PARCOURS MASTER 2
ATiAM
Parcours multi-mentions du Master Sciences et Technologies
Université Pierre et Marie Curie - Paris 6
en collaboration avec TELECOM ParisTech et l'Ircam

ircam
Centre
Pompidou

Rapport de stage
Master 2 atiam

Suivi de Tempo appliqué aux Musiques Improvisées

Auteur :
Vincent Roggerone

Responsables :
Arshia Cont
Gilbert Nouno

1 aout 2014

Remerciement

Je voudrais remercier en particulier mes deux maîtres de stage Arshia Cont et Gilbert Nouno, chercheurs à l'IRCAM, pour m'avoir fait confiance sur ce projet. Je les remercie pour m'avoir laissé mon indépendance et ma liberté de travail, mais aussi pour leurs conseils et éclairements avisés ainsi que leur intérêt pour mon travail et notre réussite de stage.

Je voudrais aussi remercier le Master ATIAM, autant l'équipe et les enseignants, qui font preuve d'une écoute et d'une attention pour leurs élèves qu'il m'a rarement été donné de voir dans ma scolarité, que les élèves et amis issus du master faisant sans cesse plus preuves de curiosités et d'entraide.

Je remercie également Helianthe Caure de l'IRCAM pour son aide et sa patience pour m'apprendre à faire des maths "propres" et Jonathan Vacher de l'ENS Cachan pour son aide sur les maths disons plutôt "sales".

Merci aussi à Jérome Nika pour m'avoir aidé à constituer un corpus sur lequel j'ai pu tester mes algorithmes, Rémi Boronad pour son aide pointue dans la langue de Shakespeare, et Clémence Ménis pour son aide tout aussi pointue dans la langue de Molière.

Je voudrais enfin remercier Gwennaëlle Le Bars, Anais Acquaviva et Thomas Pagesy pour avoir supporté un colloc ne jurant plus que par son stage et son rapport ; Juliette Delpeyroux pour sa "customisation" mythique, voir mystique de mon clavier qui n'a rendu la rédaction de ce rapport que plus agréable, et mes parents, parce qu'il faut toujours remercier ses parents.

Résumé / Abstract

Résumé

Le suivi de tempo en temps réel peut trouver de nombreuses applications comme l'aide à la synchronisation en studio, ou l'aide à la synchronisations entre les différents musiciens et techniciens lors d'une performance musicale live. C'est également une analyse indirecte de notre capacité à suivre un rythme musical. Cette question est abordée dans de nombreux outils de recherche d'informations musicales. On se place ici dans le contexte général de la musique improvisée, c'est-à-dire sans connaissance a priori de la partition, et on cherche à suivre non pas le tempo en lui-même mais une de ses sous métriques qu'est le tatum. Après une revue de la littérature sur l'état de l'art concernant le suivi de rythme musical et une présentation et analyse de la méthode de recherche de la périodicité du tatum proposé par Gilbert Nouno dans sa thèse, on propose une application de cette méthode en temps réel. Pour ce faire, on propose dans un premier temps différentes modifications de la méthode originale. Puis plusieurs façons de transformer et re-construire les résultats obtenus pour que ceux-ci deviennent comparables entre eux et puissent s'appliquer au temps réel. On rajoute un traitement statistique par le biais d'un filtrage de Kalman. Et l'on finit par différentes méthodes pour permettre la reconstruction de la grille en temps réel. Les différentes méthodes sont ensuite comparées entres elles sur différentes séquences réelles ou construites.

Mots clés : Rythme musical, Tempo, Métrique musicale, Traitement du signal musical, Processus d'estimation, Filtre de Kalman, Temps réel, Musiques improvisées, extraction d'information à partir d'attaques, séries temporelles, Mesure de distance

Abstract

Tempo following in real time can be used in many ways. Some examples could be the synchronization of several tracks in studio for musicians and technicians in a live performance. Tempo following is also an indirect analysis of our ability to extract metrical information from a musical content. Several fields of Music Information Retrival are using it. Here, we deal with an improvised music context, which means we don't have any information about the score. Besides, We try to follow the value of "tatum" and not the Tempo. An overview of the state-of-art about tempo following, and a presentation and analysis of Gilbert Nouno method is first given. Then we propose to apply this method in real time. For that, we introduce several changes in the original

method, and also propose several ways to transform and construct results to make them suitable for real time. Then we perform a Kalman filter on results, and finally propose several ways to construct a real time grid from our tatum estimation. All this methods are finally evaluated et compared between them

Keywords : Musical meter, Beat, Musical signal processing, Estimation process, Kalman Filtering, Real Time, Improvised music, Onsets information extraction, temporal series, Distance mesure.

Table des matières

1 Introduction et Présentation du sujet **7**

1.1 Cadre 7

1.2 Contexte 7

1.3 Objectif du stage 8

1.4 Applications possibles 9

2 Etat de l'Art et description de la méthode : l'Algorithme de grille optimale **10**

2.1 Etat de l'Art du suivi de Tempo 10

2.1.1 Bref historique du suivi de Tempo 10

2.1.2 Définition de quelque concepts nécessaires 11

2.1.3 Principes généraux de l'analyse rythmique automatique 12

2.2 Principe originale de l'inférence de tempo sur une grille 14

2.2.1 Principe de la mesure de Nouno 15

2.2.2 Grille optimale et théorème fondamental 16

2.3 Premières modifications de l'algorithme pour l'adaptation en temps réel 19

2.3.1 Taille de grille et Intervalle de recherche 19

2.3.2 Considérations sur les mesures et taille et de la taille de la séquence d'entrée 21

2.3.3 Nouvel algorithme 22

2.4 Considération sur l'accélération 23

2.4.1 Notion et formalisme de Courbe de Tempo 24

2.4.2 Variation linaire du Tempo 26

3 Suivi de tatum en temps réel **31**

3.1 Généralité 31

3.1.1 Extraction de série temporelle 31

3.1.2 Paramètrisation des fenêtres 32

3.1.3 Construction du vecteur d'état . . . 33
3.2 Methode de construction du vecteur d'état . . . 36
3.2.1 Interpolation . . . 36
3.2.2 Estimation par noyau . . . 37
3.2.3 Normalisation des mesures . . . 40
3.3 Filtrage de Kalman . . . 41
3.3.1 Principe du Filtrage de Kalman . . . 41
3.3.2 Mise en application . . . 42
3.4 Reconstruction de la phase . . . 43
3.4.1 Mise à jour de la phase à chaque fenêtre . . . 44
3.4.2 Mise à jour à chaque événement : l'oscillateur de Larges et Jones . . . 47

4 Resultats obtenus **51**
4.1 Mise en place d'une mesure simple . . . 51
4.2 Cas de séquence d'entrée construites . . . 52
4.2.1 Grille bruité . . . 52
4.2.2 Saut de tempo . . . 53
4.2.3 Décélération de tempo . . . 56
4.3 Cas réel . . . 59
4.3.1 Cas studio (pulse parfaite) . . . 62
4.3.2 Cas improvisé . . . 64

5 Conclusion et Perspectives **66**

Bibliographie **69**

A Modification de l'espace temporel pour une courbe de Tempo donnée **71**
A.1 Définition du problème . . . 71
A.2 Résolution du problème, exprimer t_j en fonction de τ et d_i . . . 74

B Filtrage adaptatif, Méthode basés sur les résidus et méthode basé sur les innovations **76**
B.1 Estimation de $\mathbf{R}_p$ . . . 76
B.1.1 Méthode basée sur l'innovation [R70], [R71], [AK99] . . . 76
B.1.2 Méthode basée sur les résidus de la séquence, [J00] . . . 77

B.2 Estimation de $\mathbf{Q}_p$ 77
B.2.1 Méthode Commune aux deux méthodes, [DWR07] 77

Chapitre 1

Introduction et Présentation du sujet

1.1 Cadre

Ce Stage a été effectué dans le cadre du Master 2 ATIAM (*Acoustique, Traitement du Signal et Informatique Appliqué à la Musique*) de l'UPMC (*Université Pierre et Marie Curie*), en partenariat avec l'IRCAM (*Institut de Recherche et Coordination Acoustique/Musique*). Il s'est déroulé dans l'équipe **MuTant** (projet commun de l'IRCAM, INRIA et le CNRS) dont le projet de recherche se situent à la confluence de deux problématiques importantes en informatique musicale :

- La reconnaissance et l'extraction des données musicales en temps réel depuis un signal audio (écoute artificielle)
- La programmation synchrone réactive en informatique musicale.

Le couplage de ces deux thématiques, souvent considérées comme distinctes, est au coeur de la pratique musicale (la composition et la performance). L'étude de ce couplage à pour objectif de permettre a un ordinateur de s'adapter à un environnement musical pour lui permettre de réagir et se synchroniser avec celui-ci, à l'image de l'activité humaine des compositeurs et interprètes lors d'une performance par exemple. L'équipe a conçu le logiciel ANTESCOFO, lauréat du Prix de la recherche 2011.

Ce stage s'inscrit dans cette problématique en s'intéressant une méthode d'inférence de Tempo en temps réel basé sur le formalisme développé par Gilbert Nouno dans sa thèse [Nou08], dont il est la suite directe. Il a permis notamment d'apporter des réponses sur la manière d'implémenter la méthode de manière efficace, en apportant des modifications sur l'algorithme de base ainsi qu'en rajoutant un traitement statistique d'apprentissage automatique.

