négative (propagation droite-gauche).

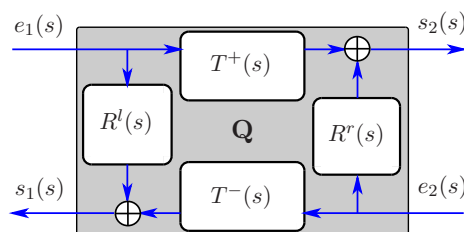


FIGURE 1 – Quadripôle

1.2 Quadripôles de passage

Afin de respecter la causalité et de faciliter le branchement des éléments entre eux, nous avons décidé de choisir aux extrémités de chaque élément un état acoustique indépendant de l'élément lui-même : le couple des ondes planes progressives positives et négatives $[p^+(\ell, s), p^-(\ell, s)]$. Nous introduisons dans ce but un tube virtuel de référence de rayon r_c choisi empiriquement (dans le but d'optimiser le conditionnement numérique) et de longueur nulle. Chaque élément est relié à chacune de ses extrémités à ce tube de réf-

rence. Nous pouvons ainsi définir les matrices de passage entre le couple pression/débit $[P(\ell, s), U(\ell, s)]$ et le couple d'ondes planes progressives $[p^+(\ell, s), p^-(\ell, s)]$:

$$\begin{bmatrix} p^+(\ell, s) \\ p^-(\ell, s) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & Z_c \\ 1 & -Z_c \end{bmatrix} \begin{bmatrix} P(\ell, s) \\ U(\ell, s) \end{bmatrix}. \quad (6)$$

où Z_c est l'impédance caractéristique du tube de référence, calculée avec $c_0 = 344 \text{ m.s}^{-1}$ la vitesse du son, $\rho_0 = 1.2 \text{ kg.m}^{-3}$ la densité de l'air et $S_c = \pi r_c^2$ la surface du tube de référence :

$$Z_c = \frac{\rho_0 c_0}{S_c}, \quad Y_c = \frac{1}{Z_c}. \quad (7)$$

A partir de ces formules de passage, nous pouvons construire les différents quadripôles de passage qui représentent les différents types d'interface entre un élément et le tube de référence. Ces quadripôles sont présentés sur la figure 2.

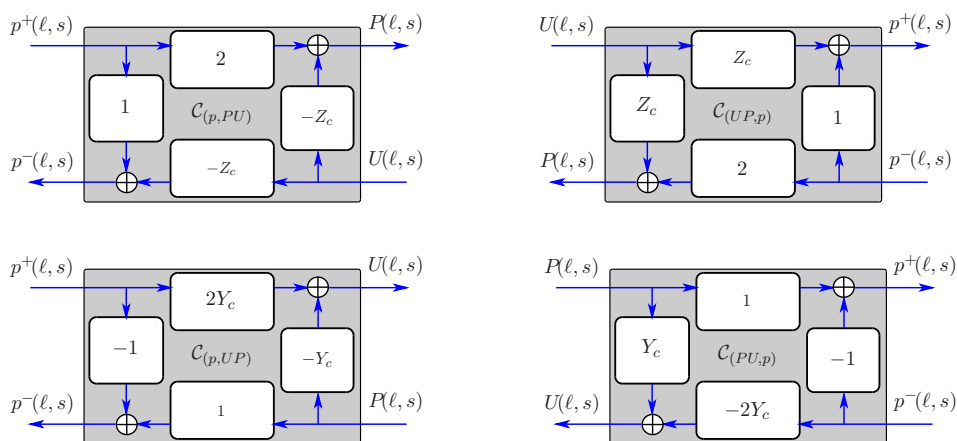


FIGURE 2 – Quadripôles de passage $\mathcal{C}_{(p,PU)}$, $\mathcal{C}_{(UP,p)}$, $\mathcal{C}_{(p,UP)}$ and $\mathcal{C}_{(PU,p)}$.

Nous allons maintenant présenter la décomposition de chaque élément en cellules MIMO en prenant soin d'intégrer les quadripôles de passage à ces décompositions

1.3 Découpage des tronçons

D'après les équations (3), (4) et (5), un tronçon est caractérisé par cinq paramètres : sa longueur L , sa courbure Υ , les deux paramètres linéaires A et B et le coefficient des pertes visco-thermiques κ_0 . Les paramètres A et B peuvent être facilement liés linéairement aux rayons à gauche du tronçon r_l et à droite du tronçon r_r . Un tronçon peut être décomposé en sept quadripôles, comme le montre la figure 3, chacun des quadripôles étant défini en fonction des paramètres qui viennent d'être cités. Les sept quadripôles comprennent deux quadripôles \mathbf{Q}_a modélisant les ruptures de section à gauche et à droite du tronçon, deux quadripôles \mathbf{Q}_s modélisant les ruptures de pente à gauche et à droite du tronçon, deux quadripôles \mathbf{Q}_{cl} modélisant les ruptures de courbure à gauche et à droite du tronçon

ainsi que les pertes visco-thermique, et un quadripôle de propagation \mathbf{Q}_P représentant le retard et l'amortissement de l'onde qui se propage dans le tronçon.

Quadripôles \mathbf{Q}_a

Les quadripôles \mathbf{Q}_a^l et \mathbf{Q}_a^r représentent les effets de rupture de section en entrée et en sortie du tronçon. k_l et k_r sont les coefficients de réflexion causée par le branchement du tube sur le tube de référence, ils s'expriment en fonction des sections du tube à ses extrémités $S_l = \pi r_l^2$ et $S_r = \pi r_r^2$ ou des impédances caractéristiques $Z_l = \rho_0 c_0 / S_l$ et $Z_r = \rho_0 c_0 / S_r$.

$$k_l = \frac{S_c - S_l}{S_l + S_c} = \frac{Z_l - Z_c}{Z_l + Z_c} \quad (8)$$

$$k_r = \frac{S_c - S_r}{S_r + S_c} = \frac{Z_r - Z_c}{Z_r + Z_c} \quad (9)$$

Quadripôles \mathbf{Q}_s

Les quadripôles \mathbf{Q}_s^l et \mathbf{Q}_s^r représentent les effets de rupture de pente à gauche et à droite du tronçon. Ils s'expriment à l'aide des fonctions $R_l^s(s)$ et $R_r^s(s)$ qui dépendent de la pente à gauche $r'_l = r'(\ell = 0)$ et à droite $r'_r = r'(\ell = L)$ du tronçon.

$$R_l^s(s) = \frac{\alpha_l}{s - \alpha_l}, \text{ avec } \alpha_l = -\frac{c_0}{2} \frac{r'_l}{r_l} \quad (10)$$

$$R_r^s(s) = \frac{\alpha_r}{s - \alpha_l}, \text{ avec } \alpha_r = +\frac{c_0}{2} \frac{r'_r}{r_r} \quad (11)$$

$$(12)$$

Quadripôles \mathbf{Q}_{cl}

Ces quadripôles, avec la fonction $R(s)$, prennent en considération la courbure constante du tronçon ainsi que les pertes visco-thermiques dans le tronçon.

$$R(s) = \frac{s/c_0 - \Gamma(s)}{s/c_0 + \Gamma(s)} \quad (13)$$

où $\Gamma(s)$, issue du modèle de *Webster-Lokshin* est défini par :

$$\Gamma(s) = \sqrt{\left(\frac{s}{c_0}\right)^2 + 2\epsilon\left(\frac{s}{c_0}\right)^{\frac{3}{2}} + \Upsilon} \quad (14)$$

Quadripôles \mathbf{Q}_P

La fonction de transfert $T(s)$ représente le retard et l'amortissement de l'onde progressive et peut s'écrire sous la forme :

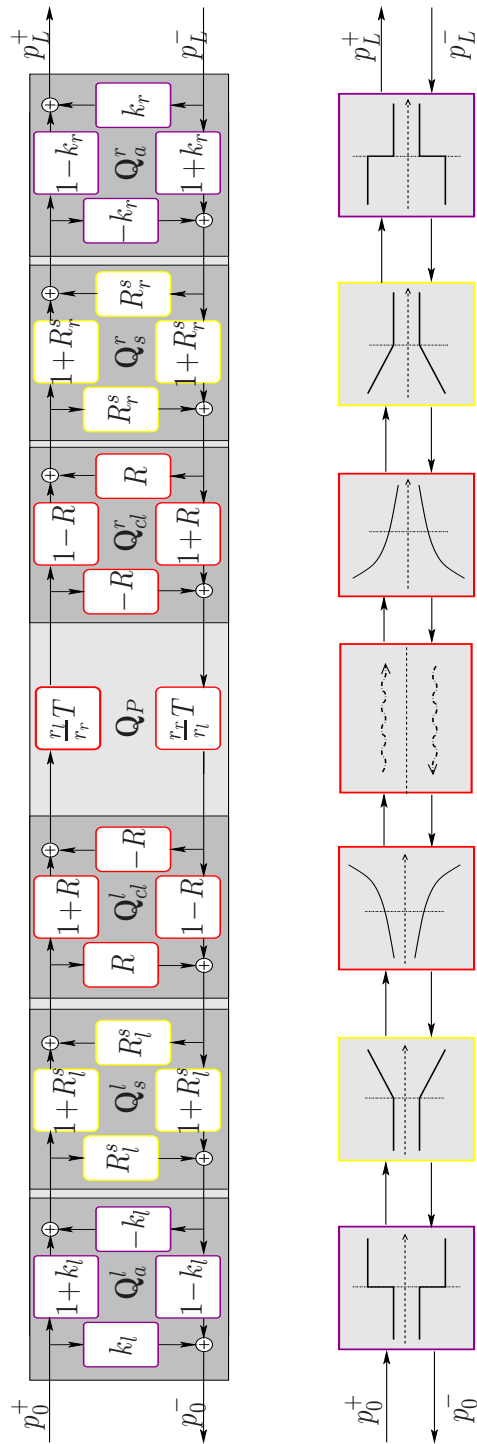


FIGURE 3 – Décomposition du tube en 7 quadripôles

$$T(s) = e^{-\Gamma(s)L} = D(s)e^{-s/c_0L} \quad (15)$$

$$\text{avec } D(s) = e^{-(\Gamma(s)-s/c_0)L} \quad (16)$$

1.4 Embouchure

Une embouchure est composée de deux parties (cf figure 4) : une cuvette, décrite par son volume V_m , et une queue, décrite par sa longueur L_s et son rayon équivalent r_s . Par analogie avec un circuit électrique (cf [FR91]), l'embouchure est modélisée par une capacité acoustique C_a , une masse acoustique M_a et une résistance acoustique R_a dont les expressions sont :

$$C_a = \frac{V_m}{\rho_0 c_0^2}, M_a = \frac{\rho_0 L_s}{\pi r_s^2} \text{ et } R_a = \frac{8\mu L_s}{\pi r_s^4}, \quad (17)$$

où $\mu = 1.8 \cdot 10^{-5} \text{ kg} \cdot (\text{m} \cdot \text{s})^{-1}$ est le coefficient de viscosité de cisaillement.

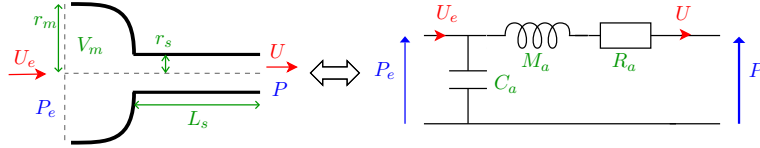


FIGURE 4 – A gauche : schéma d'une embouchure. A droite : circuit électrique équivalent.

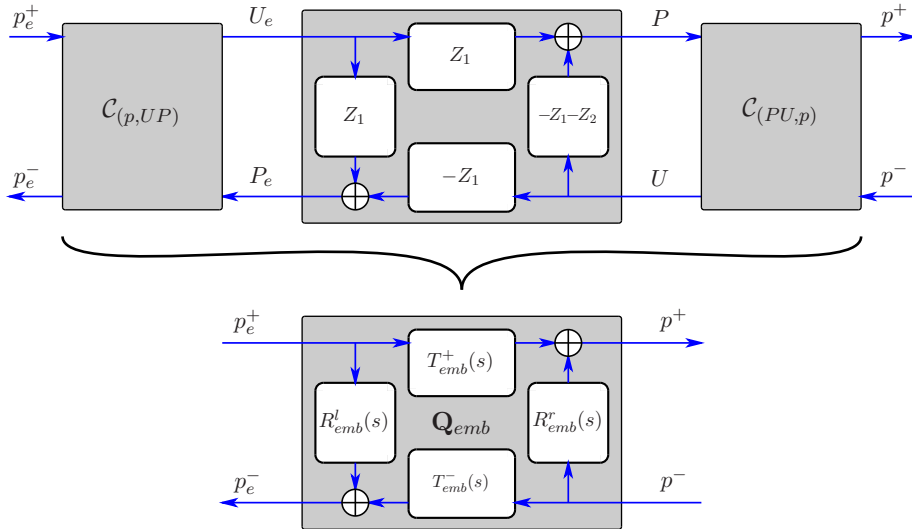


FIGURE 5 – Quadripôle de l'embouchure.

Nous obtenons alors le quadripôle en variable $[P, U]$ de la figure 5 que nous concaténons avec les quadripôles de passage pour donner les fonctions de transmission et de

réflexion en ondes progressives suivantes :

$$\begin{cases} R_{emb}^l(s) = 2(M_a Y_c^2 s + (1 + R_a Y_c) Y_c) / d(s) - 1, \\ T_{emb}^-(s) = 2Y_c / d(s), \\ T_{emb}^+(s) = 2Y_c / d(s), \\ R_{emb}^r(s) = 2(-C_a s - Y_c) / d(s) + 1, \end{cases} \quad (18)$$

avec $d(s) = Y_c M_a C_s s^2 + (C_a + Y_c R_a C_a + M_a Y_c^2) s + (2 + R_a Y_c) Y_c$.

1.5 Rayonnement

Plusieurs modèles ont été établis par *Thomas Hélie* (cf. [TH02]), tous basés sur le rayonnement d'une portion de sphère pulsante. Nous utilisons ici le modèle passe-haut d'ordre 2 qui est représenté en variable $[P, U]$ par une impédance de rayonnement Z_{ray} (cf. figure 6) :

$$Z_{ray}(s) = Z_s \frac{\alpha_r \frac{s}{\omega_r} + \left(\frac{s}{\omega_r}\right)^2}{1 + 2\xi_r \frac{s}{\omega_r} + \left(\frac{s}{\omega_r}\right)^2}. \quad (19)$$

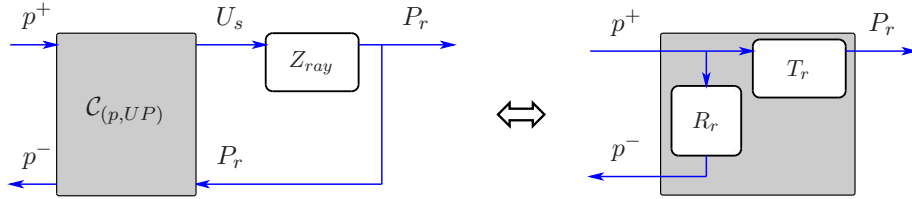


FIGURE 6 – Quadripôle de rayonnement

Après concaténation avec un quadripôle de passage pour retrouver nos variables $[p^+, p^-]$, nous obtenons un quadripôle constitué d'une seule réflexion et d'une seule transmission (les deux autres fonctions de transfert sont nulles) notées R_r et T_r

$$R_r(s) = \frac{Z_{ray}(s) - Z_c}{Z_{ray}(s) + Z_c} \quad (20)$$

$$T_r(s) = R_r(s) + 1. \quad (21)$$

1.6 Jonction à N portes

Nous considérons maintenant une jonction de N tubes. Nous modélisons cette jonction sans dimension spatiale en respectant les continuités de pression et de débit. Nous obtenons alors des équations analogues aux lois de *Kirchhoff* qui s'écrivent :

$$\begin{cases} P_n = P_1, \text{ pour } 1 \leq n \leq N, \\ \sum_{n=1}^N U_n = 0. \end{cases}$$

Le passage en ondes progressives et la résolution du système linéaire nous amène à l'équation suivante

$$P^- = J_N P^+, \text{ avec } J_N = \frac{2}{N} \mathbb{1}_N - Id_N, \quad (22)$$

où $P^+ = [p_1^+, p_2^+, \dots, p_N^+]^T$, $P^- = [p_1^-, p_2^-, \dots, p_N^-]^T$, Id_N est la matrice identité $N \times N$ et $\mathbb{1}_N$ est la matrice $N \times N$ telle que $\mathbb{1}_N(i, j) = 1, \forall (i, j) \in [1, N]^2$.

1.7 Modèle de switch continu

La première cellule variable que j'ai mise en place est le "switch continu". Il s'agit d'une jonction variable à trois portes qui relie la première porte à l'une des deux autres portes et qui permet de passer d'une porte à l'autre. Le passage d'une porte à l'autre se fait de façon continue (le switch peut se trouver dans une position intermédiaire dans laquelle les deux portes sont partiellement ouvertes). Le switch continu est donc représenté par une unique paramètre g variant dans l'intervalle $[0, 1]$, $g = 0$ représente une liaison directe de la porte 1 à la porte 2 et $g = 1$ représente une liaison directe de la porte 1 à la porte 3. On peut représenter son fonctionnement par le schéma présenté figure 7 et par le système linéaire suivant :

$$Y = \begin{pmatrix} 0 & g & 1-g \\ g & 0 & 0 \\ 1-g & 0 & 0 \end{pmatrix} U, \quad (23)$$

où $U = [p_1^+, p_2^-, p_3^-]^T$ et $Y = [p_1^-, p_2^+, p_3^+]^T$.

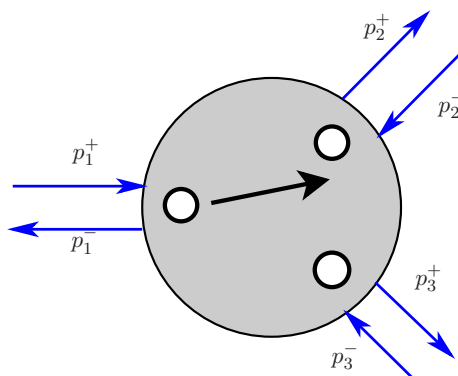


FIGURE 7 – Switch continu en position intermédiaire.

1.8 Modèle de diaphragme et de piston

Diaphragme

Physiquement, un diaphragme peut être considéré comme un tube de rayon égal à $r_d = k.r$, où r est le rayon du tube dans lequel se trouve le diaphragme et k un réel décrivant l'ouverture du diaphragme et variant dans l'intervalle $[0, 1]$, et dont la longueur

tend vers 0 (cf figure 8). Le diaphragme est modélisé par un quadripôle paramétré par cet unique coefficient k , présenté sur la figure 9. Si $k = 1$, le diaphragme est entièrement fermé et si $k = 0$, le diaphragme est entièrement ouvert.

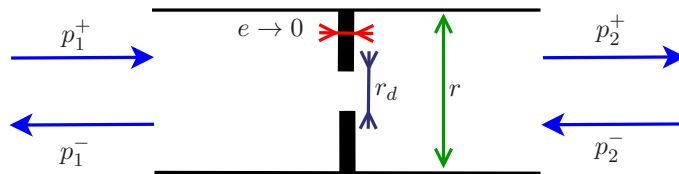


FIGURE 8 – Schéma d'un diaphragme dans un tube

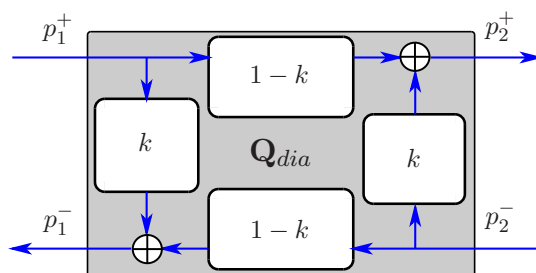


FIGURE 9 – Quadripôle du diaphragme

Piston

Le modèle de piston mis en place par *Rémi Mignot* et moi-même est composé de cinq cellules selon le schéma présenté en figure 10. Ces cinq cellules comprennent deux jonctions à trois portes (J_1 et J_2) et trois diaphragmes (D_1 , D_2 et D_3). Le piston est un élément qui possède quatre connexions externes : C_1 , C_2 , C_3 et C_4 .

Un seul paramètre k permet de décrire le fonctionnement du piston. Les diaphragmes D_1 et D_2 admettent comme paramètre k et D_3 admet comme paramètre $1 - k$. Ainsi, quand $k = 0$ les diaphragmes D_1 et D_2 sont fermés et le piston se comporte comme une liaison directe entre les connexions C_1 et C_4 , et quand $k = 1$, cette connexion est intégralement coupée et la totalité du débit passe par la portion d'instrument qui est branchée entre les connexions C_2 et C_3 . Ici encore, le paramètre k peut prendre toutes les valeurs réelles comprises entre 0 et 1 et le passage d'une position à l'autre se fait de façon continue.

La représentation en cellules MIMO est déduite de la représentation des différents éléments constituant le piston, nous verrons par la suite comment concaténer des cellules entre elles pour obtenir la représentation du piston sous la forme d'une unique cellule.

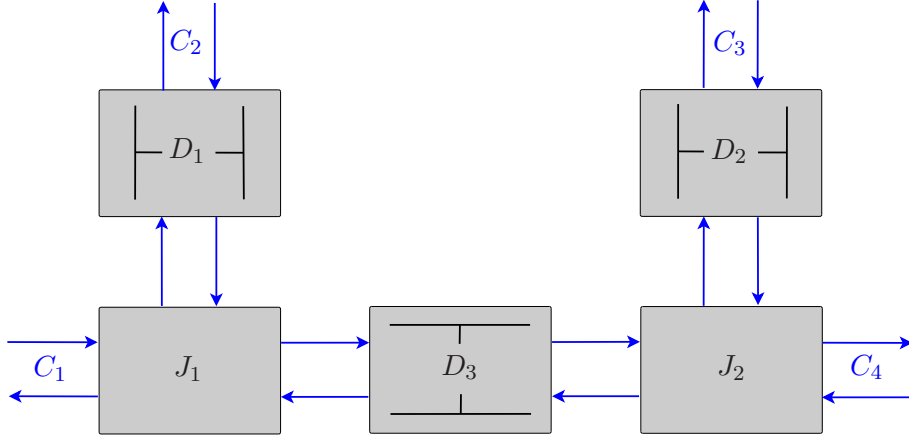


FIGURE 10 – Décomposition d'un piston

2 Représentation d'état en temps continu

Dans le but de construire une représentation discrétisée simulable en temps réel, nous commençons par écrire toutes les cellules en représentation d'état en temps continu, sous la forme générale :

$$\begin{cases} sX = \mathbf{A}X + \mathbf{B}U \\ Y = \mathbf{C}X + \mathbf{D}U \end{cases} \quad (24)$$

où U est le vecteur des variables d'entrée du système, Y le vecteur des variables de sortie, X l'état interne du système et \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} les matrices du système. Nous explicitons ici ces matrices pour les différentes cellules présentées ci-dessus.

Par convention, le vecteur des variables d'entrée d'un système est composé des variables d'entrée de la cellule dans l'ordre des numéros de porte, de même pour le vecteur des variables de sortie.

Les calculs des représentations d'état des éléments statiques ayant déjà été fait précédemment, seuls les résultats obtenus sont présentés ici (voir [PDD08] pour plus de détails). Je détaillerai en revanche d'avantage les représentations d'état des éléments variables.

2.1 Découpage des tronçons

Cellules \mathbf{Q}_a

Les filtres présents dans le quadripôle \mathbf{Q}_a ne possèdent pas de dynamique (les fonctions de transfert sont indépendantes de la variable de Laplace s), la représentation correspondante se résume donc à une matrice \mathbf{D} déduite directement du quadripôle. La cellule \mathbf{Q}_a^l s'écrit :

$$\begin{aligned} \mathbf{A} &= \square, & \mathbf{B} &= \square, \\ \mathbf{C} &= \square, & \mathbf{D} &= \begin{bmatrix} k_l & 1 - k_l \\ 1 + k_l & -k_l \end{bmatrix}. \end{aligned} \quad (25)$$

La matrice \mathbf{D} de la cellule \mathbf{Q}_a^r se déduit directement de la précédente en remplaçant k_l par $-k_r$.

Cellules \mathbf{Q}_s

Les cellules \mathbf{Q}_s contiennent une fonction de transfert du premier ordre. La représentation d'état de \mathbf{Q}_s^l est la suivante :

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \alpha_l \end{bmatrix}, & \mathbf{B} &= \begin{bmatrix} 1 & 1 \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \alpha_l \\ \alpha_l \end{bmatrix}, & \mathbf{D} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned} \quad (26)$$

On déduira aisément \mathbf{Q}_s^r en remplaçant α_l par α_r .

Cellules \mathbf{Q}_{cl}

La présence de $\sqrt{\Gamma(s)}$ dans les cellules \mathbf{Q}_{cl} introduit des fonctions de transfert irrationnelles comportant des lignes continues de singularités, nommées coupures, qui relient un point de branchement complexe à l'infini. Ces systèmes à dimension infinie appelés *représentations diffusives* (voir [GM98] et [DM98]) peuvent être approximés (voir [GM98] et [HMM07bis]) par un système fini de pôles que l'on place sur les coupures. Le choix du placement des pôles ainsi que l'optimisation des poids de ces pôles sont issus des travaux de *Rémi Mignot*, détaillés dans [RM06]. Nous obtenons comme approximation de $R(s)$ en dimension finie :

$$\tilde{R}(s) = \sum_{j=1}^L \frac{\mu_j^R}{s + \xi_j} + \sum_{j=1}^L \left(\frac{\omega_j^R}{s + \gamma_j} + \frac{\bar{\omega}_j^R}{s + \bar{\gamma}_j} \right),$$

où les $-\xi_j$ sont les pôles placés sur la coupure le long du demi axe \mathbb{R}^- avec μ_j leurs poids, et les couples $(-\omega_j, -\bar{\omega}_j)$ sont les couples de pôles complexes conjugués du reste du plan avec $(\gamma_j, \bar{\gamma}_j)$ leurs poids.

La représentation d'état de ces cellules s'écrit alors :

$$\begin{aligned} \mathbf{A} &= \text{diag}([\xi_1, \dots, \xi_L, \gamma_1, \dots, \gamma_M, \bar{\gamma}_1, \dots, \bar{\gamma}_M]), \\ \mathbf{B} &= \begin{bmatrix} 1, & \dots & 1 \\ -1, & \dots & -1 \end{bmatrix}^T, \\ \mathbf{C} &= \begin{bmatrix} \mu_1^R, \dots, \mu_L^R, w_1^R, \dots, w_M^R, \bar{w}_1^R, \dots, \bar{w}_M^R \\ \mu_1^R, \dots, \mu_L^R, w_1^R, \dots, w_M^R, \bar{w}_1^R, \dots, \bar{w}_M^R \end{bmatrix}, \\ \mathbf{D} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned} \quad (27)$$

Cellules Q_P

Les retards dûs au terme e^{-s/c_0L} seront simulés en temps discret par des tampons circulaires. La fonction de transfert $D(s)$ est approximée, par la même méthode que $R(s)$, par un système à dimension finie :

$$\begin{aligned} \mathbf{A} &= \text{diag}([\xi_1, \dots, \xi_L, \gamma_1, \dots, \gamma_M, \bar{\gamma}_1, \dots, \bar{\gamma}_M]), \\ \mathbf{B} &= [1, \dots, 1]^T, \\ \mathbf{C} &= [\mu_1^D, \dots, \mu_L^D, w_1^D, \dots, w_M^D, \bar{w}_1^D, \dots, \bar{w}_M^D], \\ \mathbf{D} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \end{aligned} \tag{28}$$

2.2 Embouchure

La représentation d'état de l'embouchure s'écrit :

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a_0 & -a_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_0 & -a_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \\ \mathbf{C} &= \frac{2}{M_a C_a} \begin{bmatrix} 1+R_a Y_c & 1 \\ M_a Y_c & 0 \\ 1 & -1 \\ 0 & -\frac{C_a}{Y_c} \end{bmatrix}^T, \quad \mathbf{D} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \\ \text{avec } a_0 &= \frac{2+R_a Y_c}{M_a C_a}, \text{ et } a_1 = \frac{C_a/Y_c + R_a C_a + M_a Y_c}{M_a C_a}. \end{aligned} \tag{29}$$

Notons qu'ici la matrice A n'est pas diagonale, il sera nécessaire de la diagonaliser (cf I.2.8).

2.3 Rayonnement

La représentation d'état du rayonnement s'écrit :

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ \mathbf{C} &= \frac{2\omega_r Z_c Z_s}{(Z_s + Z_c)^2} \begin{bmatrix} -\omega_r \\ (\alpha_r - 2\xi_r) \end{bmatrix}^T, \quad \mathbf{D} = \begin{bmatrix} \frac{Z_s - Z_c}{Z_s + Z_c} \\ \frac{2Z_s}{Z_s + Z_c} \end{bmatrix}. \\ \text{avec } a_0 &= \frac{Z_c \omega_r^2}{Z_s + Z_c} \text{ et } a_1 = \frac{\omega_r (2Z_c \xi_r + Z_s \alpha_r)}{Z_s + Z_c}. \end{aligned} \tag{30}$$

Même remarque que pour l'embouchure.

2.4 Jonction à N portes

Les jonctions à N portes ne possèdent pas de dynamique, leur représentation d'état s'écrit donc directement :

$$\begin{aligned}\mathbf{A} &= \mathbb{0}, & \mathbf{B} &= \mathbb{0}, \\ \mathbf{C} &= \mathbb{0}, & \mathbf{D} &= \frac{2}{N}\mathbf{1}_N - Id_N.\end{aligned}\tag{31}$$

2.5 Modèle de switch continu

Afin de minimiser la quantité de calculs lors des changements de paramètre d'un élément variable, j'ai fait en sorte que la représentation d'état de chaque élément variable soit sans dynamique. On les déduit très simplement à partir des systèmes linéaires qui décrivent leur comportement.

Pour le switch continu, on obtient :

$$\begin{aligned}\mathbf{A} &= \mathbb{0}, & \mathbf{B} &= \mathbb{0}, \\ \mathbf{C} &= \mathbb{0}, & \mathbf{D} &= \begin{bmatrix} 0 & g & 1-g \\ g & 0 & 0 \\ 1-g & 0 & 0 \end{bmatrix}.\end{aligned}\tag{32}$$

2.6 Modèle de diaphragme et de piston

Diaphragme

Comme pour le switch continu, la représentation d'état du diaphragme est immédiate :

$$\begin{aligned}\mathbf{A} &= \mathbb{0}, & \mathbf{B} &= \mathbb{0}, \\ \mathbf{C} &= \mathbb{0}, & \mathbf{D} &= \begin{bmatrix} k & 1-k \\ 1-k & k \end{bmatrix}.\end{aligned}\tag{33}$$

Piston

Les représentations d'état de chaque cellule composant le piston sont calculées indépendamment. Une fois les cellules calculées et discrétisées, elles seront concaténées selon la méthode décrite ci-après afin que le piston soit bien modélisé par une seule cellule à quatre portes que nous pourrons simuler en temps-réel.

2.7 Concaténation de deux cellules

Comme on peut le voir sur la partie supérieure de la figure 11, il apparaît une boucle instantanée entre deux cellules adjacentes. En représentation "cellule MIMO", cette boucle est formée par la réflexion $R^r(s)$ de la cellule de gauche et la réflexion $R^l(s)$ de la cellule de droite. Cette boucle instantanée n'est évidemment pas simulable en temps-réel, il est donc nécessaire de la supprimer. La solution la plus évidente pour la supprimer est la concaténation des deux cellules entre elles.

La concaténation de deux cellules connectées entre elles se fait grâce à l’algèbre mise en place par *Stéphane Tassart* (voir [ST98] pour détails). La figure 11 présente la concaténation de deux systèmes quadripôlaire. L’extension des fomules de concaténation de deux systèmes quadripôlaire à deux systèmes multi-portes a été mise en place par *Pierre-Damien Dekoninck*, le lecteur pourra se reporter à [PDD08] pour retrouver les formules de concaténation.

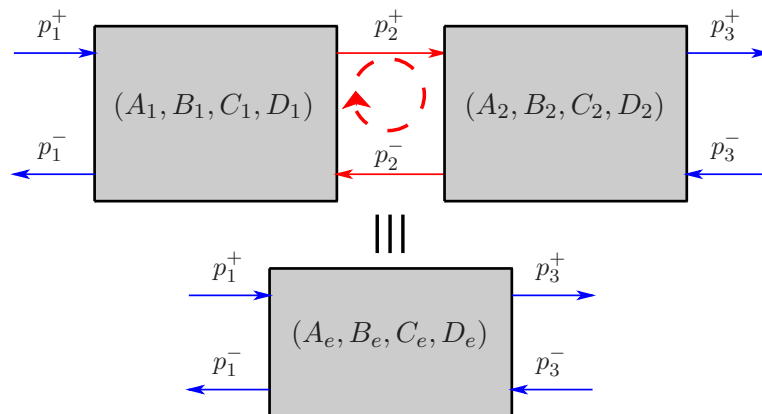


FIGURE 11 – Concaténation de deux cellules en représentation d’état

Les matrices des représentations d’état des cellules variables sont exprimées algébriquement en fonction de leurs paramètres. La concaténation d’une cellule variable avec une autre cellule (variable ou statique) doit donc être effectuée algébriquement (afin d’obtenir, dans l’idéal, une seule cellule représentant l’instrument en entier et dont les matrices s’expriment en fonction des différents paramètres variables de l’instrument). Il apparaît malheureusement très rapidement que cette solution n’est pas réalisable en raison de la complexité des calculs algébriques à effectuer. Nous ne concaténons donc pas les cellules variables pour l’instant et nous verrons plus tard comment résoudre le problème de la boucle instantanée pour ces cellules.