1.2 Contexte

Un des défis majeurs en informatique est de synchroniser des processus informatiques temporels avec ceux provenant d'un environnement extérieur. La musique représente un cas particulièrement intéressant pour l'étude de ces processus où la variation en temps réel du jeu des musiciens pose des problèmes majeurs de conception pour les modèles mathématiques sous-

jacents. Le problème de suivi de tempo consiste alors à estimer et à suivre en temps réel le tempo d'un musicien à partir d'une série temporelle extraite d'une performance musicale. Différentes approches concernant le suivi de tempo ont été étudiées dans la littérature notamment résumées par Collins [CD04] à des fins d'utilisation en temps réel ou d'analyses. Cependant, ces approches sont en général basées sur la connaissance de la partition [Con08] , sur des analyses fréquentielles [Cem01], [Cem03], [Pee05] ou sur des méthodes basées sur des oscillateurs [LJ99].

Récemment, Gilbert Nouno [Nou08] propose une méthode originale basée sur une série temporelle d'attaque via un modèle mathématique de grille optimale qui coïncide le mieux possible avec la distribution des données observées. La méthode est développée pour s'appliquer aux musiques improvisées (en absence d'une partition musicale) et se base sur une discrétisation des candidats au tempo qui découlent du modèle posé. La notion de grille optimale est fondée sur une mesure de distance non linéaire entre la série observée et les différents candidats pour la grille optimale. Cette mesure a l'avantage de réduire considérablement le nombre de candidats possibles au tempo.

Cette méthode, pour être appliquée au problème du temps réel, doit faire l'objet d'ajout de traitements statistiques comme nous en trouvons dans la littérature [May79], [Mur02]. Ces traitements permettront une plus grande interaction entre les différents agents (musiciens, instruments, processus électroniques, commandes sur scène, etc.) et une meilleure prise en compte de leur variabilité.

1.3 Objectif du stage

Dans le but de tester la méthode en temps réel, il s'agira ici de :

- Comprendre et s'approprier la méthode, pour produire un ensemble de données exploitables. Cela implique notamment des notions de choix sur les différentes constantes que l'on peut utiliser, et le formalisme que l'on va retenir sur la méthode. A titre d'exemple, on peut se poser la question de la quantité d'informations nécessaires pour un résultat fiable, ou encore la pertinence et priorité des différentes estimations produites par la mesure.
- Adapter la méthode pour rendre ses résultats comparables. Il faut pour cela que les résultats produits par la méthode adaptée soient comparables entre eux, ou développer un traitement pour les rendre comparables.
- Ajouter un traitement statistique d'apprentissage automatique. Pour rendre la méthode utilisable, il faut pouvoir prendre en compte le passé dans notre estimation du tempo, il s'agit de déterminer par exemple si on est en présence d'une variation de tempo ou d'une imprécision du musicien. Les techniques suggérées sont le filtrage particulaire, le filtrage de Kalman étendu et de manière générale les inférences dans un contexte bayesien dynamique.
- Etre capable de re-construire une grille en temps réel, en tenant compte de la dimension causale due à la notion de temps réel.

– Produire un code qui puisse simuler le temps réel et ainsi comparer les résultats obtenus avec différentes approches, puis comparer les résultats avec ceux de la littérature. Cela implique le développement de certains critères (numérique ou perceptif) sur lesquels s'appuieront la comparaison.

1.4 Applications possibles

Il existe de nombreuses applications à l'inférence en temps réel d'un Tempo.

L'application la plus évidente prend son contexte dans le cadre d'une performance musicale. La machine, par l'intermédiaire de boucle ou de "sample" est de plus en plus présente dans notre environnement musical. Dans le cas où celle-ci interagit avec des musiciens, ceux-ci sont alors contraints de suivre le tempo imposé par la machine. Un algorithme d'inférence de Tempo pourrait alors renverser cette hiérarchie, permettant aux musiciens par leur propre jeu d'imposer leur tempo à la machine. Cela laisse une plus grande liberté au musicien qui peut alors jouer avec des variations de tempo, comme c'est couramment le cas dans les orchestres de Jazz par exemple. On peut citer les mêmes arguments lors d'une prise studio, ou la production d'un "click" où l'adaptation de la machine au musicien pourrait accélérer les enregistrements.

Dans un cadre similaire, trouver le tempo en temps réel d'une performance musicale pourrait permettre de synchroniser tout les "à coté" qui composent un concert. Ainsi, on peut imaginer un éclairage dont le rythme se base sur le Tempo à l'inverse du son brut comme c'est le cas aujourd'hui. La prédiction d'un tempo pourrait permettre des motifs lumineux plus complexes.

Enfin, il peut exister aussi différentes applications moins axées sur la performance professionnelle, dont certaines ont déjà été mentionnées dans la littérature, comme un karaoke qui s'adapte a la vitesse du chanteur et non l'inverse [Got01] ; ou les chaussures de footing qui adaptent la musique écoutée par le joggeur au rythme de ses pas [JAHF09]

Chapitre 2

Etat de l'Art et description de la méthode : l'Algorithme de grille optimale

2.1 Etat de l'Art du suivi de Tempo

2.1.1 Bref historique du suivi de Tempo

L'estimation du tempo ou de la position des temps a explosé ces 30 dernières années avec l'arrivée d'ordinateurs de plus en plus puissants permettant des estimations de plus en plus fines. Ces travaux se sont d'abord attachés à reproduire la battue humaine, puis des systèmes d'estimation automatique ont été mis en place progressivement d'abord sur des signaux MIDI puis sur des signaux audio.

Ainsi, ces systèmes sont basés en premier lieu sur des modèles perceptifs qui proposent des explications sur notre capacité à percevoir et prévoir la structure temporelle d'une suite d'évènements. Drake et Botte [DcB93] proposent ainsi une approche statistique, convaincante en terme de précision rythmique, mais insuffisante dans le cas de rythme complexes. Povel et Essens [PE85] proposent une approche basée sur l'encodage de rythme, mais très vite insuffisante si on rajoute de l'incertitude sur la position des attaques. [Large et Jones] proposent une approche par réseaux oscillateurs résonnants pour tenter de combler les lacunes des deux précédents modèles, mais qui ne fonctionne que pour une répétition suffisante des motifs dans le temps couplée à un grand nombre d'oscillateurs.

D'autres raffinements ont ensuite été inclus pour petit à petit intégrer des modèles sur nos a priori culturels (perception des premiers temps par exemple [Par94], des modèles probabilistes et [Kla03b]ou des méthodes d'apprentissage automatique [AHG12] [Cem03]. La plupart de ces modèles ne sont pas causals et s'utilisent hors ligne ou après un entrainement suffisant sur un corpus annoté pour le cas de l'apprentissage automatique, et sortent donc de notre sujet ici.

L'utilisation de l'audio brut a impliqué des algorithmes de détection d'attaques ou d'accents

(pour retomber sur un format de type MIDI), mais a aussi impliqué le développement de méthodes originales basées sur de la décomposition en sous-bandes avec couplage à des réseaux d'oscillateurs [Kla03a]. Ces méthodes sont toujours utilisées aujourd'hui et sont de plus en plus performantes.

On peut finalement catégoriser l'analyse rythmique en 4 grandes étapes générales décrites dans la section 2.1.3. On ne traite dans ce stage que deux d'entre elles, dont plus de détails peuvent être trouvés dans la section 2.1.3.

2.1.2 Définition de quelque concepts nécessaires

Dans les définitions données ici, le terme "temps" correspond aux différents temps que comporte une mesure. Il correspond en général à la battue ou pulsation que l'on fait en écoutant un morceau, à sa position temporelle près que l'on appelle "phase".

Rythme musicale

Il n'existe pas de consensus communément accepté pour définir le rythme musical. En revanche, il est admis par la communauté des chercheurs que l'on peut mettre en relation un événement sonore avec celui qui le précède et celui qui le suivra probablement. Notre mémoire et notre attente (ou prédiction) constitue donc notre manière d'organiser les événements dans le temps et d'extraire le rythme d'une musique. Ces deux paramètres fondamentaux forment d'ailleurs la base de l'oscillateur non linéaire de Large et Jones. Remarquons que, comme l'on peut évaluer la position des temps, voir des temps forts (premier temps d'une mesure en général, mais pas tout le temps) cela suppose une structurelle temporelle de la musique à différentes échelles, au delà de la considération des derniers événements.

Les travaux de Lerdhal et Jackendoff [FLS85] montrent une organisation du rythme en deux catégories :

- **Le groupement** est la combinaison de motifs rythmiques du durée équivalente pour former des unités musicales cohérentes de plus en plus longues jusqu'à obtenir la taille totale du morceau. Un exemple d'une telle organisation est la découpe de la musique populaire en "couplet", "refrain", "pont" ... mais aussi la structure au sein même des couplets, ou même au sein d'un album complet si celui-ci est vu comme une pièce complète (citons pink floyd).
- **La métrique** qui est une notion plus perceptive, sous entendant un découpage temporel régulier de la musique. Un niveau métrique est ainsi un choix de découpage de la musique (temps, premiers temps, tactus, tatum ... voir section 2.1.2. Cette notion perdure même si les événements s'arrêtent, comme c'est le cas pour un "break" dans la musique populaire par exemple.