2.8 Diagonalisation de la matrice \mathbf{A}

Il est possible, notamment après une concaténation, que la matrice \mathbf{A} de la représentation d’état d’un système ne soit plus diagonale. Nous souhaitons alors la diagonaliser afin de gagner considérablement en temps de calcul. Comme nous travaillons dans \mathbb{C} , les matrices sont presque toujours diagonalisables. Après diagonalisation, le système s’écrit, en fonction de la matrice de passage \mathbf{P} :

$$\begin{cases} sX' &= (\mathbf{P}^{-1}\mathbf{A}\mathbf{P}) X' + (\mathbf{P}^{-1}\mathbf{B}) U, \\ Y &= (\mathbf{C}\mathbf{P}) X' + \mathbf{D} U. \end{cases} \quad (34)$$

où $X' = \mathbf{P}^{-1}X$.

2.9 Autres modifications des représentation d'état

Afin d'assurer la stabilité (entre autres) des différentes cellules, l'observabilité et la contrôlabilité de chaque cellule sont vérifiées à l'aide des fonctions Matlab correspondantes.

Il est par ailleurs nécessaire de vérifier l'absence de pôle complexe célibataire (chaque pôle complexe doit avoir son conjugué) afin de s'assurer que la sortie du système soit bien réelle. J'ai implémenté pour cela un rapide algorithme qui permet de supprimer les parties imaginaires "épsilonesques" issues de la précision Matlab et de ne garder que les véritables paires de pôles complexes conjugués.

3 Structure simulable

3.1 Discrétisation

Une fois mise sous forme de représentation d'état avec une matrice \mathbf{A} diagonale, nous pouvons aisément discrétiser une cellule. En effet, le système se comporte comme J systèmes du premier ordre avec les pôles $\mathbf{A}_{j,j}$:

$$sX_j = a_j X_j + V_j, \quad \text{pour } 1 \leq j \leq J, \quad (35)$$

où $a_j = \mathbf{A}_{j,j}$ et $V_j = \sum_{n=1}^N \mathbf{B}_{(j,n)} U_n$.

De nombreux schémas de discrétisation peuvent être envisagés, le lecteur intéressé pourra trouver plus d'informations dans [FPW90]. Avec le schéma de discrétisation par interpolation linéaire utilisé ici, ces J équations nous amènent à la représentation d'état discrétisée suivante :

$$\begin{cases} zX^d &= \mathbf{A}^d X^d + (z\mathbf{\Lambda}_1 + \mathbf{\Lambda}_0)U^d, \\ Y^d &= \mathbf{C}X^d + \mathbf{D}U^d, \end{cases} \quad (36)$$

où

$$\begin{cases} \mathbf{\Lambda}_l = \text{diag} \left(\{ \lambda_{(j,l)} \}_{1 \leq j \leq J} \right) \mathbf{B} \text{ pour } l \in \{0, 1\} \\ \mathbf{A}^d = \text{diag} \left(\{ \alpha_j \}_{1 \leq j \leq J} \right) \end{cases} \quad (37)$$

et

$$\begin{cases} \alpha_j &= e^{a_j T_s}, \\ \lambda_{(j,0)} &= -\frac{1 - \alpha_j}{a_j^2 T_s} - \frac{1}{a_j}, \\ \lambda_{(j,1)} &= \frac{1 - \alpha_j}{a_j^2 T_s} + \frac{\alpha_j}{a_j}. \end{cases} \quad (38)$$

Les équations (36) ne sont pas de la forme standard des équations dynamiques des représentations d'état car, en raison de la présence de z dans le terme de droite de la première équation, X_n dépend de U_n . Nous pouvons résoudre ce problème avec un changement d'état. L'annexe A.1 présente la méthode générale que j'ai mise en place pour

résoudre le problème quelque soit le schéma de discrétisation. Dans le cas de l'interpolation linéaire, le nouvel état est $W^d = X^d - \mathbf{\Lambda}_1 U^d$, ce qui nous mène au système suivant :

$$\begin{cases} zW^d &= \mathbf{A}^d W^d + \mathbf{B}^d U^d \\ Y^d &= \mathbf{C}^d W^d + \mathbf{D}^d U^d \end{cases}, \quad (39)$$

avec

$$\begin{cases} \mathbf{B}^d &= \mathbf{A}^d \mathbf{\Lambda}_1 + \mathbf{\Lambda}_0 \\ \mathbf{C}^d &= \mathbf{C} \\ \mathbf{D}^d &= \mathbf{C} \mathbf{\Lambda}_1 + \mathbf{D} \end{cases}. \quad (40)$$

3.2 Débouclage partiel

Le problème de la boucle instantanée dans le cas des cellules variables ne peut être résolu que pour une liaison entre une cellule variable et une cellule statique. Je traite ici le cas d'une cellule statique quadripolaire liée à une cellule variable quelconque. L'extension au cas d'une cellule statique quelconque nécessite une étude plus approfondie.

La solution consiste à séparer la cellule statique reliée à la cellule variable en deux parties : l'une, qui contient la dynamique, que l'on nomme "historique" et l'autre, qui ne contient pas de dynamique, que l'on nomme "directe". La partie "historique" calcule en fait la réponse libre (par rapport à l'entrée reliée à la cellule variable) anticipée, c'est-à-dire à l'instant suivant l'instant de travail. Ainsi la liaison entre la cellule historique et la cellule directe est simulable en temps-réel, il ne reste plus qu'à concaténer la cellule directe à la cellule variable. Cependant, cette opération nécessite de raccourcir d'un échantillon le retard relié à l'entrée de la cellule statique qui n'est pas reliée à la cellule variable. Il en découle que la séparation d'une cellule pourra se faire si et seulement si cette cellule est reliée à un retard d'au moins un échantillon sur la porte opposée à celle sur laquelle on souhaite effectuer le débouclage. Après séparation, la partie directe (entièrement décrite par un seul coefficient réel) est fusionnée avec la cellule variable et la liaison entre la partie historique et la nouvelle cellule concaténée est simulable en temps-réel. La séparation se fait selon le schéma de la figure 12 et la méthode est décrite en détail en annexe A.2.

Pour une cellule statique quadripolaire, la séparation se fait de la façon suivante :

Système original :

$$\begin{cases} X_{n+1} = \mathbf{A}X_n + \mathbf{B}U_n = \mathbf{A}X_n + \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} \\ Y_n = \mathbf{C}X_n + \mathbf{D}U_n = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} X_n + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} \end{cases}$$

Système historique ayant pour vecteur d'entrée $E_n = [u_1(n+1) \ u_2(n)]^T$, pour vecteur de sortie $W_n = [y_1(n) \ y_2^h(n+1)]^T$ et pour vecteur d'état $X'_n = [X_n^T \ u_1(n)]^T$ avec $y_2^h(n) = \mathbf{C}_2 X_n + D_{21}.u_1(n)$:

$$\begin{cases} X'_{n+1} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ 0 & 0 \end{bmatrix} X'_n + \begin{bmatrix} 0 & \mathbf{B}_2 \\ 1 & 0 \end{bmatrix} E_n \\ W_n = \begin{bmatrix} \mathbf{C}_1 & D_{11} \\ \mathbf{C}_2 \mathbf{A} & \mathbf{C}_2 \mathbf{B}_1 \end{bmatrix} X'_n + \begin{bmatrix} 0 & D_{12} \\ D_{21} & \mathbf{C}_2 \mathbf{B}_2 \end{bmatrix} E_n \end{cases} \quad (41)$$

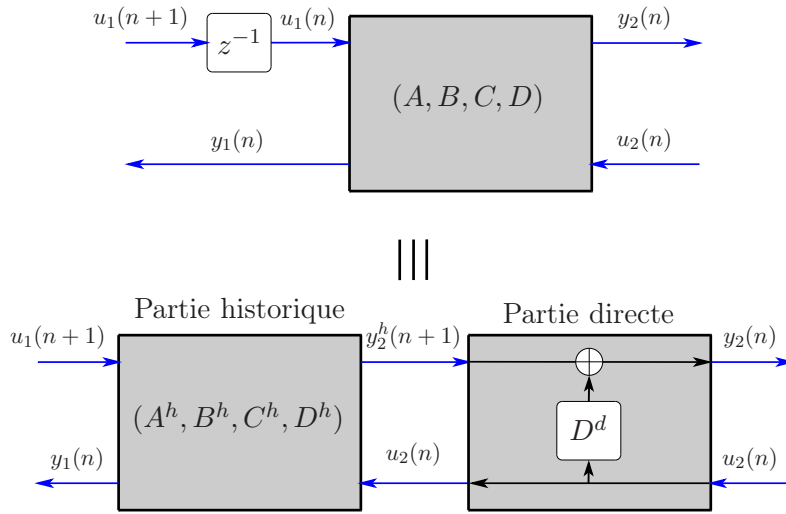


FIGURE 12 – Débouclage

Système direct sans dynamique entièrement décrit par le réel D_{22} :

$$\begin{bmatrix} u_2(n) \\ y_2(n) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & D_{22} \end{bmatrix} \begin{bmatrix} y_2^h(n+1) \\ u_2(n) \end{bmatrix} \quad (42)$$

Pour chaque élément variable, toutes les cellules adjacentes à l'élément sont séparées en parties historiques et directes puis les parties directes et la (ou les) cellule(s) de l'élément variable sont fusionnées entre elles afin de supprimer toutes les liaisons non-simulables. A chaque changement de paramètre d'un élément variable, la (ou les) cellule(s) variable(s) constituant ce dernier est (sont) recalculée(s) puis toutes les cellules constituant l'élément et les cellules directes qui l'entourent sont fusionnées, algébriquement quand c'est possible ou numériquement à l'aide de l'algèbre de *Tassart* si les calculs formels sont trop lourds.

3.3 Fonctionnement des différentes cellules variables

3.3.1 Switch continu

Les trois cellules statiques adjacentes au switch continu sont séparées en parties historiques et directes. La concaténation du switch avec les parties directes qui l'entourent selon le schéma de la figure 13 est effectué algébriquement afin d'établir les formules explicites de la représentation d'état en temps discret de la cellule variable concaténée qui sera simulée. La cellule résultante ne possède pas de dynamique et le switch continu après concaténation est entièrement décrit par sa matrice D'_{SC} . La matrice D'_{SC} s'écrit,

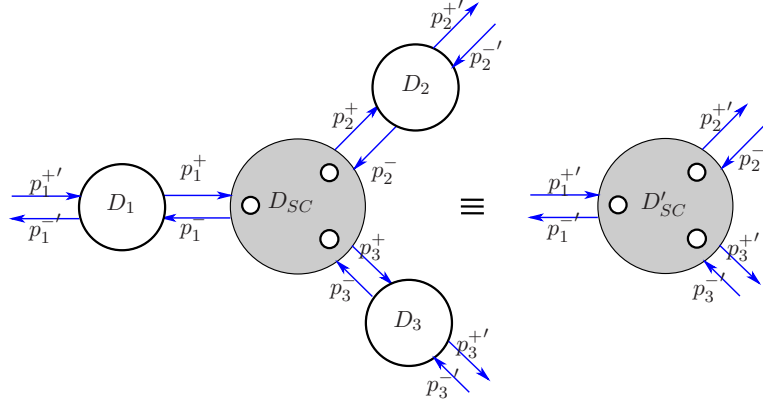


FIGURE 13 – Concaténation algébrique du switch continu

en fonction du paramètre g du switch :

$$D'_{CS} = \frac{1}{\Delta} \begin{bmatrix} D_2 g^2 + D_3 (1 - g)^2 & g & 1 - g \\ g & D_1 g^2 & D_1 (1 - g) \\ 1 - g & D_2 (1 - g) & D_1 (1 - g)^2 \end{bmatrix} \quad (43)$$

où $\Delta = 1 - D_1(D_2 g^2 + D_3 (1 - g)^2)$ et $\{D_1, D_2, D_3\}$ sont les coefficients des parties directes.

C'est cette matrice qui est implémentée dans le programme de simulation temps-réel. Elle est recalculée à chaque fois que l'utilisateur change le paramètre g du switch continu.

3.3.2 Piston

Après le débouclage partiel, le piston est entouré de quatre cellules directes. Les deux cellules directes reliées aux jonctions à trois portes sont concaténées avec ces dernières par le constructeur de structures simulables et les deux autres cellules directes sont passées au programme de simulation avec les cinq cellules constituant le piston. La concaténation de sept cellules pour le piston est beaucoup trop lourde pour être faite analytiquement, même avec des outils de calcul formel type Maple. La méthode de concaténation est donc numérique.

Lorsque l'utilisateur modifie le paramètre k du piston, les matrices des trois diaphragmes sont recalculées puis les cellules sont concaténées numériquement, selon un ordre bien déterminé, grâce aux formules énoncées plus haut. Ainsi, nous retrouvons une unique cellule représentant le piston et le débouclage est assuré.

4 Implémentation Matlab

Le programme de construction automatique de structures simulables a été implémenté en grande partie par *Pierre-Damien Dekoninck* (voir [PDD08]) lors de son stage avec *Rémi Mignot* et *Thomas Hélie* à partir des travaux de thèse de *Rémi Mignot*. J'ai,

pour ma part, continué son travail en résolvant les quelques bugs qui subsistaient dans le programme à la fin de son stage et surtout en élaborant et implémentant les différents éléments variables avec *Rémi Mignot*. J'ai aussi formaté le fichier de structure simulable qui sert d'interface entre le constructeur Matlab et le logiciel C++ de simulation temps-réel.

La procédure du logiciel Matlab de construction de structures simulables suit l'organisation de la partie I du présent document. Il prend en entrée un script qui décrit la géométrie de l'instrument en listant les différents éléments statiques et variables et construit la structure simulable dans l'ordre suivant :

- représentation en cellules MIMO,
- représentations d'état en temps continu,
- concaténation des cellules statiques,
- vérification de controlabilité, d'observabilité et diagonalisation,
- débouclage partiel autour des cellules variables,
- concaténation des cellules directes et variables puis diagonalisation.

Le lecteur intéressé pourra trouver plus d'informations sur le fonctionnement précis du logiciel dans la documentation qui l'accompagne.

5 Simulation temps-réel

Une fois la structure simulable construite grâce au logiciel Matlab que nous venons de décrire, elle est transmise, par l'intermédiaire d'un fichier contenant toutes les informations décrivant la structure simulable et mis au format que j'ai établi, au programme de simulation écrit en C++. Il reste un élément indispensable à la simulation dont nous n'avons pas encore parlé : l'excitateur. Dans un premier temps, j'ai fait des tests avec des signaux d'excitation basiques (impulsion de Dirac pour récupérer la réponse impulsionnelle, bruit blanc pour obtenir un aperçu sonore du comportement du tube modélisé) mais une simulation réaliste nécessite de se pencher sur un modèle de lèvres.

Je vais présenter ici le modèle de lèvre que nous avons utilisé, issu des travaux de *Christophe Vergez*. Pour plus de détail sur le modèle, le lecteur pourra se reporter à [CV98].

5.1 Excitation : modèle de lèvres

Le modèle est présenté en figure 14. Les expériences menées par *Christophe Vergez* et *Xavier Rodet*, reportées dans [CV98], ont montré que la vibration de la lèvre inférieure est négligeable donc seule la lèvre supérieure est simulée.

Le modèle comprend une masse parallélépipédique m reliée à un ressort de raideur k et un amortisseur de coefficient d'amortissement r . Nous négligeons la vitesse de l'air dans la bouche. Nous supposons l'existence d'un flux d'air entre les lèvres et le contact avec l'embouchure est modélisé par la fonction de réflexion acoustique $R^l(t)$. Le couplage non-linéaire entre les lèvres et l'instrument est alors modélisé par les équations de *Bernoulli*

qui relie le flux d'air $u(t)$ entre les lèvres, la position de la lèvre $x(t)$, la pression dans la bouche $p_m(t)$ et la pression à l'entrée de l'embouchure $p(t)$.

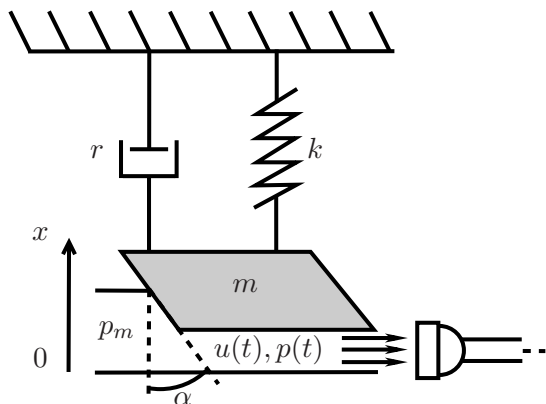


FIGURE 14 – Modèle de lèvre

Sous les hypothèses de propagation en ondes planes, nous pouvons écrire $p(t)$ comme la somme des ondes $p^+(t)$ et $p^-(t)$. Nous appelons $\tilde{x}(t)$ le rapport entre la position des lèvres $x(t)$ et le diamètre de l'entrée de l'embouchure, et $\xi = \text{sgn}(p_m - p^+ - p^-)$. On montre aisément que $\xi = \text{sgn}(p^+ - p^-) = \text{sgn}(p^m - 2p^-)$. Le modèle obtenu est le suivant :

Quand les lèvres sont ouvertes ($\tilde{x}(t) \geq 0$) :

$$\begin{cases} \ddot{\tilde{x}} + v\dot{\tilde{x}} + \omega^2\tilde{x} = Ap_m + N(p^+ + p^-) \\ p^+ = p^- - \xi\tilde{x}\frac{C}{2} \left[C\tilde{x} - \sqrt{(C\tilde{x})^2 + 4|p_m - 2p^-|} \right] \\ p^- = R^l * p^+ \end{cases} \quad (44)$$

Quand les lèvres sont fermées ($\tilde{x}(t) < 0$) :

$$\begin{cases} \ddot{\tilde{x}} + 5v\dot{\tilde{x}} + 4\omega^2\tilde{x} = A(p_m - p^+ - p^-) \\ p^+ = p^- \\ p^- = R^l * p^+ \end{cases} \quad (45)$$

où (voir [CV98])

- $v = \frac{r}{m}$ est la viscosité massique des lèvres,
- $\omega = \sqrt{\frac{k}{m}}$ est proportionnel au facteur de contraction d'une lèvre,
- A, B and C sont des constantes dépendant de la géométrie des lèvres, de la masse m et des constantes physiques (densité de l'air et vitesse du son),
- $*$ est l'opérateur de convolution,
- R^l est la fonction de réflexion à l'entrée de l'instrument.