Dans notre cas, nous nous intéressons uniquement à la notion de métrique. L'objectif est en effet de pouvoir trouver un niveau métrique particulier, sans prendre en compte le groupement qui n'a pas d'intérêt dans le cas d'un suivi en temps réel.

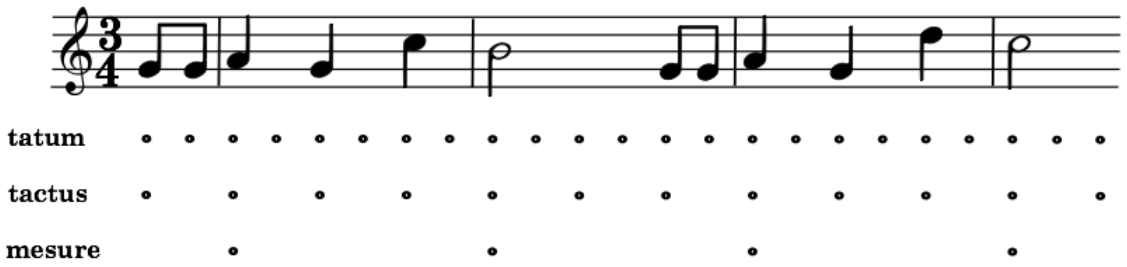

FIGURE 2.1 – Illustration du concept de niveau métrique. Ici, la phase du tatum, du tactus et de la mesure sont représentées par des points noirs sous un extraits de partition

Structure métrique

En parlant de "temps", on parle des temps de la mesure qui sont indiqués sur la partition ou pensés par le musicien dans le cas d'une improvisation. Il correspond en général à la battue de l'auditeur, mais pas tout le temps. Si ce n'est pas le cas, il sera alors souvent un multiple du "temps". Ces deux derniers sont extraits des écarts entre deux notes consécutives de la mesure, qui peut être bien plus petit ou plus long par exemple avec la présence d'une croche, ou d'une blanche. Le temps, le premier temps de chaque mesure, ou un le plus petit écart entre deux notes consécutives sont autant de métrique possible pour une musique. Par exemple, si on considère une phrase mélodique avec un accompagnement rythmique (une batterie par exemple) donnée, et que l'on considère ensuite la même phrase avec un tempo deux fois plus rapide ; l'auditeur percevra probablement une battue identique mais un tempo deux fois plus rapide pour le deuxième cas. La densité d'événements sonores change ainsi notre perception du tempo. Il existe plusieurs niveaux métrique pour un morceau de musique.

On définit le **tatum**, dérivé de "temporal atom", comme le plus petit niveau métrique. C'est le plus petit découpage régulier qui coïncide le mieux avec l'ensemble des événements sonores considérés. Les autres niveaux métriques seront donc des multiples de celui-ci.

Définissons les deux autres niveaux métriques courant que sont le **tactus** qui correspond a la pulsation du tempo perceptif, et la **mesure**, correspondant à la périodicité des mesures. Ces 3 métriques sont représentées sur la figure 2.1.2.

On ne s'intéressera dans ce stage qu'au tatum, les autres pouvant se déduire de celui-ci en prenant en compte nos a priori culturels. La méthode que nous développons base toute ces hypothèses sur celui-ci.

2.1.3 Principes généraux de l'analyse rythmique automatique

Les quatres grandes étapes de l'analyse rythmique dans le cas général

L'estimation de la position des temps et des premiers temps d'un morceau de musique à partir du signal audio s'effectue le plus souvent en quatre étapes consécutives. On détermine d'abord les accents rythmiques du morceau, c'est-à-dire les attaques des événements sonores. On

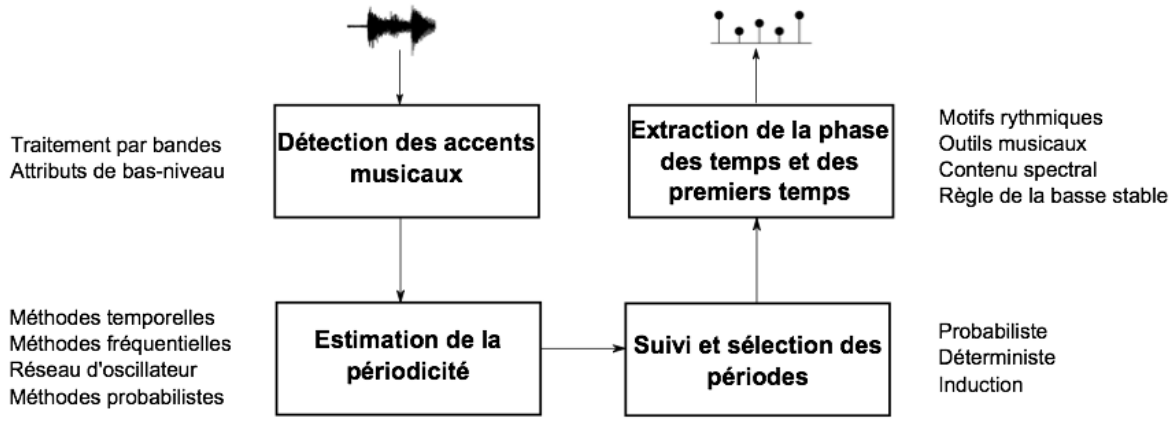

FIGURE 2.2 – Résumé d'une méthode générale d'estimation de la position des temps et des premiers temps

estime ensuite des périodicités correspondant à différents niveaux métriques, qui font ressortir les répétitions du morceau. On choisit ensuite celle-ci en fonction du passé pour faire un suivi dans le temps des différentes métriques. Enfin, on retrouve les différentes phases des périodicités (i.e leurs positions dans le temps). Ces étapes sont montrées sur la figure 2.1.3.

Dans notre cas, on ne considère pas l'extraction d'attaque du signal audio, mais on possède déjà en entrée la séquence temporelle d'onsets. Cela est possible en extrayant les "NoteOn" d'un formalisme MIDI par exemple.

Dans notre cas, on estime uniquement une seule périodicité : le tatum. Ainsi, on a s'affranchit de différencier les différentes périodicités présentes dans le morceau. De même, le suivi dans le temps et la reconstruction en phase seront considérés uniquement pour cette métrique qu'est le tatum.

Etat de l'art de l'estimation de la périodicité

C'est sur cette étape qu'a travaillé Gilbert Nouno dans son développement de la méthode décrite dans la section 2.2. Avant de décrire celle-ci, nous présentons ici les deux manières les plus courantes de faire dans la littérature. Le problème consiste à estimer la périodicité a partir des notes jouées. Une partition étant rarement monotone au niveau rythmique (on entend par là une succession de notes de même durée), le problème n'est pas trivial.

Méthodes temporelles : Celles-ci sont très utilisées pour leur simplicité et leur adéquation au problème. [Sep01] [Dix01] [GM99] [Sch98]. On peut en citer en particulier deux :

– Méthode utilisant la fonction d'auto-corrélation : On estime la corrélation r_x du signal x avec lui-même. Elle s'exprime par :

$$r_x(l) = \frac{1}{N-l} \sum_{n=O}^{N-1-l} x(n) * x(n+l)$$

avec N la longueur de la fenêtre d'observation en terme d'échantillon, et l la périodicité que l'on souhaite étudier en terme d'échantillon. La présence d'une métrique de période l dans x impliquera une valeur de $r_x(l)$ élevé.

– Méthode par banc de filtres résonnants [Sch98] puis [Kla03b]. On utilise ici un banc de filtres résonnants de la forme :

$$y_c(\tau, n) = \alpha_\tau * y_c(\tau, n - \tau) + (1 - \alpha_\tau) * x_c(n)$$

Avec n le numéro de la trame, τ le retard, x_c la fonction indicatrice des attaques et $\alpha_\tau = 0.5^{\frac{\tau}{T_0}}$ le gain de retour sur T_0 secondes. Plus T_0, plus on peut détecter de grande périodicité, mais la résonance des filtres augmentes.

Méthodes fréquencielles : On peut estimer la périodicité temporelle d'un signal à l'aide de méthode fréquentielles comme la Transformée de Fourier à temps discret ou DFT. A la manière d'une représentation en fréquence d'un signal, on peut, si on prend une fenêtre assez grande, regarder les fréquences qui ne sont pas dans un domaine audible et ainsi estimer la périodicité du signal. Il existe des dérivés de ce genre de méthode.

Méthodes fréquencielles et temporelles combinées Geoffrey Peeters [Pee06] a proposé une approche permettant de combiner les deux méthodes précédentes pour une estimation plus précise du tempo.

2.2 Principe originale de l'inférence de tempo sur une grille

Notre recherche de la périodicité se fait dans le cadre de certaines hypothèses restrictives qui ont déjà été sous entendues dans les parties précédentes, mais que nous rappelons ici. Cette section est issue de la thèse de Gilbert Nouno [Nou08], qui représente notre "état de l'art" puisque unique travail sur cette mesure.

– On considère comme entrée une suite d'événements temporels de n éléments $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq n}} \subset \mathbb{R}$ en seconde, qui représente la distribution des attaques de l'instrument au cours du temps. Cette considération implique déjà une notion de fenêtre temporelle d'observation. On observe les n derniers événements. La définition de la fenêtre comme posée ici implique une fenêtre avec un nombre d' éléments n fixe, plutôt qu'une taille temporelle de fenêtre constante. Cependant, il est facile de passer de l'un à l'autre on considérant les n événements sur une fenêtre d'observation donnée. On pourra ainsi se poser la question du choix optimal pour notre programme.
– Il n'y donc pas de considération de note, durée, vélocité (ou intensité) ou encore de timbre. On considère uniquement des "onsets".
– On sait que les observations sont bruitées par rapport à la "partition", c'est-à-dire que le jeu du musicien est imparfait par rapport à localisation temporelle d'une attaque dans le cas d'un jeu parfait. On le modélisera plus tard par une variable aléatoire.

On peut donc maintenant définir un modèle de grille ainsi qu'une mesure de distance entre un modèle de grille et notre distribution temporelle.

2.2.1 Principe de la mesure de Nouno

On définit alors un **Modèle** de Grille avec un paramètre de phase s et un pas (ou périodicité) Δ tel que :

$$\mathscr{G}(s,\Delta) = \{s + h * \Delta\}_{\substack{h\subset\mathbb{N} \\ 0\leq h\leq P}} \tag{2.1}$$

Et on définit alors une première mesure de distance Φ de la grille par rapport à la distribution telle que :

$$\Phi(s,\Delta) = \sum_{k=1}^{n} \min_{h=0,1,\dots,P} |t_k - (s + h * \Delta)| \tag{2.2}$$

Cette mesure de distance se calcule facilement par l'expression :

$$\begin{aligned}
\Phi(s,\Delta) &= \sum_{k=1}^{n} erreur_k(s,\Delta) && (2.3)\\
&= \sum_{k=1}^{n} \min\left\{ t_k - s - \left\lfloor \frac{t_k - s}{\Delta} \right\rfloor \cdot \Delta \ , \left(\left\lfloor \frac{t_k - s}{\Delta} \right\rfloor + 1 \right) \cdot \Delta - (t_k - s) \right\} && (2.4)\\
&= \sum_{k=1}^{n} \Delta \cdot \min\left\{ frac\left(\frac{t_k - s}{\Delta}\right), 1 - frac\left(\frac{t_k - s}{\Delta}\right) \right\} && (2.5)
\end{aligned}$$

Avec frac définit comme $frac(x) = x - \lfloor x \rfloor$

On peut se représenter cette mesure en observant la figure 2.3, qui montre ce à quoi correspondent chaque terme de la somme.

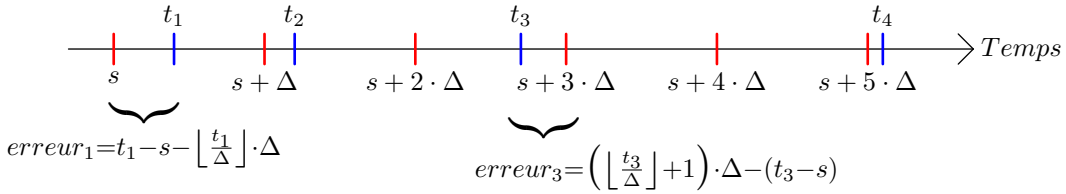

FIGURE 2.3 – Visualisation de la mesure Φ par le biais de chaque erreur qui la compose

Remarques :

- C'est une distance de Manhattan, ou distance de norme 1, le calcul n'est pas linéaire.
- On s'affranchit de connaitre le h relatif au Δ le plus proche de l'événement t_k, concrètement, on boucle sur les "k" et non sur les "h" dans l'équation 2.2. On peut remarquer que le nombre de points de la grille n'influe pas sur la mesure, et qu'a chaque événement est associé une erreur. C'est donc le nombre d'événements qui définit la mesure.
- Cette distance est homogène au paramètre Δ, exprimé en seconde. Elle dépend notamment

de la taille n de la série $\{t_j\}_{\substack{j\in\mathbb{N}\\1\leq i\leq n}} \subset \mathbb{R}$ par le biais de la somme sur k. Ainsi, la mesure dans son état actuel ne permet pas de comparer plusieurs séquences de taille n différentes entre elles. Remarquons aussi qu'elle diminue inévitablement lorsque Δ diminue, puisque directement proportionnelle à ce paramètre. Il ne présente donc a priori pas de minimum pour un pas $\Delta \in \mathbb{R}_*^+$

Pour palier ce problème, on peut remarquer que :

$$erreur_k(s,\Delta) \leq \frac{\Delta}{2} \tag{2.6}$$

Et donc par linéarité de la somme :

$$\Phi(s,\Delta) \leq n\frac{\Delta}{2} \tag{2.7}$$

On définit alors l'indice de coïncidence σ :

$$\begin{aligned} \sigma(s,\Delta) &= \Phi(s,\Delta)\cdot\frac{2}{n\Delta} \leq 1 && (2.8)\\ &= \frac{2}{n}\cdot\sum_{k=1}^{n}\min\left\{frac\left(\frac{t_k-s}{\Delta}\right), 1-frac\left(\frac{t_k-s}{\Delta}\right)\right\} \leq 1 && (2.9)\end{aligned}$$

Remarques :

- Cette fois la mesure ne dépend ni de la taille de la séquence d'entre, ni directement proportionnelle au paramètre Δ. Elle est utilisable pour comparer différentes mesures entre elles.
- On va maintenant pouvoir se servir de cette mesure pour définir notre concept de grille optimale.

2.2.2 Grille optimale et théorème fondamental

On définit alors la notion de grille optimale comme suit :

Définition 1. *Grille de coïncidence optimale*

Soit $\{t_j\}_{\substack{j\in\mathbb{N}\\1\leq i\leq n}} \subset \mathbb{R}$ une suite de n points telle que $0 \leq t_1 < t_2 < ... < t_n$.
Soit $\mathscr{G}(s,\Delta) = \{s + h*\Delta\}_{\substack{h\subset\mathbb{N}\\0\leq h\leq P}}$ une grille unidimensionnelle de P points en plus du centre, avec $s < \Delta$
On appelle grille optimale de $\{t_j\}_{\substack{j\in\mathbb{N}\\1\leq i\leq n}}$ les paramètres de la grille $(s*, \Delta*)$ si et seulement si ceux-ci minimisent la fonction d'indice de coïncidence σ tel que :

$$\sigma(s,\Delta) = \frac{2}{n}\cdot\sum_{k=1}^{n}\min\left\{frac\left(\frac{t_k-s}{\Delta}\right), 1-frac\left(\frac{t_k-s}{\Delta}\right)\right\}$$

Remarques :

– Dans sa version originale, Gilbert Nouno définit la grille optimale avec la fonction $\Phi(s, \Delta)$. Nous choisissons ici $\sigma(s, \Delta)$ pour les raisons évoquées ci-dessus.
– On est dans l'obligation de définir au préalable une taille maximale de grille P_{max}. Intuitivement, cela revient à définir un pas minimum de la grille.

À partir de cette définition, Gilbert Nouno étudie ses propriétés pour arriver à un théorème fondamental dans notre sujet qui s'exprime comme suit :

Théorème 1. *Discrétisation des solutions possibles de grille optimale*
Soit $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq n}} \subset \mathbb{R}$ une suite de n points telle que $0 \leq t_1 < t_2 < ... < t_n$ dont la grille optimale est définie par les paramètres $(s*, \Delta*)$
La grille optimale coïncide au moins avec deux points de la séquence d'entrée, c'est-à-dire qu'il existe h_1 et h_2 ainsi que k_1 et k_2 tel que :

$$\begin{cases} t_{k_1} = & s* + h_1\Delta* \\ t_{k_2} = & s* + h_2\Delta* \end{cases}$$

Conséquences et remarque :
– Ce théorème est toujours valable si on prend σ à la place de Φ
– On discrétise les valeurs possibles de Δ et de s (sous multiple des intervalles entre deux points) de la grille optimale. Ce théorème peut être regardé comme un résultat de finitude. En principe, le centre de la grille optimale ainsi que la valeur du pas Δ peut être choisi dans des intervalles continues qui appartiennent à $\mathbb{R}$, mais ce théorème assure que l'on ne perd pas en optimalité si le choix se restreint à des ensembles finis définis par les points d'entrée de la séquence. Une visualisation de ce procédé est disponible à la figure 2.2.2.
– On peut donc facilement trouver un algorithme qui explore toutes les possibilités tel que en parcourant k, et q tel que :

$$\sigma(q = f(s), \Delta) = \frac{2}{n} \cdot \sum_{k=1}^{n} \min \left\{ t_k - t_q - \left\lfloor \frac{t_k - t_q}{\Delta} \right\rfloor \cdot \Delta \ , \left(\left\lfloor \frac{t_k - t_q}{\Delta} \right\rfloor + 1 \right) \cdot \Delta - (t_k - t_q) \right\}$$

– Dans l'expression précédente, on remplace donc le centre s par un point de la grille t_q. On peut donc se retrouver a tester une quantité $t_k - t_q < 0$ si $k < q$, mais le calcul global de $\sigma(s, \Delta)$ n'est pas affecté par la présence de ce terme négatif.

– On repasse facilement de q à s par le calcul suivant : $s = \Delta \cdot frac\left(\frac{t_q}{\Delta}\right)$

– Ce théorème traduit le fait que pour qu'une métrique (tatum par exemple) nous soit accessible, il faut que le musicien joue au moins 2 notes sur la grille.