Les variables d'entrée du modèle sont donc la pression dans la bouche p_m , la pulsation ω , le facteur v et la fonction d'onde retournée par l'embouchure.

5.2 Principe de la simulation

Après avoir construit une structure simulable grâce au programme Matlab et implémenté le modèle de lèvres, nous pouvons les raccorder afin de récupérer $p^-(t)$ à l'entrée de l'embouchure nécessaire au couplage du modèle de lèvres avec le modèle de l'instrument.

La simulation se déroule ensuite échantillon par échantillon. A chaque tour de boucle, nous calculons la valeur de l'onde en entrée de l'embouchure à partir de données utilisateurs (p_m , ω et v) et de l'échantillon de p^- précédent. La simulation discrétisée nous impose en effet d'introduire un retard d'un échantillon entre les lèvres et l'embouchure (il s'agit encore ici du problème de boucle instantannée). Cela revient à introduire un tube sans pertes de quelques centimètres à la sortie des lèvres, ce qui reste une approximation acceptable et qui donne des résultats sonores satisfaisants.

Une fois l'échantillon d'excitation récupéré, les cellules de la structure simulable sont exécutées dans l'ordre suivant :

- propagations,
- cellules statiques,
- cellules variables.

Ce processus est valable car notre programme de construction de structures simulables nous assure que deux cellules de la même famille (propagation, statique ou variable) ne peuvent se trouver côte à côte. En effet, la concaténation des cellules statiques regroupe en une seule cellule statique toutes celles qui se trouvent entre deux cellules variables, une cellule variable et une propagation ou deux propagations. Ainsi le débouclage est réalisé parfaitement.

5.3 Implémentation C++

Le programme décrit ci-dessus a été implémenté en C++ par *Pierre-Damien Dekoninck* mais ne donnait pas des résultats satisfaisant en termes de temps de calcul. J'ai donc dans un premier temps repris ce code afin d'optimiser le temps de calcul. J'ai pour cela modifié les différents appels de fonction en remplaçant tous les objets dans ces appels par des pointeurs et supprimé les nombreuses répétitions inutiles de tâches lourdes en temps de calcul. J'ai par la suite implémenté les différents éléments variables et les fonctions de contrôle de ces éléments.

Le lecteur intéressé par le fonctionnement détaillé du logiciel pourra se reporter à la documentation fournie avec le logiciel.

Deuxième partie

Optimisation de profil de tube

Un objectif de mon stage, un peu en marge de la simulation temps-réel, est le développement d'algorithmes d'optimisation de géométrie d'instrument. Le principe de cette optimisation est de modifier les paramètres géométriques d'un modèle (une concaténation de tronçons à courbure constante) afin de se rapprocher d'un instrument cible. Nous étudions ici des cibles de deux natures différentes : un profil géométrique (relevé de profil d'instrument existant par exemple), il s'agira alors d'optimisation géométrique ; ou bien une impédance que nous souhaitons reproduire avec le modèle, il s'agira alors d'optimisation en impédance.

Contrairement à la simulation temps-réel, ce travail ne s'appuie sur aucun travail précédent et a été entièrement réalisé par moi-même avec l'aide de *Thomas Hélie*.

6 Présentation du problème

6.1 Notations

Nous utiliserons les notations suivantes :

- La variable spatiale dimensionnée (abscisse curviligne) est notée ℓ et les fonctions de cette variable sont notées avec un $\tilde{\cdot}$ (exemple : $\tilde{r}(\ell)$).
- La variable spatiale adimensionnée est notée x et les fonctions de cette variable sont notées sans $\tilde{\cdot}$ (exemple : $r(x)$).
- Les fonctions décrivant la cible du problème d'optimisation sont notées avec un $\underline{\cdot}$ (exemple : $\underline{\tilde{r}}(\ell)$, $\underline{r}(x)$, $\underline{\mathcal{J}}(\omega)$).
- Un indice \cdot_n ou un exposant \cdot^n fait référence au numéro du tronçon.
- Un indice \cdot_k ou un exposant \cdot^k fait référence au numéro du point du relevé.

6.2 Modèle

6.2.1 Tronçon à courbure constante

Nous appelons tronçon caténoïdal une portion de tube à symétrie axiale, à section variable de rayon $\ell \mapsto r(\ell)$ avec une courbure $\Upsilon \in \mathbb{R}$ constante, c'est-à-dire telle que

$$\frac{r''(\ell)}{r(\ell)} = \Upsilon. \quad (46)$$

Selon le signe de Υ , on obtient alors trois catégories de tronçons :

$$\begin{aligned} \Upsilon < 0 & : r(\ell) = A \cos(\sqrt{-\Upsilon} \ell) + B \sin(\sqrt{-\Upsilon} \ell) \text{ avec } (A, B) \in \mathbb{R}^2 \\ \Upsilon = 0 & : r(\ell) = A + B\ell \text{ avec } (A, B) \in \mathbb{R}^2 \\ \Upsilon > 0 & : r(\ell) = A \cosh(\sqrt{\Upsilon} \ell) + B \sinh(\sqrt{\Upsilon} \ell) \text{ avec } (A, B) \in \mathbb{R}^2 \end{aligned}$$

Par ailleurs, pour des raisons géométriques évidentes, les profils admissibles sont tels que $r(\ell) > 0$ et $r'(\ell) \leq 1 \forall \ell \in [0, L]$.

6.2.2 Profil \mathcal{C}^1

De nombreuses optimisations géométriques se font avec des modèles à base de splines. Or nous ne savons résoudre analytiquement la propagation acoustique que dans des tubes axisymétriques à courbure constante. Par conséquent, le modèle $\ell \mapsto r(\ell)$ visé pour approximer le tube sera une concaténation de N tronçons caténoïdaux. Chaque tronçon étant caractérisé par les paramètres $\{\Upsilon, A, B, L\}$ (où L est la longueur du tronçon), les paramètres du modèle complet sont :

- N : nombre de tronçons,
- $\{A_n, B_n, \Upsilon_n, L_n\}_{1 \leq n \leq N}$: paramètres des tronçons.

Par ailleurs, nous souhaitons obtenir une concaténation à régularité \mathcal{C}^1 , nous poserons donc des contraintes d'égalité sur $r(\ell)$ et $r'(\ell)$ aux points de raccordement des tronçons (comme dans le cas des splines par exemple).

7 Optimisation géométrique

Le problème que nous nous posons dans un premier temps est l'approximation d'un profil de tube à l'aide d'un profil modèle. Nous souhaitons optimiser notre modèle constitué d'une concaténation de tronçons élémentaires caténoïdaux afin d'être le plus proche possible d'une cible constituée par le relevé de la perce d'un instrument à symétrie axiale.

Nous disposons d'un relevé de points d'un profil de tube $(\ell_k, \tilde{r}_k)_{k \in \mathbb{K}}$ avec $\mathbb{K} = \llbracket 1, K \rrbracket$, où ℓ_k sont les abscisses curvilignes ($\ell_1 = 0$ et $\ell_K = L$ longueur du tube) et \tilde{r}_k les rayons correspondants. Nous représentons le profil en continu par l'interpolation linéaire du relevé de points :

$$\tilde{r}(\ell) = \sum_{k=1}^K (\tilde{\alpha}_k (\ell - \ell_k) + \tilde{r}_k) \cdot \mathbb{1}_{[\ell_k, \ell_{k+1}[}(\ell) \quad \forall \ell \in [0, L], \quad (47)$$

où $\tilde{\alpha}_k = \frac{\tilde{r}_{k+1} - \tilde{r}_k}{\ell_{k+1} - \ell_k}$.

7.1 Tronçon caténoïdal sur $x \in [-1, 1]$

Soit \mathcal{H} l'espace de Hilbert $\mathcal{L}^2([-1, 1])$ muni de son produit scalaire $\langle f, g \rangle = \int_{-1}^1 f(x)g(x)dx$ et de la norme associée.

Théorème 7.1 (Définition de C et S) *Soit $\Upsilon \in \mathbb{R}$. Une base orthogonale de l'espace des solutions de (46) sur \mathcal{H} est donnée par :*

$$\begin{cases} C & : (x, \Upsilon) \mapsto \phi_1(\Upsilon x^2) \\ S & : (x, \Upsilon) \mapsto x\phi_2(\Upsilon x^2) \end{cases}, \quad (48)$$

où

$$\phi_1 : x \mapsto \sum_{k \geq 0} \frac{x^k}{(2k)!} \quad \text{et} \quad \phi_2 : x \mapsto \sum_{k \geq 0} \frac{x^k}{(2k+1)!} \quad (49)$$

sont analytiques sur \mathbb{C} donc $\mathcal{C}^\infty(\mathbb{R})$ et développables en série entière avec un rayon de convergence $\rho = \infty$.

Remarque : Ces fonctions peuvent s'écrire (indépendamment du choix de $\sqrt{\cdot}$)

$$\phi_1(x) = \cosh(\sqrt{x}), \quad (50)$$

$$\phi_2(x) = \begin{cases} \frac{\sinh(\sqrt{x})}{\sqrt{x}} & \text{si } x \neq 0 \\ 1 & \text{si } x = 0 \end{cases}. \quad (51)$$

J'utiliserai par la suite indifféremment les notations $\{C_\Upsilon(x), S_\Upsilon(x)\}$ et $\{C(x, \Upsilon), S(x, \Upsilon)\}$.

On note $\mathbf{E}_\Upsilon(x) = [C(x, \Upsilon), S(x, \Upsilon)]^T$ et $\tilde{\mathbf{E}}_{\tilde{\Upsilon}}(\ell) = [C(\ell, \tilde{\Upsilon}), S(\ell, \tilde{\Upsilon})]^T$.

Propriété 7.1 (Propriétés basiques de C et S) *Les fonctions C et S sont analytiques et donc développables en séries entières en x avec un rayon de convergence $\rho(\Upsilon) = \infty$ et en Υ avec un rayon de convergence $\rho(x) = \infty$.*

C et S sont respectivement paire et impaire , (52)

$$\begin{aligned} C(x, \Upsilon) &= \sum_{k \geq 0} \frac{\Upsilon^k x^{2k}}{(2k)!} , \\ S(x, \Upsilon) &= \sum_{k \geq 0} \frac{\Upsilon^k x^{2k+1}}{(2k+1)!} , \end{aligned} \quad (53)$$

$$\begin{aligned} \partial_x C(x, \Upsilon) &= C_\Upsilon'(x) = \Upsilon x \phi_2(\Upsilon x^2) = \Upsilon S(x, \Upsilon) , \\ \partial_x S(x, \Upsilon) &= S_\Upsilon'(x) = \phi_1(\Upsilon x^2) = C(x, \Upsilon) , \end{aligned} \quad (54)$$

$$\begin{aligned} \partial_\Upsilon C(x, \Upsilon) &= \frac{x^2}{2} \phi_2(\Upsilon x^2) = \frac{x}{2} S(x, \Upsilon) , \\ \partial_\Upsilon S(x, \Upsilon) &= \frac{x}{2\Upsilon} (\phi_1(\Upsilon x^2) - \phi_2(\Upsilon x^2)) \\ &= \frac{1}{2\Upsilon} (xC(x, \Upsilon) - S(x, \Upsilon)) . \end{aligned} \quad (55)$$

Un tronçon caténoïdal est défini sur $[-1, 1]$ par son rayon :

$$r(x) = A.C(x, \Upsilon) + B.S(x, \Upsilon) . \quad (56)$$

7.2 Changements de variables

Comme je l'ai dit en introduction, le modèle que nous souhaitons optimiser est une concaténation de tronçons caténoïdaux, nous allons donc découper notre profil cible en N tronçons. Dans le but de simplifier les calculs et de rentrer dans le cadre des théorèmes et propriétés précédents sur les tronçons du profil modèle, nous effectuons un changement de variable sur chaque tronçon afin d'adimensionner la variable spatiale et de ramener les abscisses curvilignes dans l'intervalle $[-1, 1]$:

$$x = \mathcal{X}_n(\ell) = \frac{2\ell - \ell_{K_n}^n - \ell_1^n}{L_n} , \quad (57)$$

$$\ell = \mathcal{L}_n(x) = \mathcal{X}_n^{-1}(x) = \frac{xL_n + \ell_{K_n}^n + \ell_1^n}{2} . \quad (58)$$

Le changement de variable revient à écrire sur chaque tronçon, pour le profil cible et pour le profil modèle :

$$\begin{aligned} r_n(x) &= \tilde{r}_n(\mathcal{L}_n(x)) , \\ \underline{r}_n(x) &= \tilde{\underline{r}}_n(\mathcal{L}_n(x)) , \\ r_n(\mathcal{X}_n(\ell)) &= \tilde{r}_n(\ell) , \\ \underline{r}_n(\mathcal{X}_n(\ell)) &= \tilde{\underline{r}}_n(\ell) . \end{aligned}$$

Les formules de passage entre les variables dimensionnées et adimensionnées sont déduites de ces égalités et sont données dans un tableau en annexe résumant toutes les formules de changement de variable.

7.2.1 Profil cible

Le relevé de points décrit en (47) est partitionné en N tronçons de longueurs $\{L_1, \dots, L_N\}$:

$$\tilde{r}(\ell) = \sum_{n=1}^N \tilde{r}_n(\ell) \cdot \mathbb{1}_{[\lambda_n, \lambda_{n+1}]}(\ell), \quad (59)$$

$$\text{avec} \quad \tilde{r}_n(\ell) = \sum_{k=1}^{K_n-1} (\tilde{\alpha}_k^n(\ell - \ell_k^n) + \tilde{r}_k^n) \cdot \mathbb{1}_{[\ell_k^n, \ell_{k+1}^n]}(\ell)$$

$$\text{et} \quad \lambda_n = \ell_1^n = \ell_{K_{n-1}}^{n-1}, \quad \tilde{\alpha}_k^n = \frac{\tilde{r}_{k+1}^n - \tilde{r}_{k+1}^n}{\ell_{k+1}^n - \ell_k^n}.$$

Sur chaque tronçon, nous effectuons un changement de variable de $\ell \in [\ell_1^n, \ell_{K_n}^n]$ vers $x \in [-1, 1]$:

$$x = \mathcal{X}_n(\ell) = \frac{2\ell - \ell_{K_n}^n - \ell_1^n}{L_n},$$

$$\ell = \mathcal{L}_n(x) = \mathcal{X}_n^{-1}(x) = \frac{xL_n + \ell_{K_n}^n + \ell_1^n}{2}.$$

Remarque : $\ell = \tilde{x}$.

En procédant au changement de variable $\ell \rightarrow x$, on obtient

$$\underline{r}_n(x) = \sum_{k=1}^{K_n-1} (\underline{\alpha}_k^n(x - x_k^n) + \underline{r}_k^n) \cdot \mathbb{1}_{[x_k^n, x_{k+1}^n]}(x), \quad x \in [-1, 1], \quad (60)$$

où $\underline{\alpha}_k^n$, x_k^n et \underline{r}_k^n sont déduits des formules de passage données en annexe.

7.2.2 Profil modèle

On note $\tilde{\mathbf{U}}_n = [\tilde{A}_n, \tilde{B}_n]^T$ et $\mathbf{U}_n = [A_n, B_n]^T$.

Le profil modèle pour chaque tronçon s'écrit, avec les paramètres $\{\tilde{\mathbf{U}}_n, \tilde{\Upsilon}_n, L_n\}$:

$$\tilde{r}(\ell) = \sum_{n=1}^N \tilde{r}_n(\ell) \cdot \mathbb{1}_{[\ell_1^n, \ell_{K_n}^n]}(\ell) \quad \ell \in [0, L], \quad (61)$$

$$\tilde{r}_n(\ell) = \tilde{\mathbf{E}}_{\tilde{\Upsilon}_n}^T(\ell) \tilde{\mathbf{U}}_n \quad \ell \in [\ell_1^n, \ell_{K_n}^n]. \quad (62)$$

En procédant au changement de variable $\ell \rightarrow x$, on obtient¹

$$r_n(x) = \mathbf{E}_{\Upsilon_n}^T(x) \mathbf{U}_n \quad x \in [-1, 1], \quad (63)$$

1. où \mathbf{U}_n et Υ_n sont déduits des formules de passage données en annexe.