Gilbert Nouno définit alors un premier algorithme de recherche de grille optimale, c'est l'algorithme 1. Dans cet algorithme, le paramètre h qui représente l'ordre de la division de l'intervalle entre t_k et t_q est limité par le paramètre de la grille P_{max}. Cette limitation vient de la considération du cas $k = n$ et $q = 1$, qui représente la grille obtenue par une division du plus grand

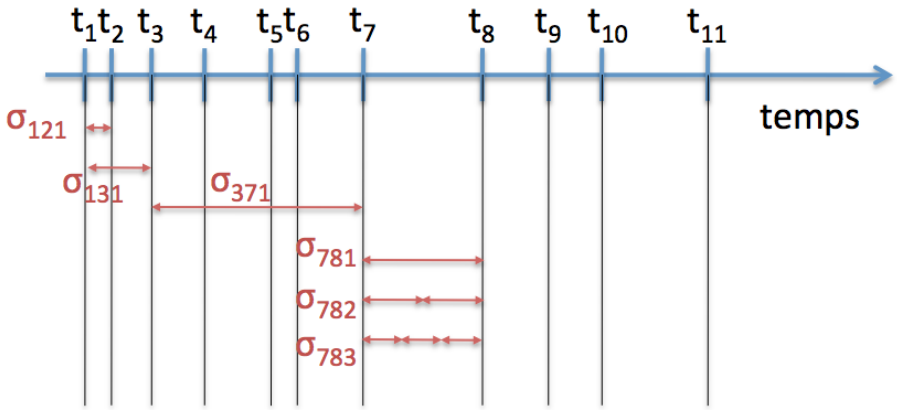

FIGURE 2.4 – Visualisation de l'exploration des multiples des sous intervalles. Le symbole σ_{qkh} représente l'intervalle associé à deux points t_k et t_q de la grille dont la distance a été divisé par h .

intervalle possible de la séquence d'entre, pour que celle-ci ait une taille maximum P_{max}. De ce fait, le paramètre P_{max} ne représente pas la taille maximum de la grille dans le cas général. Cependant, l'algorithme est toujours valide car cela revient à imposer un pas Δ de la grille minimum (et donc une taille P_{max} maximum de grille) qui serait défini par :

$$\Delta_{min} = \min_{k=1,2\cdots n-1} \frac{|t_{k+1} - t_k|}{P_{max}}$$

Algorithm 1 Algorithme original de recherche de grille optimale

Require:
$\{t_1, \ldots, t_n\}$;
P_{max}, nombre de points de grille en plus du centre.
for q = 1 to n **do**
  for k = q+ 1 to n **do**
    for h = 1 to P_{max} **do**

$$\Delta_{qkh} \leftarrow \frac{t_k - t_q}{h_n}$$

$$\sigma_{qkh} \leftarrow \frac{2}{n} \cdot \sum_{k=1} \min \left\{ frac\left(\frac{t_k - t_q}{\Delta_{qkh}}\right), 1 - frac\left(\frac{t_k - t_q}{\Delta_{qkh}}\right) \right\}$$

Avec frac définit comme $frac(x) = x - \lfloor x \rfloor$

    end for
  end for
end for
$(q*, k*, h*)$ tel que $\sigma_{q*,k*,h*} = \min_{q,k,h} \sigma_{qkh}$
$\Delta* \leftarrow \Delta_{q*,k*,h*}$
return q*, $\Delta*$

2.3 Premières modifications de l'algorithme pour l'adaptation en temps réel

Sont présentées dans cette section les modifications que l'on a apporté à l'algorithme de recherche de grille optimale.

2.3.1 Taille de grille et Intervalle de recherche

Taille de grille

On souhaite définir de manière un peu plus rigoureuse la taille maximale de la grille, pour mieux contrôler l'échelle de l'unique métrique que l'on recherche. Il est en fait plus intuitif d'imposer un paramètre de pas Δ minimum, qui ne dépend pas de la séquence d'entrée comme c'est le cas dans l'algorithme 1. A plus haut niveau encore, les musiciens et compositeurs ont l'habitude de travailler la notion de BPM (*Battements Par Minutes*). On propose donc de se servir de cette notion pour imposer notre pas minimum. On introduit ainsi non plus un paramètre P_{max} mais un paramètre BPM_{max} qui servira à limiter notre métrique. On peut alors facilement passer du BPM_{max} au Δ_{min} par la relation :

$$\Delta_{min} = \frac{60}{BPM_{max}} \tag{2.10}$$

Il suffit maintenant de tester dans l'algorithme 1 si la valeur Δ_{qkh} est inférieur a Δ_{min} pour sortir de la boucle sur h avant d'effectuer le calcul.

Pour être sûrs de bien parcourir tous les pas Δ possibles avec cette condition, on doit s'assurer que le paramètre P_{max} est assez grand. Pour cela, on se place dans le cas le plus défavorable possible et l'on s'assure que tous les deltas sont parcourus. C'est le cas pour $k = n$ et $q = 1$ et on impose donc logiquement P_{max} tel que :

$$P_{max} = \left\lfloor \frac{|t_n - t_1|}{\Delta_{min}} \right\rfloor + 1 = \left\lfloor \frac{|t_n - t_1| \cdot BPM_{max}}{60} \right\rfloor + 1 \tag{2.11}$$

Intervalle de recherche pour la périodicité

Maintenant que l'on a définit un pas minimum Δ_{min}, on peut se poser la question d'une limite supérieure pour le pas. Comme indiqué dans le section 2.1.3, on cherche uniquement la métrique du tatum. Limiter la recherche du pas à une borne supérieure revient à fixer les limites de la métrique que l'on cherche. De manière instinctive, on peut alors considérer que se limiter à la borne supérieure $\Delta_{max} = 2 \cdot \Delta_{min}$ est suffisant. Comme le montre l'exemple de la figure 2.1.2, si l'intervalle $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$ contient la période du tactus, et que la note de durée la plus courte est une croche, alors l'intervalle $[2 \cdot \Delta_{min} \quad 4 \cdot \Delta_{min}]$ contiendra le tactus, ou "battue". En revanche, si la note la plus courte est une double croche, il faudra monter à l'intervalle $[4 \cdot \Delta_{min} \quad 8 \cdot \Delta_{min}]$ pour retrouver la "battue", et le tatum se trouvera plutôt dans l'intervalle $\Delta_{min}/2 \quad \cdot \Delta_{min}]$. Si on considère maintenant un morceau dans sa totalité, il est probable que l'on trouve des passages

a la croche, et d'autres à la double croche. Le tatum varie donc dans un intervalle beaucoup plus grand que $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$.

En fait, on peut montrer selon certaines hypothèses qu'il y a de bonnes chances pour que la mesure σ soit minimum pour la métrique du tatum.

On peut montrer facilement à partir de l'équation 2.8 que considérant un terme

$$erreur'_k(s, \Delta) = \frac{erreur_k(s, \Delta)}{\Delta} = \min \left\{ \frac{t_k - s}{\Delta} - \left\lfloor \frac{t_k - s}{\Delta} \right\rfloor, 1 - \frac{t_k - s}{\Delta} + \left\lfloor \frac{t_k - s}{\Delta} \right\rfloor \right\} \in [0 \quad {}^1/_2]$$

issue d'une mesure $\sigma(s, \Delta)$, que l'on a :

$$erreur'_k(s, \frac{\Delta}{2}) \in [0 \quad {}^1/_2] = \begin{cases} 2 \cdot erreur'_k(s, \Delta) & \text{si } erreur'_k \leq {}^1/_4 \\ 1 - 2 \cdot erreur'_k(s, \Delta) & \text{si } erreur'_k \geq {}^1/_4 \end{cases} \tag{2.12}$$

On prend ensuite l'hypothèse d'un tempo constant, (les notions d'accélération et courbe de tempo seront abordés dans la section 2.4), et d'un musicien parfait. Un tel musicien produira alors toutes ses notes notées $\left\{t_j^p\right\}_{\substack{j \in \mathbb{N} \\ 1 \leq i \leq n}} \subset \mathbb{R}$ sur la grille du tatum. Donc, il existe pour tout j, h_j, s^p* et Δ^p* tel que $t_j^p = s^p * + h_j \cdot \Delta^p*$. On peut en déduire que pour un tel musicien, $\sigma^p(s^p*, \Delta^p*) = 0$.

On considère maintenant un musicien imparfait qui tente de reproduire le musicien parfait dont on peut suivre la "battue". On prend pour hypothèse qu'il produit une série de note $\left\{t_j^{ip}\right\}_{\substack{j \in \mathbb{N} \\ 1 \leq i \leq n}} \subset \mathbb{R}$ tel que $t_j^{ip} = t_j^p + T_j$, avec T une variable aléatoire telle que $T \sim \mathcal{N}(0, \sigma_T)$. Chaque $erreur_k(s^{ip}*, \Delta^{ip}*)$ suivra donc une loi demi-normale de paramètre σ_T, dont la variance s'exprime par : $Var_{|T|} = \sigma_T^2 \left(1 - \frac{2}{\pi}\right)$. En supposant que le musicien a une variance $\sigma_T \ll \frac{\Delta^p*}{2}$, puisque l'on peut malgré son imperfection toujours détecter une "battue" , on aura $Var_{|T|} \ll \frac{{\Delta^p*}^2}{4}$ et donc on sera très probablement dans le cas $erreur'_k \leq {}^1/_4$. On en déduit que pour une périodicité plus petite que le tatum, la mesure renvoie une mesure en moyenne deux fois plus grande que celle du tatum. Le cas où la périodicité est plus grande que celle du tatum renverra elle-aussi une valeur de la mesure plus grande, puisque des points t_j ne seront pas positionné sur la grille.