7.3 Résolution analytique des contraintes \mathcal{C}^1

Les contraintes de régularité \mathcal{C}^1 entre les tronçons s'écrivent

$$\forall n \in \llbracket 1, N-1 \rrbracket \begin{cases} \tilde{r}_n(\ell_{K_n}^n) = \tilde{r}_{n+1}(\ell_1^n) \\ \partial_\ell \tilde{r}_n(\ell_{K_n}^n) = \partial_\ell \tilde{r}_{n+1}(\ell_1^n) \end{cases} .$$

Après le changement de variable $\ell \rightarrow x$, la résolution de ce système conduit à l'équation de récurrence (cf annexe)

$$\mathbf{U}_{n+1} = \mathbf{\Phi}_{n+1} \mathbf{G}_n \mathbf{\Phi}_n \mathbf{U}_n, \quad (64)$$

dont on déduit que (cf annexe)

$$\mathbf{U}_n = \mathbf{Q}_n \mathbf{U}_1 = \mathbf{\Phi}_n \prod_{p=1}^{n-2} \left[\mathbf{G}_{n-p} \mathbf{\Psi}_{n-p} \right] \mathbf{G}_1 \mathbf{\Phi}_1 \mathbf{U}_1, \quad (65)$$

où

$$\mathbf{\Phi}_n = \begin{pmatrix} \phi_1(\Upsilon_n) & \phi_2(\Upsilon_n) \\ \Upsilon_n \phi_2(\Upsilon_n) & \phi_1(\Upsilon_n) \end{pmatrix}, \quad \mathbf{G}_n = \begin{pmatrix} 1 & 0 \\ 0 & \frac{L_{n+1}}{L_n} \end{pmatrix},$$

$$\mathbf{\Psi}_n = (\mathbf{\Phi}_n)^2 = \begin{pmatrix} \phi_1(4\Upsilon_n) & 2\phi_2(4\Upsilon_n) \\ 2\Upsilon_n \phi_2(4\Upsilon_n) & \phi_1(4\Upsilon_n) \end{pmatrix}.$$

7.4 Présentation du problème d'optimisation

Comme je l'ai dit dans la partie précédente, chacun des N tronçons du profil modèle est paramétré par quatre paramètres $\{\Upsilon_n, L_n, A_n, B_n\}$. Le problème présente donc $4N$ degrés de liberté qui peuvent être regroupés en $2N$ paramètres linéaires : les couples $\{A_n, B_n\}$, et $2N$ paramètres non linéaires : les couples $\{\Upsilon_n, L_n\}$.

La résolution au paragraphe précédent des $2(N-1)$ contraintes d'égalité de régularité \mathcal{C}^1 et l'expression par récurrence (64) obtenue permet de résoudre $2(N-1)$ degrés de liberté sur les paramètres linéaires. Il est donc possible d'exprimer linéairement tous les couples $\{A_n, B_n\}$ en fonction d'un unique couple de paramètres (deux degrés de liberté restants). Ce couple peut être simplement $\{A_1, B_1\}$ (comme présenté dans l'équation 65) auquel cas notre travail d'optimisation portera non seulement sur les paramètres $\{\Upsilon_n, L_n\}$ mais aussi sur le couple $\{A_1, B_1\}$ ².

2. Il sera aussi possible de résoudre simplement les deux degrés de liberté associés à $\{A_1, B_1\}$ à l'aide de deux contraintes supplémentaires (fixer $\tilde{r}(\ell)$ et/ou $\tilde{r}'(\ell)$ en un ou deux point(s) du profil par exemple). Dans le logiciel développé, le couple $\{A_1, B_1\}$ est systématiquement calculé à partir du couple $\{\tilde{r}(0), \tilde{r}'(0)\}$. Enfin, il sera possible de ne poser qu'une seule contrainte et donc d'exprimer les couples $\{A_n, B_n\}$ en fonction cette fois d'un seul paramètre qui sera intégré dans le travail d'optimisation sur les paramètres non linéaires.

Le critère que nous cherchons à minimiser lors de l'optimisation est la distance quadratique entre $\tilde{r}(\ell)$ et $\tilde{r}(\ell)$ qui s'exprime de la façon suivante :

$$\mathcal{E}(\theta) = \int_0^L (\tilde{r}(\ell) - \tilde{r}(\ell))^2 d\ell. \quad (66)$$

Nous utiliserons une méthode numérique pour la minimisation de ce critère.

7.5 Critère

7.5.1 Définition du critère

Le critère s'écrit en fonction du vecteur de paramètres libres θ , concaténation de Υ vecteur des $\{\Upsilon_n\}$, de \mathbf{L} vecteur des $\{L_n\}$ et éventuellement du vecteur \mathbf{U}_1 .

$$\begin{aligned} \mathcal{E}(\theta) &= \int_0^L (\tilde{r}(\ell) - \tilde{r}(\ell))^2 d\ell \\ &= \int_0^L \tilde{r}^2(\ell) d\ell + \int_0^L \tilde{r}^2(\ell) d\ell - 2 \int_0^L \tilde{r}(\ell)\tilde{r}(\ell) d\ell. \end{aligned}$$

Le terme $\int_0^L \tilde{r}^2(\ell) d\ell$ ne nous intéresse pas car il ne dépend pas du vecteur θ . Nous travaillerons donc sur le critère réduit (par soucis de clarté, comme nous ne travaillerons par la suite plus que sur le critère réduit, nous conservons la notation du critère pour le critère réduit dans la suite du document) :

$$\mathcal{E}(\theta) = \int_0^L \tilde{r}^2(\ell) d\ell - 2 \int_0^L \tilde{r}(\ell)\tilde{r}(\ell) d\ell. \quad (67)$$

On obtient en variable adimensionnée et en écriture matricielle :

$$\mathcal{E}(\theta) = \mathbf{U}_1^T \mathbf{Q}(\Upsilon, \mathbf{L}) \mathbf{U}_1 - 2 \mathbf{V}(\Upsilon, \mathbf{L})^T \mathbf{U}_1 \quad (68)$$

où

$$\mathbf{Q}(\Upsilon, \mathbf{L}) = \sum_{n=1}^N \mathbf{Q}_n^T \mathbf{M}_n \mathbf{Q}_n, \quad (69)$$

$$\mathbf{V}(\Upsilon, \mathbf{L}) = \sum_{n=1}^N \mathbf{V}_n, \quad (70)$$

et

$$\mathbf{M}_n = \begin{pmatrix} \phi_2(4\Upsilon_n) + 1 & 0 \\ 0 & \frac{\phi_2(4\Upsilon_n) - 1}{\Upsilon_n} \end{pmatrix}, \quad (71)$$

$$\mathbf{V}_n = \mathbf{Q}_n^T \mathbf{W}_n, \quad (72)$$

$$\mathbf{W}_n^T = (\mathcal{C}_A^n \quad \mathcal{C}_B^n), \quad (73)$$

avec

$$\begin{aligned} \mathcal{C}_A^n = & \sum_{k=1}^{K_n} \left[\alpha_k^n \left((x_{k+1}^n - x_k^n) x_{k+1}^n \phi_2(\Upsilon_n x_{k+1}^{n2}) \right. \right. \\ & \left. \left. - \frac{\phi_1(\Upsilon_n x_{k+1}^{n2}) - \phi_1(\Upsilon_n x_k^{n2})}{\Upsilon_n} \right) \right. \\ & \left. + \underline{x}_k^n \left(x_{k+1}^n \phi_2(\Upsilon_n x_{k+1}^{n2}) - x_k^n \phi_2(\Upsilon_n x_k^{n2}) \right) \right], \end{aligned} \quad (74)$$

$$\begin{aligned} \mathcal{C}_B^n = & \frac{1}{\Upsilon_n} \sum_{k=1}^{K_n} \left[\alpha_k^n \left((x_{k+1}^n - x_k^n) \phi_1(\Upsilon_n x_{k+1}^{n2}) \right. \right. \\ & \left. \left. - x_{k+1}^n \phi_2(\Upsilon_n x_{k+1}^{n2}) + x_k^n \phi_2(\Upsilon_n x_k^{n2}) \right) \right. \\ & \left. + \underline{x}_k^n \left(\phi_1(\Upsilon_n x_{k+1}^{n2}) - \phi_1(\Upsilon_n x_k^{n2}) \right) \right]. \end{aligned} \quad (75)$$

7.5.2 Etude du critère lorsque $\Upsilon_n \rightarrow 0$

L'expression de \mathbf{U}_n en fonction \mathbf{U}_1 (eq. 65) nous montre que A_n et B_n sont parfaitement définis pour toutes les valeurs de Υ_n , et ce pour tout n . En revanche, certains termes des expressions (74), (75) et (71) ci-dessus ne sont pas définis pour $\Upsilon_n = 0$. Nous allons exprimer les développements en série entière de ces termes.

Théorème 7.2 (DSE partiel du critère)

La fonction $f_1 : \Upsilon \mapsto \frac{\phi_1(\Upsilon a^2) - \phi_1(\Upsilon b^2)}{\Upsilon}$ est développable en série entière en $\Upsilon = 0$ avec un rayon de convergence $\rho = \infty$.

$$f_1(\Upsilon) = \sum_{p \geq 0} \frac{a^{2p+2} - b^{2p+2}}{(2p+2)!} \Upsilon^p \quad (76)$$

La fonction $f_2 : \Upsilon \mapsto \frac{(a-b)\phi_1(\Upsilon a^2) - a\phi_2(\Upsilon a^2) + b\phi_2(\Upsilon b^2)}{\Upsilon}$ est développable en série entière en $\Upsilon = 0$ avec un rayon de convergence $\rho = \infty$.

$$f_2(\Upsilon) = \sum_{p \geq 0} \left(\frac{(a-b)b^{2p+2}}{(2p+2)!} + \frac{b^{2p+3} - a^{2p+3}}{(2p+3)!} \right) \Upsilon^p \quad (77)$$

La fonction $f_3 : \Upsilon \mapsto \frac{\phi_2(4\Upsilon_n) - 1}{\Upsilon_n}$ est développable en série entière en $\Upsilon = 0$ avec un rayon de convergence $\rho = \infty$.

$$f_3(\Upsilon) = \sum_{p \geq 0} \frac{2 \times 4^p}{(p+1)(2p+2)(2p+1)!} \Upsilon^p \quad (78)$$

7.6 Contraintes supplémentaires : critère pénalisé, lagrangien, lagrangien augmenté

Dans le cas où le vecteur \mathbf{U}_1 est défini à l'aide de deux contraintes, le modèle ne présente plus aucun degré de liberté linéaire. Il est donc impossible de résoudre des contraintes supplémentaires analytiquement. J'ai mis en place trois méthodes différentes afin de pouvoir prendre en compte des contraintes supplémentaires : un critère pénalisé, un lagrangien et un lagrangien augmenté. Nous considérerons ici le cas d'une seule contrainte s'exprimant $c(\theta) = 0$ (par exemple $\tilde{r}(\ell_0) - \tilde{r}(\ell_0) = 0$), les méthodes dans le cas de plusieurs contraintes seront déduites immédiatement.

7.6.1 Critère pénalisé

La première solution envisagée est de rajouter une pénalisation au critère sous la forme suivante :

$$\mathcal{E}_p(\theta) = \mathcal{E}(\theta) + \epsilon c(\theta)^2.$$

où ϵ est le paramètre de pénalisation. Afin de rendre ce paramètre plus pertinent, on utilise en pratique un critère pénalisé relatif :

$$\mathcal{E}_p(\theta, \epsilon) = \frac{\mathcal{E}(\theta)}{\|\tilde{\mathbf{z}}(\ell)\|_2} + \epsilon c_r(\theta)^2. \quad (79)$$

où $c_r(\theta)$ est la contrainte relative, de la forme $\frac{\tilde{r}(\ell_0) - \tilde{r}(\ell_0)}{\tilde{r}(\ell_0)}$

7.6.2 Lagrangien

La deuxième solution envisagée est l'utilisation d'un lagrangien :

$$\mathcal{E}_l(\theta, \mu) = \mathcal{E}(\theta) + \mu c(\theta).$$

De même, afin de rendre le paramètre μ plus pertinent, on utilise en pratique un lagrangien relatif :

$$\mathcal{E}_l(\theta, \mu) = \frac{\mathcal{E}(\theta)}{\|\tilde{\mathbf{z}}(\ell)\|_2} + \mu c_r(\theta). \quad (80)$$

7.6.3 Lagrangien augmenté

La troisième solution envisagée est l'utilisation d'un lagrangien augmenté, qui n'est autre qu'un lagrangien pénalisé :

$$\mathcal{E}_{lp}(\theta, \mu, \epsilon) = \mathcal{E}(\theta) + \mu c(\theta) + \epsilon c(\theta)^2.$$

De même, on utilise :

$$\mathcal{E}_{lp}(\theta, \mu, \epsilon) = \frac{\mathcal{E}(\theta)}{\|\tilde{\mathbf{z}}(\ell)\|_2} + \mu c_r(\theta) + \epsilon c_r(\theta)^2. \quad (81)$$

J'ai finalement retenu la solution du critère pénalisé car si les lagrangiens et les lagrangiens augmentés sont plus performants en théorie, leurs implémentations numériques sont très délicates et mes quelques tentatives ne m'ont apporté que des instabilités qui rendaient l'optimisation inefficace.

7.7 Implémentation Matlab de l'optimisation géométrique

7.7.1 Fonctions non définies en 0

Autour de 0, la fonction ϕ_2 ne peut pas être définie numériquement à l'aide de l'éq. (51) à cause de la singularité en 0. De même pour les termes \mathcal{C}_A^n et \mathcal{C}_B^n qui ne sont pas définis pour des courbures nulles. Ces fonctions sont donc programmées avec un disque centré sur 0 dans lequel elles seront calculées par un développement limité et non par leur formule exacte. Les valeurs du rayon du disque et de l'ordre du développement limité ont été déterminées (empiriquement) afin d'assurer une précision et un raccordement entre le DL et la fonction exacte au périmètre du disque satisfaisants.

7.7.2 Point de départ de l'optimisation et choix de N

J'ai choisi comme point du départ pour l'algorithme d'optimisation tel qu'il est implémenté actuellement un profil droit (i.e $\Upsilon_n = 0, \forall n$) divisé en N tronçons de même longueur (i.e $L_n = \frac{L}{N}, \forall n$) avec \mathbf{U}_1 calculé pour que le profil modèle soit tel que $\tilde{r}(0) = \tilde{\underline{r}}(0)$ et $\tilde{r}'(0) = \tilde{\underline{r}}'(0)$.

Le choix du nombre de tronçons est sans doute le problème le plus délicat. Le nombre de tronçons sera choisi au cas par cas, afin de trouver un compromis entre nombre de tronçons, qualité de l'approximation, qualités acoustiques et performances temps-réel

7.7.3 Problème de minima locaux

Le principal obstacle auquel j'ai dû faire face était le nombre de minima locaux du critère. En effet, le critère est une fonction très complexe comportant un nombre très important de minima locaux et je me suis très vite aperçu dans l'utilisation des fonctions de recherche de minimum de Matlab (type `fminsearch`) que le choix du point de départ était critique pour le bon fonctionnement de l'optimisation. J'ai alors mis au point un algorithme d'optimisation qui permet de rester fixé au minimum local qui m'intéresse :

- **1** Optimisation sur la courbure du premier tronçon
- puis, pour chaque tronçon suivant,
- **n** Optimisation sur la courbure du n-ième tronçon
- **n bis** Optimisation sur les courbures des tronçons 1 à n
- **n ter** Optimisation sur les courbures et les longueurs des tronçons 1 à n (facultatif)

Cette série d'optimisation nous assure dans la plupart des cas rencontrés jusqu'ici le suivi du même minimum local tout au long de la procédure d'optimisation. Un bon compromis temps / performance consiste à ne réaliser l'étape n ter qu'une fois arrivé au dernier tronçon.

7.7.4 Résultats

Une première série de résultats, présentée à la figure 15 illustre l'utilité de l'algorithme d'optimisation décrit ci-dessus et de l'optimisation sur les $\{L_n\}$. Les graphiques présentent en rouge la courbe cible, il s'agit ici d'un polynôme de degré 3 non réaliste dans le cadre d'une perce d'instrument mais utilisé ici pour tester la robustesse de l'optimisation, et en bleu le modèle sur lequel on peut voir les points de raccordements entre les tronçons. Les figures de la première ligne présentent une approximation à l'aide d'un modèle à 3 tronçons et celles de la deuxième ligne à l'aide d'un modèle à 4 tronçons. Les figures de la première colonne présentent des optimisations sur les seuls paramètres $\{\Upsilon\}$, les optimisations de la seconde colonne utilisent le compromis cité plus haut et les optimisations de la troisième colonne suivent toutes les étapes d'optimisation.