La mesure devrait donc, sous ces hypothèses, pouvoir déceler le tactus pour peu que l'on prenne un Δ_{min} assez petit. Cependant, on gardera quand même l'intervalle de recherche $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$, pour plusieurs raisons :

- Parce qu'un morceau comporte souvent des passages dont la plus petite durée de note varie beaucoup, on peut potentiellement être en présence de beaucoup de "saut"' de périodicité sur le tatum. Cela rend une exploitation par l'ordinateur plus complexe.
- Tester un plus grand intervalle prend évidemment plus de temps de calcul.
- On constate de manière empirique que les motifs formés sur l'espace (Δ, σ) se répètent à un facteur multiplicatif près d'octave en octave sur un espace logarithmique binaire (ou l'intervalle $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$ est équivalent à l'intervalle $[2 \cdot \Delta_{min} \quad 4 \cdot \Delta_{min}]$ en terme de distance.)

– Il est de toute façon très facile de repasser à un intervalle de type $[\Delta_{min} \quad x \cdot \Delta_{min}]$, avec x une puissance de 2 dans la programmation de l'algorithme final.

2.3.2 Considérations sur les mesures et taille et de la taille de la séquence d'entrée

L'algorithme 1 à partir de la séquence d'entrée $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq j\leq n}}$ retourne le couple $(s*, \Delta*)$ optimale. Puisque notre objectif est à terme de faire un suivi en temps réel de la métrique du tatum, on cherche à suivre ce couple dans le temps. Aussi, il sera plus intéressant pour notre traitement statistique de récupérer non pas le couple optimal mais plutôt un ensemble de couple $\{s_j, \Delta_j\}_{\substack{j\in\mathbb{N} \\ 1\leq j\leq N_{obs}}}$ auquel on associera une valeur de la mesure $\{\sigma_j\}_{\substack{j\in\mathbb{N} \\ 1\leq j\leq N_{obs}}}$, avec N_{obs} le nombre de mesure que l'on retiendra. La mesure σ_j donnera ainsi une indication de la probabilité que ce couple soit le bon. On pourra ensuite appliquer un traitement statistique d'apprentissage automatique qui peut être vu comme un filtrage. Cela implique que les données soit comparables dans le temps. Or, si deux mesures de périodicité Δ_j sont comparables, comparer deux mesures d'une phase s_j n'a de sens que si les périodicités sont identiques. Si elles ne le sont pas, on est dans l'obligation de définir un point de référence pour la comparaison, et ainsi induire un paramètre arbitraire (qui va en fait dépendre des caractéristiques des fenêtres d'observations temporelles que l'on choisi). Aussi, si il n'y a pas de précaution particulière à prendre pour comparer les Δ_j entre eux avec un filtrage, il faudra faire des choix pour comparer les phases.

Dans son état actuel, l'algorithme 1, sous couvert d'une légère modification, peut retourner un ensemble de couple $\{s_j, \Delta_j\}$ qui peuvent différer deux à deux uniquement par leur phase s_j avec un grande variation de la mesure. Par exemple, dans le cas idéal d'un couple $\{s*, \Delta*\}$ représentant l'exact tatum d'un musicien parfait, on aura $\sigma(s*, \Delta*) = 0$. On peut en déduire que $\sigma(s* +^{\Delta*}/_2, \Delta*) = 1$ et d'une manière plus générale que $\sigma(s, \Delta*) \in [0 \quad 1], \forall s \in [0 \quad \Delta*]$.

Conserver de tels couples s'avère peu intéressant puisque cela peut introduire un biais sur la valeur recherchée de $\Delta*$, d'autant plus que faire un suivi en phase n'a finalement que très peu de sens. Enfin, notre suivi de périodicité pourra nous donner au mieux une valeur estimée $\hat{\Delta*}$ de $\Delta*$ qui va nous obliger à reconstruire la meilleure phase possible en fonction de cette valeur estimée. En fait, la mesure de la phase peut être vue comme un témoin de l'incomplétude de la mesure de $\Delta*$. Rien ne nous permet d'affirmer que la phase s_j d'un couple $\{s_j, \Delta_j\}$ est la meilleure. En revanche, on sait d'après le théorème fondamental que la mesure optimale se trouve parmi les couples testés. On extrapole et constate de manière empirique que pour un Δ donné, il y a de grandes chances que l'on test sa phase optimale associée. Un exemple de ce qu'on entend par "l'incomplétude des données" est visible sur la figure 3.1 avec des précisions dans la section 3.1.3.

En conséquence, on choisit de modifier l'algorithme 1 pour forcer les couples $\{s_j, \Delta_j\}$ à ne présenter que des valeurs de Δ_j différentes entre elles. Cette modification est facile à faire en rajoutant dans l'algorithme une étape de test si le Δ courant a déjà été testé. Si c'est le cas, on teste alors la phase associée ainsi que la mesure courante et on remplace la valeur de la phase si celle-ci présente une meilleure mesure. Ainsi, on s'assure que toutes les périodicités

issues de la mesure seront uniques. Cela nécessite d'introduire un nouveau paramètre ϵ_Δ qui représente la valeur à partir de laquelle on considère deux Δ_j identiques. Cette valeur est choisie en fonction de la littérature psycho-acoustique, en exploitant les valeurs des JNDs (*Justiciable Noticiable Difference*) d'un percussionniste professionnel sur la reconnaissance d'un rythme, soit environ 1ms. On prendra alors la valeur 0.01ms pour palier le fait que ces tests sont effectués sur la métrique de la battue et non du tatum (beaucoup plus petit en général sur le jeu d'une percussion), ainsi que les fenêtres d'observations contiennent un certain nombre n d'événements impliquant un report de l'erreur sur un temps donné. On peut ainsi choisir la valeur similaire de ϵ_s (tolérance sur la phase) comme approximativement $\epsilon_s = {}^{\epsilon_\Delta}/_{P_{max}}$ pour palier le report d'erreur d'une grille contenant au maximum P_{max} points.

Le paramètre N_{obs} va dépendre du nombre de mesures disponibles dépendant lui-même de la taille de la séquence d'entrée n. Pour avoir un maximum de chance d'éviter les mesures incomplètes (c'est a dire avec une phase s non optimale pour un Δ associé), il faut que N_{obs} soit très inférieur au nombre de mesure. Le nombre de mesure dans l'intervalle de recherche n'est pas calculable directement car il dépend de la séquence d'entre et du Δ_{min}. Empiriquement, on constate qu'avec environ $n \approx 20$ événements et un $BPM_{max} = 600$, on obtient en général un nombre de mesure d'ordre de grandeur de 500 qui nous assure un nombre d'observations $N_{obs} = 100$ que l'on considère arbitrairement comme convenable.

2.3.3 Nouvel algorithme

On présente ici le nouvel algorithme 2 qui prend en compte nos remarques précédentes, basé sur l'algorithme 1.

Précision sur la notation :

- $\Delta = NaN(n, n, p)$ représente l' initialisation d'une matrice de dimension (n,n,p) contenant des cellules vides (ou assimilés)
- Les tests de type $|var_{temp} - var(q', k', h')| \leq \epsilon$ représente d'égalité avec une tolérance de ϵ entre le scalaire var_{temp} et la matrice de données var. Si le test est positif, le scalaire de la matrice de données qui correspond à var_{temp} est indexé par (q', k', h'). Ce scalaire est forcement unique vu la construction de l'algorithme.
- $\{\mathbf{q,k,h}\}_{N_{obs}}$ représente une série de N_{obs} triplets de type (q, k, h).
- sort $\left(\min_{q,k,h}\{\sigma(q, k, h)\}, N_{obs}\right)$ représente les N_{obs} premiers triplets (q, k, h) classés selon le critère $\sigma(q, k, h)$ de manière ascendante.
- $var(\{\mathbf{q,k,h}\}_{N_{obs}})$ représente les N_{obs} images de la série de triplets de type (q, k, h) par la matrice var.