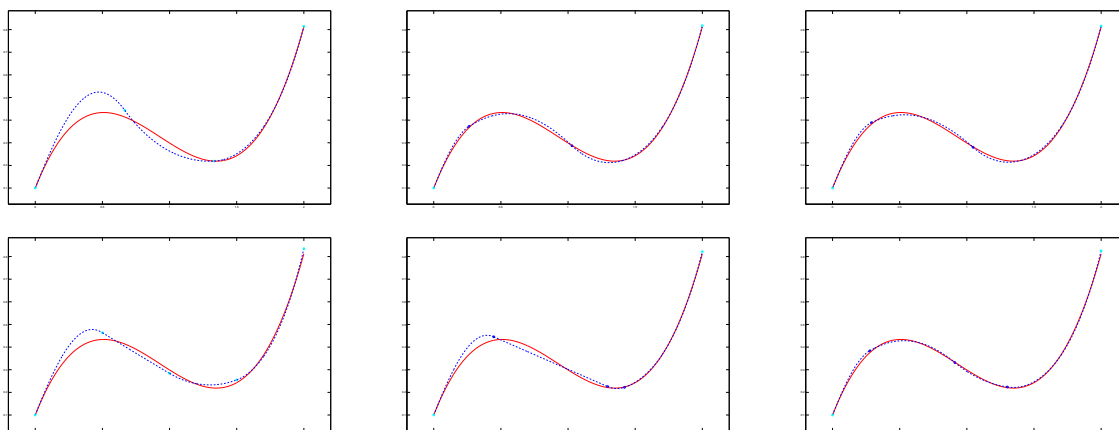


FIGURE 15 – Première série de résultats d'optimisation.

Nous constatons aisément que l'inclusion des longueurs dans les paramètres d'optimisation des tronçons offre de bien meilleurs résultats en général. On constate cependant quelques cas isolés comme celui de l'optimisation sur le modèle à quatre tronçons dans la deuxième colonne où le critère se trouve bloqué dans un minimum local, l'algorithme faisant parfois même tendre la longueur d'un tronçon vers 0.

De même, il arrive des cas où l'augmentation du nombre de tronçons n'améliore pas les performances. Sur un des profils de test, j'ai pu constater que les performances étaient meilleures pour un nombre de tronçons impair et qu'au delà d'un certain nombre de tronçons, l'algorithme d'optimisation réduisait tout seul le nombre de tronçons en faisant tendre certains paramètres L_n vers 0.

Nous pouvons constater tout de même que, dès lors que l'on optimise sur les courbures et sur les longueurs, l'algorithme donne de bons résultats dans l'ensemble, le gros défaut restant encore le temps de calcul, l'obtention d'un des résultats présentés figure 15 demande, suivant le nombre de paramètres à optimiser, entre 5 et 40 secondes.

Une deuxième série de mesures, présentée en figure 16 montre les performances des contraintes numériques ajoutées sur le rayon et la pente à l'extrémité droite du profil. La cible utilisée est cette fois-ci un relevé de perce de trombone effectué à l'IRCAM par *René Caussé* et le modèle possède trois tronçons. Toutes les simulations ont été effectuées avec optimisation sur les longueurs uniquement à la dernière étape.

Le premier graphique résulte d'une optimisation sans contrainte numérique. Les résultats obtenus sont les suivants (e désigne l'erreur relative égale à la norme de la différence divisée par la norme de la cible, e_r désigne l'erreur relative sur le rayon en $\ell = L$ et e_p désigne l'erreur relative sur la pente en $\ell = L$) :

$$\begin{cases} e &= 1.4 \cdot 10^{-2} \\ e_r &= 2.7 \cdot 10^{-2} \\ e_p &= 2.8 \cdot 10^{-1} \end{cases}$$

Le deuxième graphique résulte d'une optimisation avec contrainte numérique sur le rayon en $\ell = L$. Les résultats sont les suivants :

$$\begin{cases} e &= 2.0 \cdot 10^{-2} \\ e_r &= 6.8 \cdot 10^{-5} \\ e_p &= 1.9 \cdot 10^{-1} \end{cases}$$

Le troisième graphique résulte d'une optimisation avec contrainte numérique sur le rayon et la pente en $\ell = L$. Les résultats sont les suivants :

$$\begin{cases} e &= 7.6 \cdot 10^{-2} \\ e_r &= 8.3 \cdot 10^{-4} \\ e_p &= 1.7 \cdot 10^{-3} \end{cases}$$

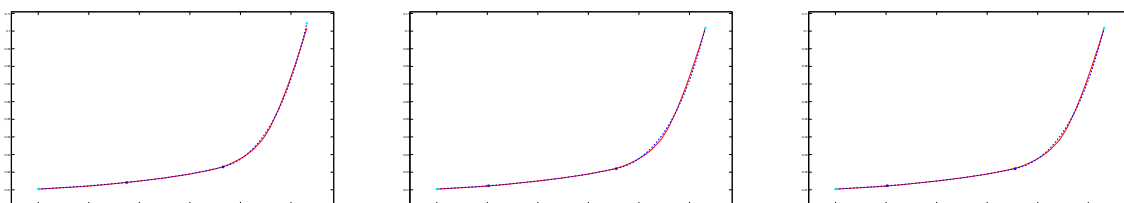


FIGURE 16 – Deuxième série de résultats d'optimisation.

Nous constatons que les contraintes numériques sont assez bien réalisées. Par contre, la dégradation de l'approximation est assez nette et les profils trouvés sont souvent mauvais d'un point de vue acoustique. Une solution est évidemment l'augmentation du nombre de tronçons. Il faudra alors, suivant les cas, trouver le compromis entre nombre de tronçons, qualité de l'approximation, qualités acoustiques et performances temps-réel.

7.8 Applications

L'application principale de l'optimisation géométrique est le clonage d'instrument. Le fait de disposer d'un profil modèle qui approxime le plus finement possible la géométrie d'un instrument donné permet de reproduire très fidèlement cet instrument en virtuel et donc de pouvoir par la suite construire une structure simulable en temps-réel dont le rendu sonore se rapproche tout aussi fidèlement du rendu sonore de l'instrument original.

8 Optimisation en impédance

Le deuxième problème d'optimisation est l'optimisation en impédance. Nous disposons maintenant d'une impédance cible et nous souhaitons obtenir un profil modèle dont l'impédance se rapproche au mieux de la cible. Contrairement à l'optimisation géométrique, ce travail n'en est qu'à l'état de première ébauche et ce que je vais exposer dans la suite est constitué, pour une grande partie, par des idées qui sont encore à développer.

8.1 Calcul de l'impédance à partir du modèle

Le calcul de l'impédance à partir du modèle se fait grâce à la structure simulable que j'ai décrite en partie I. Les paramètres des différents tronçons du modèle, auquel nous pouvons rajouter une embouchure donnée et un modèle de rayonnement, nous permettent de construire une structure en cellules MIMO. Pour l'instant, seules les structures composées uniquement de quadripôles (c'est à dire une simple succession de tronçons dotée d'une embouchure en tête et d'un rayonnement en queue) sont prises en charge par le logiciel.

Pour chaque cellule MIMO, les quatre fonctions de transfert sont évaluées pour différentes valeurs de la variable de Laplace, en général un vecteur décrivant linéairement ou logarithmiquement l'intervalle de fréquence qui nous intéresse. Une fois les fonctions évaluées, les quadripôles sont concaténés numériquement un par un (voir la méthode ci-après). Nous obtenons ainsi finalement un quadripôle contenant les quatre fonctions de transfert de la structure complète dont nous pouvons extraire la réflexion en entrée, la transmission et l'impédance.

La figure 17 présente la concaténation de deux quadripôles. Les formules explicitant le quadripôle résultant de la concaténation sont les suivantes :

$$\begin{cases} H_{11}^e &= H_{11}^1 + \frac{H_{12}^1 H_{11}^2 H_{21}^1}{1 - H_{22}^1 H_{11}^2}, \\ H_{12}^e &= \frac{H_{12}^2 H_{12}^1}{1 - H_{22}^1 H_{11}^2}, \\ H_{21}^e &= \frac{H_{21}^1 H_{21}^2}{1 - H_{22}^1 H_{11}^2}, \\ H_{22}^e &= H_{22}^2 + \frac{H_{12}^2 H_{22}^1 H_{21}^2}{1 - H_{22}^1 H_{11}^2}. \end{cases} \quad (82)$$

Il faut noter que ces formules sont écrites pour un cas bien précis : la deuxième porte du premier quadripôle est reliée à la première porte du deuxième quadripôle. Il est

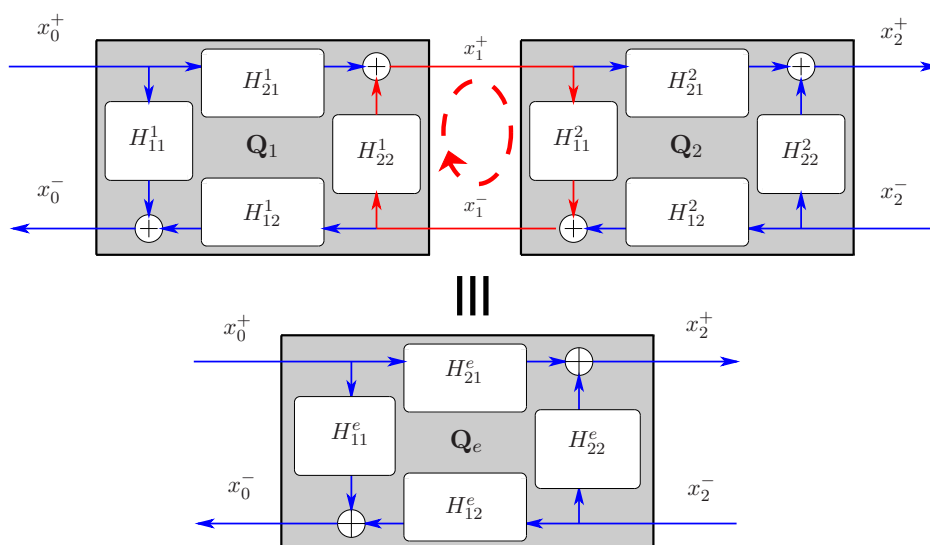


FIGURE 17 – Concaténation de 2 quadripôles

possible qu'en termes de numéros de portes, d'autres configurations se présentent. Les nouvelles formules seront alors aisément déduites des précédentes en inversant pour l'un ou l'autre des quadripôles les expressions H_{11}^n et H_{22}^n d'une part et H_{12}^n et H_{21}^n d'autre part.

8.2 Choix du critère

De fait de la méthode numérique de calcul de l'impédance, seuls des critères numériques sont envisageables. Le critère évident est, comme pour l'optimisation géométrique, la distance quadratique entre les impédances cible et modèle, calculée numériquement cette fois. C'est celui-ci qui est utilisé pour l'instant dans notre ébauche de logiciel. Il s'avère cependant que ce critère n'est pas le plus pertinent. En effet, d'un point de vue musical ou du point de vue de l'instrumentiste, les descripteurs porteurs d'information dans l'impédance d'un instrument sont les positions, en fréquence et en gain, des différents "pics" présents dans la courbe d'impédance, ainsi que leurs facteurs de qualités. Il sera donc intéressant dans le futur d'élaborer un critère prenant en compte les différences de position et/ou de facteur de qualité sur les pics.

9 Développements d'outils de "LAO"

Une poursuite intéressante de ce projet serait le développement d'outils de lutherie assistée par ordinateur "LAO" utilisables directement par des luthiers professionnels. Même si, aujourd'hui, ces logiciels ne sont encore qu'à l'état d'ébauche, ils permettent de calculer automatiquement une géométrie d'instrument basée sur notre modèle dans le but d'obtenir un clone d'un instrument donné. Les fonctions que je souhaite implémenter

dans le futur sont les outils dont ont besoin les luthiers, comme par exemple l'optimisation de l'harmonicité des pics, i.e optimiser les rapports entre les fréquences des différents pics pour obtenir une suite harmonique la plus juste possible, ou encore l'optimisation de la jouabilité de l'instrument, i.e améliorer le facteur de qualité des pics afin de permettre au musicien d'accrocher plus facilement les différentes notes.

Nous le comprenons ici, la recherche en LAO n'est pas possible sans un dialogue concret entre les chercheurs et les luthiers. C'est pourquoi nous essayons aujourd'hui à l'IRCAM de rapprocher ces travaux du projet PAFI (Plateforme modulaire d'Aide à la Facture Instrumentale), réunissant chercheurs, pôle d'innovation des métiers de la musique et artisans-luthiers, financé par l'Agence National de la Recherche et coordonné par l'Itemm (Institut Technologique Européen des Métiers de la Musique). Il s'agit dans ce projet de répondre à la problématique de la reproduction et de l'optimisation de la conception d'instruments de musique haut de gamme, caractéristiques de la lutherie française sur le marché mondial. PAFI propose donc la mise en œuvre d'outils de caractérisation et de prédiction mécanique et acoustique dédiés à l'analyse et au prototypage virtuel des instruments.

Une présentation des présents travaux est prévue à l'automne 2009 lors d'une journée de travail sur les outils logiciels d'aide à la lutherie, aujourd'hui encore en cours d'organisation, mise en place par *Joël Gilbert* et *René Caussé* dans le cadre du projet PAFI.

Troisième partie

Validation du modèle : mesures et simulation

Le dernier objectif du stage est la mise en place de séries de mesures et de simulations destinées à vérifier la pertinence du modèle que j'ai décrit en introduction. Mon objectif était de faire des mesures d'impédance sur plusieurs instruments (une trompette, un trombone et un cor) et sur un prototype de pavillon réalisé à partir d'un profil théorique afin de comparer les impédances mesurées avec les impédances simulées à partir des modèles numériques des instruments et du profil théorique.

Je disposais, pour faire les mesures, d'un banc d'impédance accompagné d'un logiciel de mesures.

10 Problème du sinus glissant

Les sweeps (ou sinus glissants) générés par le logiciel accompagnant le banc d'impédance ne me satisfaisaient pas car j'obtenais des courbes présentant un rapport signal sur bruit trop faible et inégalement réparti sur la bande de fréquence étudiée. J'ai donc décidé de créer mes propres signaux d'excitation. Je pars de l'hypothèse que la réponse du piezzo d'excitation est linéaire dans la bande qui m'intéresse, hypothèse assez bien vérifiée empiriquement.

Le but est de générer un sweep $u(t)$ parfait, c'est-à-dire tel que

$$|\mathcal{F}[u](f)| = \begin{cases} 1 & \text{dans la bande de fréquence souhaitée,} \\ 0 & \text{ailleurs.} \end{cases} \quad (83)$$

avec $\mathcal{F}[u]$ la transformée de Fourier de u .

La solution consiste à fixer le retard de groupe $T_g(f)$ correspondant à un sweep et de calculer la phase $\phi(f) = -2\pi \int_0^f T_g(\theta) d\theta$ correspondante. J'ai étudié deux fonctions possible pour $T_g(f)$:

1. $T_g(f) = \frac{T}{f_2 - f_1} (f - f_1) \mathbb{1}_{[f_1, f_2]}(f)$
2. $T_g(f) = \frac{T}{\ln\left(\frac{f_2}{f_1}\right)} \ln\left(\frac{f}{f_1}\right) \mathbb{1}_{[f_1, f_2]}(f)$

où $[f_1, f_2]$ est l'intervalle de fréquence balayé par le sweep et T est la durée du sweep.

La calcul de la phase donne les résultats suivants :

$$1. \phi(f) = -\frac{\pi T}{f_2 - f_1} (f - f_1)^2 \mathbb{1}_{[f_1, f_2]}(f) \quad (84)$$

$$2. \phi(f) = -\frac{2\pi T}{\ln\left(\frac{f_2}{f_1}\right)} \left[f \left(\ln\left(\frac{f}{f_1}\right) - 1 \right) + f_1 \right] \mathbb{1}_{[f_1, f_2]}(f) \quad (85)$$

La solution du $T_g(f)$ linéaire donne des meilleurs résultats en terme de RSB (notamment dans les hautes fréquences), j'ai donc opté pour celle-ci.

Le sweep est ensuite obtenu en construisant un vecteur des valeurs de $\mathcal{F}[u]$ à partir des équations (83) et (84) ou (85), en veillant à respecter la symétrie hermitienne et la condition de Shannon $f_2 < \frac{F_s}{2}$ (où F_s est la fréquence d'échantillonnage). Le signal est finalement obtenu par FFT inverse sur le vecteur construit.

J'ai de plus pensé à rajouter un temps de relaxation à l'issue du sweep dans le signal d'excitation afin de s'assurer que la mesure récupère la totalité de la réponse libre du système mais j'ai abandonné cette solution qui dégradait de façon significative le spectre du sweep et le RSB de la mesure.



FIGURE 18 – Mesure d'impédance de la trompette

11 Campagnes de mesures effectuées à l'IRCAM

J'ai eu l'occasion d'effectuer plusieurs campagnes de mesure pendant mon stage. Malheureusement, par faute de temps et de moyens, nous n'avons pu construire aucune structure clone des instruments mesurés. Je vais donc présenter rapidement ici les mesures effectuées en espérant qu'elles puissent un jour être exploitées pour vérifier les modèles utilisés tout au long de mes travaux.

Les mesures ont été réalisées à l'IRCAM, en chambre anéchoïque ou bien en cabine insonorisée, à l'aide du banc d'impédance réalisé par le Laboratoire d'Acoustique de l'Université du Maine (LAUM).

11.1 Trompette

Une première série de mesures a été réalisée sur un trompette Yamaha dont dispose l'équipe *Acoustique Instrumentale* de l'IRCAM. Les mesures ont été faites en présence d'une embouchure. Une photo du montage de l'expérience est présentée en figure 18.

J'ai mesuré en chambre anéchoïque l'impédance de la trompette pour chaque doigté, comme présenté sur les photos de la figure 19.

Malheureusement, je ne possède pour l'instant aucun relevé de la perce de la trompette, il m'est donc impossible de construire une modélisation dont je peux calculer l'impédance pour comparer les résultats.



FIGURE 19 – Différents doigtés de la trompette

11.2 Cor

J'ai eu l'opportunité de mesurer un cor *Paxman* flambant neuf. Une photo de l'installation de mesure est présentée en figure 20. Je tiens à remercier encore une fois ici très chaleureusement *Jérôme Joubert* qui m'a permis de faire une campagne de mesures très complète sur son cor et m'a aidé tout au long de ces mesures.

J'ai ici aussi fait les mesures pour les différents doigtés. Par ailleurs, les cornistes ont l'habitude de jouer sur la justesse de leur instrument en bouchant plus ou moins le pavillon avec la main. Nous pouvons voir sur la figure 21 la main à l'entrée du pavillon en position normale de jeu.

Nous voyons sur la figure 22 l'influence de la position de la main à l'entrée du pavillon. On constate nettement un réhaussement des derniers pics visibles en hautes fréquences aux alentours de 2kHz lorsque la main bouche la sortie du pavillon. Une observation plus fine permet de voir que les fréquences des pics sont légèrement décalées vers le grave.

A ce jour, je ne dispose d'aucun relevé de la perce de cet instrument, il m'est donc impossible de réaliser une simulation pour comparer les impédances réelles et simulées.

12 Prototype de pavillon sur-mesure

Dans le but de comparer les impédances mesurée et simulée, *Rémi Mignot* a construit un profil de tube constitué d'un seul tronçon caténoïdale avec (en abscisse curviligne) une pente nulle à l'entrée et une pente de 1 en sortie. La profil a ensuite été passé en abscisse axiale (et présente donc une pente horizontale à l'entrée et une pente verticale à la sortie). J'ai pu alors construire une pièce en 3D à partir de ce profil à l'aide du logiciel CATIA.



FIGURE 20 – Mesure d'impédance du cor



FIGURE 21 – Main à l'intérieur du pavillon du cor

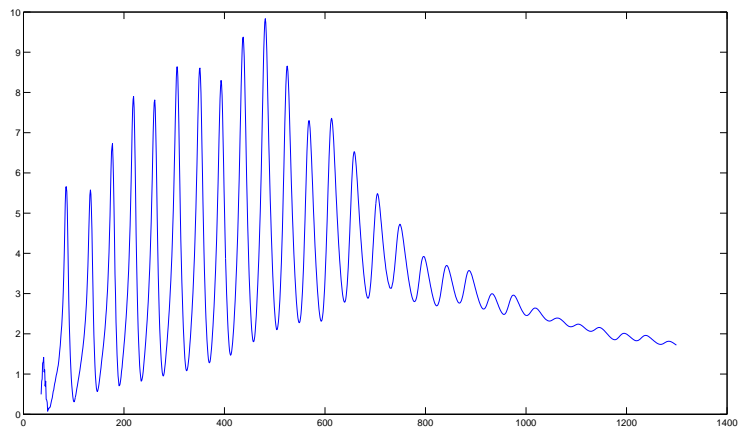
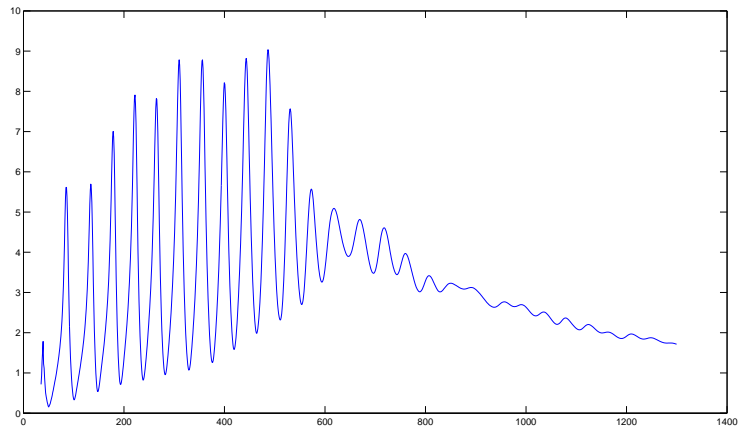


FIGURE 22 – Mesures sur le cor. En haut : sans boucher le pavillon. En bas : en bouchant le pavillon

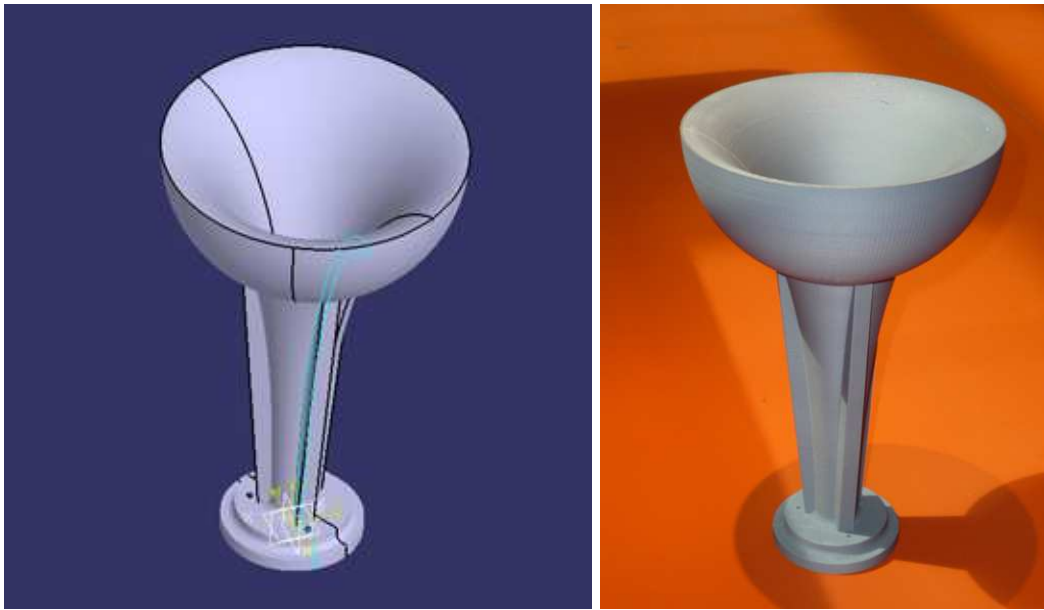


FIGURE 23 – Pièce fabriquée sur mesure. A gauche : modèle numérique CATIA. A droite : pièce réalisée

La figure 23 présente le modèle numérique CATIA ainsi qu'une réalisation de la pièce par prototypage rapide que nous avons commandé. La géométrie complexe de la pièce a été établie avec *Thomas Hélie*, *Rémi Mignot* et *Allain Terrier*, mécanicien de l'IRCAM chargé de fabriquer le système nécessaire aux différentes mesures que nous souhaitions effectuer sur ce pavillon.

Malheureusement, la pièce qui nous a été livrée présentait un état de surface très dégradé et une gorge à l'intérieur du tronçon qui rendaient la pièce inexploitable pour des mesures acoustiques précises. J'ai néanmoins fait les mesures sur ce profil avant de le renvoyer au constructeur. Comme on peut le voir sur la figure 24, ce profil ne présente pas du tout la courbe d'impédance attendue, l'amortissement est très important et les pics très déformés. Nous attendions pour ce profil des pics bien définis et réguliers présentant une décroissance régulière et assez lente.

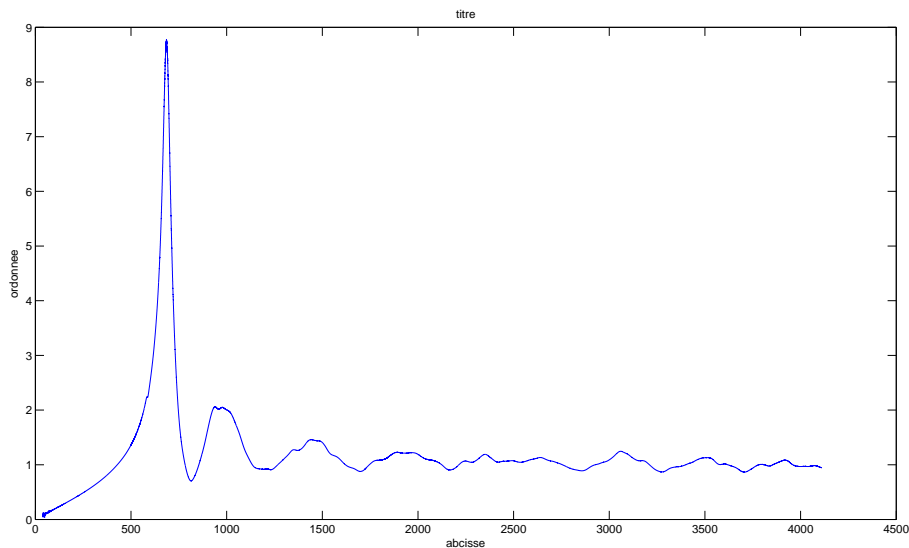


FIGURE 24 – Courbe d'impédance du prototype

Conclusion

Les objectifs de ce stage sont globalement remplis. Le programme de construction de structures simulables est fonctionnel et le programme de simulation temps-réel fonctionne maintenant avec des performances acceptables, bien qu'encore optimisables. Cependant, plusieurs problèmes subsistent : la concaténation en C++ lors du calcul du piston n'est pas encore entièrement fonctionnelle et les impédances simulées ne sont toujours pas très satisfaisantes. Une partie de recherche sur les approximations des représentations diffusives est en cours et devrait pouvoir être implémentée dans le logiciel prochainement. Il reste aussi beaucoup d'éléments que l'on pourrait ajouter à nos instruments, tels coulisses, trous etc. pour compléter notre famille d'instruments.

Le point le plus positif de ce stage est sans aucun doute les résultats obtenus par les premiers outils d'optimisation géométrique et les débuts de travaux sur les optimisations en impédance. Comme je l'ai mentionné plus haut, nous sommes déjà en contact avec des professionnels et des universitaires de la lutherie très intéressés par ces travaux. Il reste encore aujourd'hui à développer des outils véritablement efficaces et utilisables et surtout à entamer un réel dialogue avec des luthiers professionnels dans le but de comprendre leurs attentes et de rendre les outils logiciels accessibles à des artisans qui n'ont pas toujours le bagage scientifique nécessaire à la compréhension de la théorie à la base du fonctionnement du logiciel.

Les travaux de *Thomas Hélie* et *Rémi Mignot* dans lesquels s'est inscrit ce stage ont

plusieurs objectifs. Le premier est évidemment le développement d'instruments virtuels, basés sur la modélisation physique, de plus en plus réalistes. Mais nous allons encore plus loin avec les outils d'optimisation : il est désormais possible de faire un modèle numérique non pas d'"une" trompette mais de "cette" trompette, nous pouvons réaliser des véritables "clones" d'instruments destinés à la simulation temps-réel.

Ces travaux pourront avoir de nombreuses prolongations et j'espère pouvoir y prendre part dans la suite de mes études et/ou dans ma vie professionnelle future.

Annexes

A Structure simulable

A.1 Standardisation des représentations d'état

Le but de ce changement d'état est de partir de

$$\begin{cases} zX &= \sum_{n=0}^N \mathbf{A}_n X z^{-n} + \sum_{m=0}^M \mathbf{B}_m U z^{-m+1} \\ Y &= \mathbf{C} X + \mathbf{D} U \end{cases}$$

pour arriver à

$$\begin{cases} zW &= \mathbf{A}_w W + \mathbf{B}_w U \\ Y &= \mathbf{C}_w W + \mathbf{D}_w U \end{cases}$$

Les dimensions sont les suivantes :

$$\begin{cases} X &: J \times I \\ U &: K \times I \\ Y &: L \times I \end{cases}, \quad \begin{cases} \mathbf{A}_n &: J \times J \\ \mathbf{B}_m &: J \times K \\ \mathbf{C} &: L \times J \\ \mathbf{D} &: L \times K \end{cases}$$

Nous allons faire le calcul en deux temps. Dans un premier temps nous nous occuperons de réduire la récursion sur les matrices \mathbf{A}_n et dans un deuxième temps, nous réduirons la récursion sur les matrices \mathbf{B}_m .

Réduction de la récursion sur les matrices \mathbf{A}_n

On pose $\mathbf{B}(z) = \sum_{m=0}^M \mathbf{B}_m z^{-m+1}$.

Nous posons le changement d'état suivant :

$$X' = \begin{bmatrix} X \\ z^{-1} X \\ z^{-2} X \\ \vdots \\ z^{-N} X \end{bmatrix} = \begin{bmatrix} X'_0 \\ X'_1 \\ X'_2 \\ \vdots \\ X'_N \end{bmatrix}$$

On obtient alors :

$$\begin{aligned} zX &= \sum_{n=0}^N \mathbf{A}_n X z^{-n} + \mathbf{B}(z)U = zX'_0 = \sum_{n=0}^N \mathbf{A}_n X'_n + \mathbf{B}(z)U \\ \Leftrightarrow z(Xz^{-n}) &= zX'_n = z^{-(n-1)} = X'_{n-1}, \quad \forall n \geq 1 \end{aligned}$$

On en déduit le système suivant :

$$\begin{cases} z X' = \mathbf{A}' X' + \mathbf{B}'(z) U \\ Y = \mathbf{C}' X' + \mathbf{D}' U \end{cases}$$

avec

$$\mathbf{A}' = \left[\begin{array}{cccccc} \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_{n-1} & \mathbf{A}_N \\ \mathbf{Id}_J & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Id}_J & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{Id}_J & \mathbf{0} \end{array} \right] \left. \vphantom{\begin{array}{cccccc} \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{A}_{n-1} & \mathbf{A}_N \end{array}} \right\} (N+1) \times (N+1) \text{ blocs de taille } J \times J$$

$$\mathbf{B}'(z) = \left[\begin{array}{c} \mathbf{B}(z) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array} \right] \left. \vphantom{\begin{array}{c} \mathbf{B}(z) \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{array}} \right\} N+1 \text{ lignes par bloc de taille } J \times K$$

$$\mathbf{C}' = \underbrace{[\mathbf{C} \quad \mathbf{0} \quad \dots \quad \mathbf{0}]}_{N+1 \text{ colonnes par bloc de taille } L \times J}$$

$$\mathbf{D}' = \mathbf{D}$$

Réduction de la récursion sur les matrices \mathbf{B}_m

On écrit $\mathbf{B}'(z)$ sous la forme $\sum_{m=0}^M \mathbf{B}'_m z^{-m+1}$ avec $\mathbf{B}'_m = [\mathbf{B}_m \quad \mathbf{0} \quad \dots \quad \mathbf{0}]^T$ matrice de $N+1$ lignes par bloc de taille $J \times K$.