Algorithm 2 Nouvel algorithme de recherche de triplets candidats pour la grille optimale

Require: $\{t_1, \ldots, t_n\}$; BPM_{max} ; N_{obs} ; ϵ_Δ

$\Delta = NaN(n,n,p)$; $\sigma = NaN(n,n,p)$; $s = NaN(n,n,p)$

$\Delta_{min} = 60/BPM_{max}$

$P_{max} = \left\lfloor \frac{|t_n - t_1|}{\Delta_{min}} \right\rfloor + 1$

$\epsilon_s = {}^{\epsilon_\Delta}/_{P_{max}}$

for q = 1 to n **do**

  for k = q+ 1 to n **do**

    for h = 1 to P_{max} **do**

      $\Delta_{temp} \leftarrow \frac{t_k - t_q}{h}$

      $s_{temp} \leftarrow frac\left(\frac{t_k}{\Delta_{temp}}\right)$

      if $\Delta_{temp} < \Delta_{min}$ **then**

        break

      else if $\Delta_{temp} > 2 \cdot \Delta_{min}$ **then**

        continue

      else if $|\Delta_{temp} - \Delta(q',k',h')| \leq \epsilon_\Delta$ **then**

        if $|s_{temp} - s(q',k',h')| \leq \epsilon_s$ **then**

          break

        else

          $\sigma_{temp} \leftarrow \sigma(s_{temp}, \Delta_{temp})$

          if $\sigma_{temp} < \sigma(q',k',h')$ **then**

            $\sigma(q',k',h') \leftarrow NaN$; $\Delta(q',k',h') \leftarrow NaN$; $s(q',k',h') \leftarrow NaN$

            $\sigma(q,k,h) \leftarrow \sigma_{temp}$; $\Delta(q,k,h) \leftarrow \Delta_{temp}$; $s(q,k,h) \leftarrow s_{temp}$

          end if

        end if

      else

        $\sigma(q,k,h) \leftarrow \sigma_{temp}$; $\Delta(q,k,h) \leftarrow \Delta_{temp}$; $s(q,k,h) \leftarrow s_{temp}$

      end if

    end for

  end for

end for

$\{\mathbf{q^*,k^*,h^*}\}_{N_{obs}}$ tel que $\sigma(\{\mathbf{q^*,k^*,h^*}\}_{N_{obs}}) = \text{sort}\left(\min_{q,k,h}\{\sigma(q,k,h)\}, N_{obs}\right)$

return $s(\{\mathbf{q^*,k^*,h^*}\}_{N_{obs}})$, $\Delta(\{\mathbf{q^*,k^*,h^*}\}_{N_{obs}})$, $\sigma(\{\mathbf{q^*,k^*,h^*}\}_{N_{obs}})$

2.4 Considération sur l'accélération

Dans sa thèse, Gilbert Nouno étend sa méthode de recherche de grille optimale à des séquences uniformément accélérées. L'idée sous-jacente est de pouvoir approximer la "courbe de tempo" (définie dans la section 2.4.1) par des segments de droite. Cette méthode n'a finalement pas eu son utilité dans la méthode de suivi de tempo que nous avons mis en place dans ce stage car elle

est peu stable et n'apporte pas suffisamment pour être retenue en premier lieu. Cependant, elle a été étudiée et modifiée dans la première partie du stage et des vestiges de son implémentation sont présents dans les algorithmes de recherche de grille optimale du code. On présente ici les travaux qui ont été faits dans ce cadre bien que non retenus dans la version finale de ce stage de l'algorithme. Les pistes, remarques et méthodes présentées ici pourraient cependant servir à des travaux ultérieurs.

2.4.1 Notion et formalisme de Courbe de Tempo

L'algorithme présenté dans la section 2.3.3 suppose un tempo constant : le pas Δ ne varie pas dans le temps. Un tel tempo n'est cependant jamais possible dans la réalité d'une performance live par un être humain. C'est en revanche possible pour un séquenceur logiciel mais auquel cas, le tempo est un paramètre d'entrée qu'il n'est plus nécessaire de retrouver. Dans le cas d'une performance musicale, on peut supposer et donc modéliser la variation du tempo du musicien par une variation continue, dans la mesure où le musicien cherche à suivre le modèle parfait du tempo constant. En vérité, cela ne peut être qu'une modélisation mathématique puisque le tempo est par essence discret, manifesté par la nature discrète du rythme musical qui le compose. Pour cette modélisation, on considère que les événements "échantillonnent" la courbe continue du tempo et constituent l'information à laquelle nous avons accès.

Les travaux de Gilbert Nouno s'appuient sur ceux de Guerino Mazzola ([Maz94]), qui considère deux métriques de tempo positionnées dans deux espaces bien distincts. Le premier est l'espace de la partition (bien qu'à l'origine la méthode se veut appliquée aux musiques improvisés, on peut toujours considérer l'existence d'une partition implicite dans l'esprit du musicien), qui possède son propre temps abstrait que l'on pourrait exprimer en unités de pulsation et que l'on note d. On appellera maintenant cet espace, *l'espace pulsationnel*. Le second est celui de l'interprétation réelle de la partition, ou encore sa réalisation. Cet espace est muni du temps que nous connaissons, que l'on pourrait exprimer en secondes et que l'on note t. Nous l'appellerons maintenant *l'espace temporel*. Un représentation de ces espaces est visible sur la figure 2.4.1.

Mazzola et Nouno se basent sur une fonction f appelée *fonction de performance* qui permet de passer d'un espace à un autre par la relation $t = f(d)$. L'hypothèse est ensuite de définir la dérivée de cette fonction par rapport à la pulsation d comme la dérivée de la variable t, soit $t' = \dfrac{\partial}{\partial d} f(d)$. Avec cette hypothèse, Mazzola arrive à obtenir une formule sur laquelle est basée tout le développement de Gilbert Nouno qui s'exprime ainsi :

$$t = f(d) = \int_{d_0}^{d} \frac{1}{T(d)} \partial d \qquad \text{avec} \qquad \frac{\partial}{\partial d} f(d) = \frac{1}{T(d)} \tag{2.13}$$

Malheureusement, bien que que ces expressions n'ont pour la plupart aucun sens d'un point de vue mathématique (on peut cependant leur en donner un du point de vue de la physique, avec les abus de notations habituelles) les démonstrations données par Nouno et Mazzola se trouvent être peu convaincantes lorsqu'elles sont présentes.

Avec l'aide d'Helianthe Caure, doctorante mathématicienne de l'IRCAM, nous avons donc

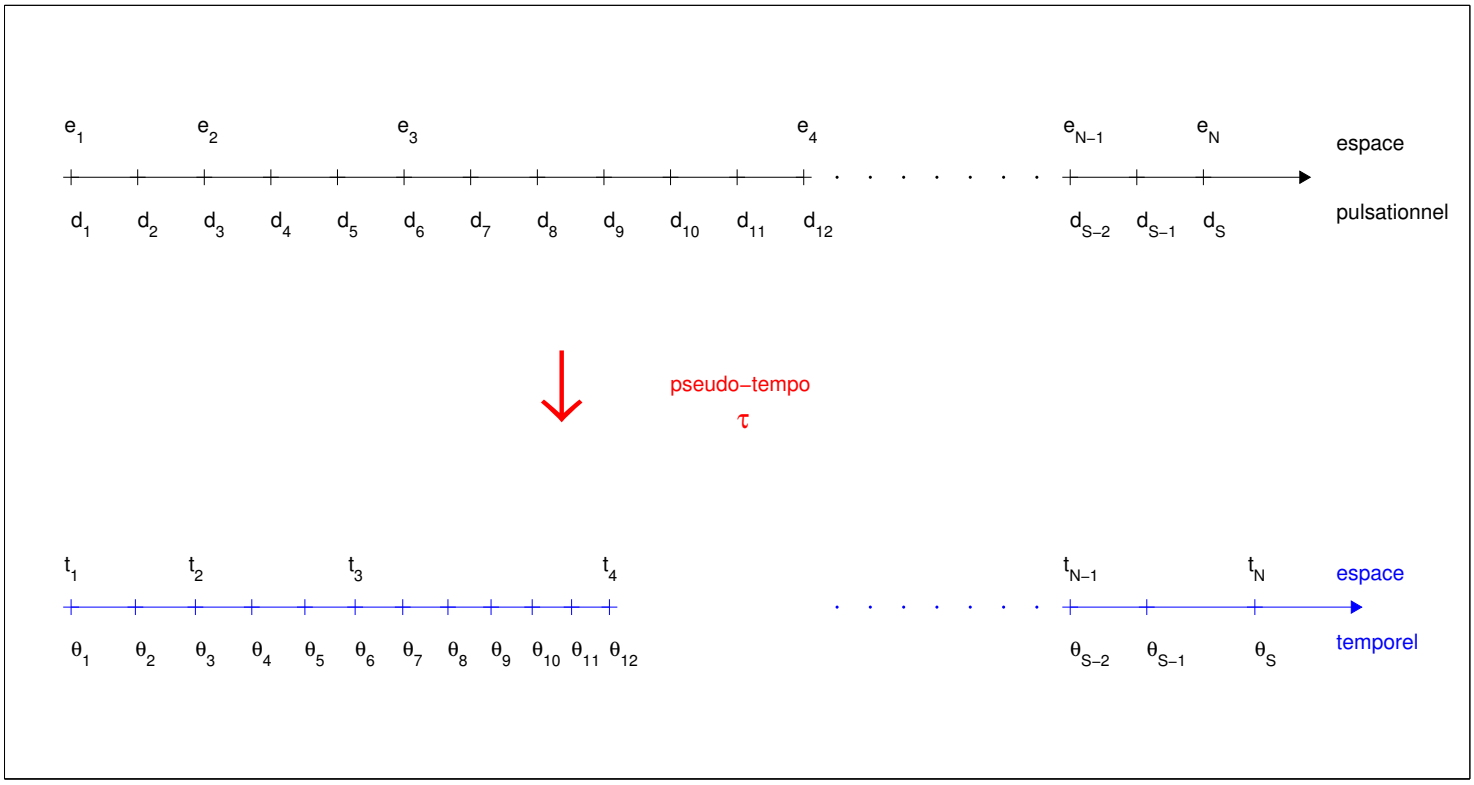

FIGURE 2.5 – Visualisation de l'espace pulsationnel et de l'espace temporel re-lié entre eux par une fonction de pseudo-tempo (voir annexe A)

entrepris de donner une démonstration fiable et convaincante d'un point de vue mathématique et pour ainsi mieux définir ce qu'on entend par cette équation. Les détails sont disponibles en Annexe A).