On fait le changement d'état

$$X'' = \begin{bmatrix} X' - \mathbf{B}'_0 U \\ U z^{-1} \\ U z^{-2} \\ \vdots \\ U z^{-M+1} \end{bmatrix} = \begin{bmatrix} X''_0 \\ X''_1 \\ X''_2 \\ \vdots \\ X''_{-M+1} \end{bmatrix}$$

On obtient alors

$$\begin{aligned} z X' &= \mathbf{A}' X' + \mathbf{B}'_0 U z + \sum_{m=1}^M \mathbf{B}'_m z^{-m+1} \\ \Leftrightarrow z(X' - \mathbf{B}'_0 U) &= \mathbf{A}'(X' - \mathbf{B}'_0 U) + (\mathbf{A}' \mathbf{B}'_0 + \mathbf{B}'_1) U + \sum_{m=2}^M \mathbf{B}'_m z^{-m+1} \\ \Leftrightarrow z X''_0 &= \mathbf{A}' X''_0 + \sum_{m=2}^M \mathbf{B}'_m X''_{m-1} + (\mathbf{A}' \mathbf{B}'_0 + \mathbf{B}'_1) U \end{aligned}$$

$$z(Uz^{-1}) = U = \mathbf{Id}_2 U$$

$$z(Uz^{-m}) = zX''_m = Uz^{-m+1} = X''_{m-1}, \forall m > 0$$

$$\begin{aligned} Y &= \mathbf{C}' X' + \mathbf{D}' U \\ &= \mathbf{C}' X' - \mathbf{C}' \mathbf{B}'_0 U + \mathbf{C}' \mathbf{B}'_0 U + \mathbf{D}' U \\ &= \mathbf{C}' X''_0 + \mathbf{C}' \mathbf{B}'_0 U + \mathbf{D}' U \end{aligned}$$

On en déduit le système final :

$$\begin{cases} zX'' = \mathbf{A}'' X'' + \mathbf{B}'' U \\ Y = \mathbf{C}'' X'' + \mathbf{D}'' U \end{cases}$$

avec

$$\mathbf{A}'' = \begin{bmatrix} \mathbf{A}' & \mathbf{B}'_2 & \mathbf{B}'_3 & \dots & \mathbf{B}'_{M-1} & \mathbf{B}'_M \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Id}_K & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{Id}_K & \mathbf{0} \end{bmatrix}$$

une matrice $M \times M$ blocs de tailles

$$\begin{cases} \mathbf{A}' : (J \times (N+1)) \times (J \times (N+1)) \\ \mathbf{B}'_m : (J \times (N+1)) \times K \\ \mathbf{0} \text{ sur la première colonne} : K \times (J \times (N+1)) \\ \mathbf{0} \text{ sur les suivantes} : K \times K \end{cases},$$

$$\mathbf{B}'' = \left. \begin{bmatrix} \mathbf{A}' \mathbf{B}'_0 + \mathbf{B}'_1 \\ \mathbf{Id} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \right\} M \text{ lignes par blocs}$$

$$\begin{cases} \text{première ligne} : (J \times (N+1)) \times K \\ \text{suivantes} : K \times K \end{cases},$$

$$\mathbf{C}'' = [\mathbf{C}' \quad \mathbf{0} \quad \dots \quad \mathbf{0}] \} M \text{ colonnes par blocs}$$

$$\begin{cases} \mathbf{C}' : L \times (J \times (N+1)) \\ \mathbf{0} : L \times K \end{cases},$$

$$\mathbf{D}'' = (\mathbf{C}' \mathbf{B}'_0 + \mathbf{D}') U \text{ de taille } L \times K.$$

A.2 Débouclage partiel

Nous partons du système original de la figure 12 :

$$\begin{cases} X_{n+1} = \mathbf{A}X_n + \mathbf{B}U_n = \mathbf{A}X_n + \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} \\ Y_n = \mathbf{C}X_n + \mathbf{D}U_n = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} X_n + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} \end{cases}$$

Nous construisons un système historique ayant pour sortie $W_n = [y_1(n), y_2^h(n+1)]^T = [w_1(n), w_2(n)]^T$ et pour entrée $E_n = [u_1(n+1), u_2(n)]^T = [e_1(n), e_2(n)]^T$ où $y_2^h(n) = \mathbf{C}_2 X_n + D_{21} \cdot u_1(n)$. On retrouve donc bien $y_2(n) = y_2^h(n) + D_{22} \cdot u_2(n)$. Il est nécessaire d'avoir un échantillon d'avance dans la deuxième sortie du système historique car la cellule historique est simulée séparément de la cellule directe concaténée avec une partie variable, introduisant alors un retard d'un échantillon à l'interface des deux cellules. Nous devons donc supprimer un retard du tampon connecté à la première entrée de la cellule. Lors de la simulation, $y_2(n)$ et $y_2^h(n+1)$ seront calculés dans le même tour de boucle indépendamment.

Par la suite, j'utiliserai la notation z^{-1} pour symboliser un retard en variable temporelle.

Première étape de calcul

$$\begin{aligned} y_1(n) &= \mathbf{C}_1 X_n + [D_{11} D_{12}] \begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} = w_1(n) \\ &= \mathbf{C}_1 X_n + [D_{11} z^{-1} D_{12}] \begin{bmatrix} e_1(n) \\ e_2(n) \end{bmatrix} \\ &= \mathbf{C}_1 X_n + [D_{11} z^{-1} D_{12}] E_n \\ \\ y_2^h(n+1) &= \mathbf{C}_2 X_{n+1} + D_{21} u_1(n+1) = w_2(n) \\ &= \mathbf{C}_2 \mathbf{A} X_n + \mathbf{C}_2 \mathbf{B} U_n + D_{21} e_1(n) \\ &= (\mathbf{C}_2 \mathbf{A}) X_n + [\mathbf{C}_2 \mathbf{B}_1 z^{-1} + D_{21} \quad \mathbf{C}_2 \mathbf{B}_2] E_n \\ \\ X_{n+1} &= \mathbf{A} X_n + [\mathbf{B}_1 \quad \mathbf{B}_2] U_n = \mathbf{A} X_n + [\mathbf{B}_1 z^{-1} \quad \mathbf{B}_2] E_n \end{aligned}$$

On obtient donc :

$$\begin{cases} X_{n+1} = \mathbf{A} X_n + \mathbf{B}^h E_n \\ W_n = \mathbf{C}^h X_n + \mathbf{D}^h E_n \end{cases}$$

avec

$$\begin{aligned} \mathbf{B}^h &= [\mathbf{B}_1 z^{-1} \quad \mathbf{B}_2] \\ \mathbf{C}^h &= \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \mathbf{A} \end{bmatrix} \\ \mathbf{D}^h &= \begin{bmatrix} D_{11} z^{-1} & D_{12} \\ \mathbf{C}_2 \mathbf{B}_1 z^{-1} + D_{21} & \mathbf{C}_2 \mathbf{B}_2 \end{bmatrix} \end{aligned}$$

On obtient alors en sortie du système compét :

$$Y_n = \mathbf{Id}_2 W_n + \begin{bmatrix} 0 & 0 \\ 0 & D_{22} \end{bmatrix} U_n$$

D_{22} est appelée sensibilité du système.

Deuxième étape de calcul Dans le but de supprimer les z^{-1} des matrices du système historique, on procède à un changement d'état :

$$X'_n = \begin{bmatrix} X_n \\ e_1(n-1) \end{bmatrix}$$

Ce qui nous donne :

$$\begin{cases} X'_{n+1} = \mathbf{A}^{h'} X'_n + \mathbf{B}^{h'} E'_n \\ W_n = \mathbf{C}^{h'} X'_n + \mathbf{D}^{h'} E'_n \end{cases}$$

avec

$$\begin{aligned} \mathbf{A}^{h'} &= \begin{bmatrix} \mathbf{A} & \mathbf{B}_1 \\ \mathbf{0} & 0 \end{bmatrix} \\ \mathbf{B}^{h'} &= \begin{bmatrix} \mathbf{0} & \mathbf{B}_2 \\ 1 & 0 \end{bmatrix} \\ \mathbf{C}^{h'} &= \begin{bmatrix} \mathbf{C}_1 & D_{11} \\ \mathbf{C}_2 \mathbf{A} & \mathbf{C}_2 \mathbf{B}_1 \end{bmatrix} \\ \mathbf{D}^{h'} &= \begin{bmatrix} 0 & D_{12} \\ D_{21} & \mathbf{C}_2 \mathbf{B}_2 \end{bmatrix} \end{aligned}$$

B Optimisation géométrique

B.1 Formules de passage

B.1.1 Variables dimensionnées

Profil cible

$$\begin{aligned} \ell &= \frac{x L_n + \lambda_n + \lambda_{n+1}}{2} \\ \tilde{\alpha}_k^n &= \frac{2}{L_n} \alpha_k^n \end{aligned}$$

Profil modèle

$$\tilde{\Upsilon}_n = \frac{4}{L_n^2} \Upsilon_n$$

$$\tilde{A}_n = \mathbf{V}_n^T \begin{pmatrix} 1 & 0 \\ 0 & -\frac{\lambda_n + \lambda_{n+1}}{L_n} \end{pmatrix} \mathbf{U}_n ,$$

$$\tilde{B}_n = \mathbf{V}_n^T \begin{pmatrix} 0 & \frac{2}{L_n} \\ -\frac{\tilde{\Upsilon}_n(\lambda_n + \lambda_{n+1})}{2} & 0 \end{pmatrix} \mathbf{U}_n = \mathbf{V}_n^T \begin{pmatrix} 0 & \frac{2}{L_n} \\ -\frac{2\Upsilon_n(\lambda_n + \lambda_{n+1})}{L_n^2} & 0 \end{pmatrix} \mathbf{U}_n$$

où

$$\begin{aligned} \mathbf{V}_n &= \left[\phi_1 \left(\tilde{\Upsilon}_n \frac{(\lambda_n + \lambda_{n+1})^2}{4} \right) \quad \phi_2 \left(\tilde{\Upsilon}_n \frac{(\lambda_n + \lambda_{n+1})^2}{4} \right) \right]^T \\ &= \left[\phi_1 \left(\Upsilon_n \frac{(\lambda_n + \lambda_{n+1})^2}{L_n^2} \right) \quad \phi_2 \left(\Upsilon_n \frac{(\lambda_n + \lambda_{n+1})^2}{L_n^2} \right) \right]^T \end{aligned}$$

B.1.2 Variables adimensionnées

Profil cible

$$x = \frac{2\ell - \lambda_n - \lambda_{n+1}}{L_n}$$

$$\underline{\alpha}_k^n = \frac{L_n}{2} \tilde{\underline{\alpha}}_k$$

Profil modèle

$$\Upsilon_n = \frac{L_n^2}{4} \tilde{\Upsilon}_n$$

$$A_n = \mathbf{V}_n^T \begin{pmatrix} 1 & 0 \\ 0 & \frac{\lambda_n + \lambda_{n+1}}{2} \end{pmatrix} \tilde{\mathbf{U}}_n ,$$

$$B_n = \mathbf{V}_n^T \begin{pmatrix} 0 & \frac{L_n}{2} \\ \frac{\Upsilon_n(\lambda_n + \lambda_{n+1})}{L_n} & 0 \end{pmatrix} \tilde{\mathbf{U}}_n = \mathbf{V}_n^T \begin{pmatrix} 0 & \frac{L_n}{2} \\ \frac{\tilde{\Upsilon}_n L_n (\lambda_n + \lambda_{n+1})}{4} & 0 \end{pmatrix} \tilde{\mathbf{U}}_n$$

où

$$\begin{aligned} \mathbf{V}_n &= \left[\phi_1 \left(\Upsilon_n \frac{(\lambda_n + \lambda_{n+1})^2}{L_n^2} \right) \quad \phi_2 \left(\Upsilon_n \frac{(\lambda_n + \lambda_{n+1})^2}{L_n^2} \right) \right]^T \\ &= \left[\phi_1 \left(\tilde{\Upsilon}_n \frac{(\lambda_n + \lambda_{n+1})^2}{4} \right) \quad \phi_2 \left(\tilde{\Upsilon}_n \frac{(\lambda_n + \lambda_{n+1})^2}{4} \right) \right]^T \end{aligned}$$

B.2 Détails des calculs de la récurrence sur \mathbf{U}_n

$$\forall n \in \llbracket 1, N-1 \rrbracket \begin{cases} \tilde{r}_n(\ell_{K_n}^n) = \tilde{r}_{n+1}(\ell_1^n) \\ \partial_\ell \tilde{r}_n(\ell_{K_n}^n) = \partial_\ell \tilde{r}_{n+1}(\ell_1^n) \end{cases}$$

s'écrit en variable adimensionnée

$$\begin{cases} A_n C_{\Upsilon_n}(1) + B_n S_{\Upsilon_n}(1) = A_{n+1} C_{\Upsilon_{n+1}}(-1) + B_{n+1} S_{\Upsilon_{n+1}}(1) \\ \frac{A_n}{L_n} \Upsilon_n S_{\Upsilon_n}(1) + \frac{B_n}{L_n} C_{\Upsilon_n}(1) = \frac{A_{n+1}}{L_{n+1}} \Upsilon_{n+1} S_{\Upsilon_{n+1}}(-1) + \frac{B_{n+1}}{L_{n+1}} C_{\Upsilon_{n+1}}(-1) \end{cases}$$

soit en écriture matricielle

$$(\mathbf{\Lambda}_n)^{-1} \mathbf{\Phi}_n \mathbf{U}_n = (\mathbf{\Lambda}_{n+1})^{-1} (\mathbf{\Phi}_{n+1})^{-1} \mathbf{U}_{n+1}$$

où

$$\mathbf{\Lambda}_n = \begin{pmatrix} 1 & 0 \\ 0 & L_n \end{pmatrix}$$

et

$$\mathbf{\Phi}_n = \begin{pmatrix} C_{\Upsilon_n}(1) & S_{\Upsilon_n}(1) \\ \Upsilon_n S_{\Upsilon_n}(1) & C_{\Upsilon_n}(1) \end{pmatrix} = \begin{pmatrix} \phi_1(\Upsilon_n) & \phi_2(\Upsilon_n) \\ \Upsilon_n \phi_2(\Upsilon_n) & \phi_1(\Upsilon_n) \end{pmatrix}.$$

De plus, $\det(\mathbf{\Phi}_n) = 1$ et

$$(\mathbf{\Phi}_n)^{-1} = \begin{pmatrix} C_{\Upsilon_n}(-1) & S_{\Upsilon_n}(-1) \\ \Upsilon_n S_{\Upsilon_n}(-1) & C_{\Upsilon_n}(-1) \end{pmatrix} = \begin{pmatrix} \phi_1(\Upsilon_n) & -\phi_2(\Upsilon_n) \\ -\Upsilon_n \phi_2(\Upsilon_n) & \phi_1(\Upsilon_n) \end{pmatrix}.$$

On retrouve bien l'équation de récurrence

$$\mathbf{U}_{n+1} = \mathbf{\Phi}_{n+1} \mathbf{G}_n \mathbf{\Phi}_n \mathbf{U}_n, \text{ où } \mathbf{G}_n = \mathbf{\Lambda}_{n+1} \mathbf{\Lambda}_n^{-1},$$

et on en déduit la formule explicite de \mathbf{U}_n en fonction de \mathbf{U}_1

$$\mathbf{U}_n = \mathbf{\Phi}_n \prod_{p=1}^{n-2} \left[\mathbf{G}_{n-p} \mathbf{\Psi}_{n-p} \right] \mathbf{G}_1 \mathbf{\Phi}_1 \mathbf{U}_1, \text{ où } \mathbf{\Psi}_n = (\mathbf{\Phi}_n)^2.$$

Références

- [HMM07] Thomas Hélie, Rémi Mignot and Denis Matignon *Waveguide Modeling of Lossy Flares Acoustic Pipes : Derivation of Kelly-Lochbaum Structure for Real Time Simulation* Workshop on Application of Signal Processing to Audio and Acoustics (WASPAA), 2007
- [MHM08] Rémi Mignot, Thomas Hélie and Denis Matignon *From the Webster Lokshin Equation to a General Framework for Simulation of Digital Waveguides*, 2008
- [MG73] J. D. Markel & A. H. Gray *On autocorrelation equations as applied to speech analysis*, IEEE Trans. Audio and Electroacoust., vol. AU-ZA, no. 2, 04/1973
- [VV95] V. Välimäki *Discrete-time modeling of acoustic tubes using fractional delay filters*, Ph.D dissertation, Helsinki University of Technology, 1995
- [HV08] Thomas Hélie and Christophe Vergez *Des instruments de musique virtuels* Paru dans **Pour la Science-373** de Novembre 2008
- [TH02] Thomas Hélie. *Modélisation physique des instruments de musique en système dynamique et inversion (Physical modeling of musical instruments with dynamique systems and inversion processes)*, Ph.D. thesis, Université Paris-Sud, Orsay, France, 2002
- [FR91] N. H. Fletcher and T. D. Rossing, *Physics of musical instruments*, Springer Verlag, New York, 1991
- [HHM03] H. Haddar, Th. Hélie, and D. Matignon, *A webster-lokshin model for waves with viscotermal losses and impedance boundary conditions : strong solutions*, 6th International Conference on Mathematical and Numerical Aspects of Wave Propagation Phenomena, INRIA, 2003, pp. 66-71
- [RM05] Rémi Mignot *Simulation de propagations d'ondes dans les tubes évasés avec pertes visco-thermiques pour la synthèse sonore temps réel*, Rapport de stage de Master 1 à l'IRCAM, Paris, 2005
- [RM06] Rémi Mignot *Modélisation de résonateur d'instruments à vents*, Rapport de stage de Master 2 ATIAM, IRCAM, Paris, 2006
- [TD06] Th. Hélie and De. Matignon *Diffusive Representations for the analysis and simulation of flared acoustics pipes with visco-thermal losses*, Math. Models and Methods in Applied Sciences (M3AS) 2006
- [ST98] St. TASSART *Modélisation, simulation et analyse des instruments à vent avec retards fractionnaires* Thèse, Paris 6
- [EW97] Eric Walter *Identification de modèles paramétriques à partir de données expérimentales* Edition Dunod, 1997
- [PDD08] Pierre-Damien Dekoninck *Lutherie virtuelle et synthèse sonore d'instruments à vent* Rapport de stage ingénieur INSA Rouen à l'IRCAM, Paris, 2008
- [CV98] Christophe Vergez et Xavier Rodet *Experiments with an artificial mouth for trumpet* ISMA98, Leavenworth, Washington State, USA, 1998

- [GM98] G. Montseny *Diffusive representation of pseudo-differential time-operator*
ESAIM : Proc, vol. 5, 1998
- [DM98] D. Matignon *Stability properties for generalized fractional differential systems*
ESAIM : Proc, vol.5, 1998
- [HMM07bis] T. Hélie, D. Matignon et R. Mignot Criterion design for optimizing low-cost approximations of infinite-dimensional systems : towards efficient real-time simulation, *Int. Journal of Tomography and Statistics*, volume 7, 2007
- [FPW90] G. F. Franklin and J. D. Powell and M. L. Workman *Digital Control of Dynamic Systems*, 1990