A l'arrivée, on obtient effectivement l'équation (qui est une des solutions possibles au problème) :

$$t_j - t_1 \quad \propto \quad \int_{d_1}^{d_{\sigma(j)}} \frac{1}{T_{bis}(d_{bis})} \mathrm{d}d_{bis} \tag{2.14}$$

Avec σ une fonction d'index reliant le tatum et les événements t_j, d_{bis} la variable de l'espace pulsationnel continue (il est plus naturel de le définir comme un espace discret en premier lieu), et T_{bis} la fonction de tempo associée à cet espace.

La démonstration fait clairement apparaître l'essence discrète du tempo, accessible uniquement par le rythme musical, et montre que son prolongement sur un espace continu peut se faire d'une infinité de façon. Il est important de préciser que cette équation résulte donc d'un choix non trivial de Mazzola et Nouno, et ne découle pas "naturellement" du modèle posé. La démonstration montre aussi l'importance d'une métrique idéale dont la périodicité est suffisamment faible pour

contenir l'ensemble des événements. Cela avait été par ailleurs montré d'un point de vue cognitif dans [PE85]

2.4.2 Variation linaire du Tempo

Variation linéaire du tempo, modélisation proposé par Nouno

Nouno développe dans sa thèse le cas d'une séquence uniformément accélérée (ou décélérée selon le signe du paramètre). Il commence donc pas poser :

$$T(t) = T_0 + \epsilon \cdot t \cdot T_0 \tag{2.15}$$

Avec ϵ une valeur de dimension $[Temps]^{-1}$, et T_0 une constante de dimension $[Pulsation][Temps]^{-1}$. Cette valeur étant un peu contre-intuitive, elle définit aussi θ tel que :

$$T(t_n) = \theta \cdot T(t_1) \tag{2.16}$$

On peut passer de l'un à l'autre par la simple formule :

$$\epsilon = \frac{\theta - 1}{t_n - t_1} \tag{2.17}$$

Partant de 2.14, il vient alors les formules de passage permettant de passer d'une séquence à sa version linéairement accélérée résumée ici :

Accélération uniforme directe	Accélération uniforme réciproque
$t^{(\epsilon)} = \frac{ln(1+t\epsilon)}{\epsilon}$	$t = \frac{e^{\epsilon \cdot t^{(\epsilon)}} - 1}{\epsilon}$
$t^{(\theta)} = t_1 + \frac{t_n - t_1}{\theta - 1} ln\left(1 + \frac{t - t_1}{t_n - t_1}(\theta - 1)\right)$	$t = t_1 + \frac{t_n^{\theta} - t_1}{ln(\theta)} \left(\theta^{\frac{t^{(\theta)} - t_1}{t_n^{(\theta)} - t_1}} - 1\right)$

Pour plus de visibilité, on pose alors la séquence réduite :

$$\{s_k\}_{\substack{k \in \mathbb{N} \\ 1 \leq i \leq n}} = \frac{t_k - t_1}{t_n - t_1} \subseteq [0, 1] \tag{2.18}$$

et les formules deviennent alors :

Accélération uniforme directe	Accélération uniforme réciproque
$s^{(\theta)} = \frac{ln\left(1 + (\theta - 1) \cdot s\right)}{\theta - 1}$	$s = \frac{\theta^{s^{(\theta)}} - 1}{ln(\theta)}$

Une vision de comment ces formules agissent sur une séquence est visible sur la figure 2.6.

Commentaires :

– On cherche maintenant un triplet (Δ, s, θ) qui minimise la distance de la séquence à la Grille.

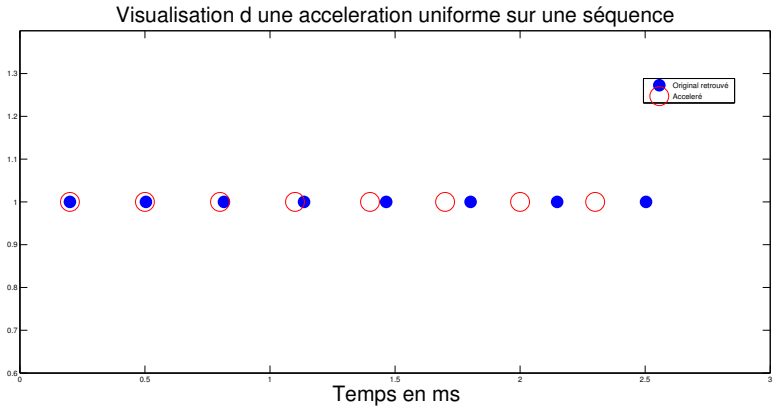

FIGURE 2.6 – Visualisation d'une accélération uniforme sur une séquence

- Minimisation en plusieurs étapes due aux 3 dimensions, c'est un schéma itératif alterné. On minimise d'abord (s, Δ) puis θ. On est pas certain de la convergence de ce schéma, et on constatera d'ailleurs de manière empirique qu'il n'est pas très stable pour des séquences un peu complexes.
- On retro-accélère la séquence pour garder l'intérêt de la partie fractionnaire.
- L'algorithme proposé par Nouno est alors l'algorithme 3.

Algorithm 3 Minimisation de $\sigma(s, \Delta, \theta)$ par méthode itérative altérnée

Require: $\{t_1, \ldots, t_n\}$; BPM_{max} ; ϵ

Initialisation $(\bar{s}, \bar{\Delta}, \bar{\theta}) \leftarrow (Nan, 1, NaN)$

while $|\sigma(\bar{s}, \bar{\Delta}, \bar{\theta}) - \sigma(\hat{s}, \hat{\Delta}, \hat{\theta})| <= \epsilon$ **do**

$(\hat{s}, \hat{\Delta}, \hat{\theta}) \leftarrow (\bar{s}, \bar{\Delta}, \bar{\theta})$

$\sigma(\bar{s}, \bar{\Delta}, \hat{\theta}) \leftarrow \min_{(s,\Delta)} \sigma(s, \Delta, \hat{\theta})$ *Avec algorithme 1*

$\sigma(\bar{s}, \bar{\Delta}, \hat{\theta}) \leftarrow \min_{(\theta)} \sigma(\bar{s}, \bar{\Delta}, \theta)$

end while

return $(\hat{s}, \hat{\Delta}, \hat{\theta})$

Pour minimiser $\sigma(\bar{s}, \bar{\Delta}, \theta)$ par rapport à θ, Nouno propose de résoudre l'équation 2.19 par la méthode de Newton et de sélectionner ensuite le θ_{hk} qui minimise la mesure σ.

$$\frac{\theta_{hk}^{s_k} - 1}{ln(\theta_{hk})} = s + h \cdot \Delta \qquad \forall h, \forall k \tag{2.19}$$

On constate lors de nos tests que ce schéma itératif alterné met plus de 300 itérations pour converger vers un triplet $(\hat{s}, \hat{\Delta}, \hat{\theta})$ et que sa convergence n'est pas assurée si la séquence est un peu complexe. Cela ne le rend pas exploitable pour du temps réel.

Remarques et modifications dans la recherche du facteur d'accélération

Remarque sur l'utilisation de θ plutôt que ϵ

L'utilisation de θ plutôt que ϵ peut sembler plus intuitif mais s'avère moins judicieux pour

faire un calcul en temps réel. En effet, la première remarque que l'on peut faire est que θ dépend finalement de la taille de la séquence considérée. Ainsi, si on trouve un facteur d'accélération θ optimale pour une séquence d'entrée, l'accélération que l'on doit appliquer sur la grille pour qu'elle suive la séquence supposée accélérée d'un facteur θ est alors caractérisé par :

$$\theta_{corr} = 1 + \frac{P \cdot \Delta}{t_n - t_1}(\theta - 1) \tag{2.20}$$

Avec P la taille de la grille. Cette équation est issue de l' équation 2.17. Cela est dû au fait que la quantité $t_n - t_1$ ne coïncide pas forcement avec $P \cdot \Delta$. En fait, les θ ne sont pas directement comparables pour des séquences de tailles différentes et si on veut réaliser un suivi de l'accélération, on doit repasser par la quantité ϵ.

Remarques sur l'équation 2.19 et l'algorithme 3

La première remarque que l'on peut faire est que la résolution de l'équation 2.19 pour tous les couples (h, k) est superflue. En effet, elle correspond a trouver un ensemble de facteurs θ_{hk} qui tente de faire correspondre à un point t_k le point correspondant $s + h \cdot \Delta$. L'ensemble des θ_{hk} balaye un très large intervalle très peu représentatif des θ possibles (par exemple quand on essaie de faire correspondre le dernier point de la grille au premier point de la séquence). De plus, il y a très peu de valeurs autour de 1 qui représentent pourtant la valeur la plus probable. Souvent, il n'y environ que n mesures exploitables. Il est plus judicieux de limiter l'algorithme à un intervalle $[\theta_{min} \quad \theta_{max}]$ avant de lancer la résolution par la méthode Newton. Par exemple dans la pratique, il est très peu probable que le musicien accélère de plus d'un facteur 2 sur une séquence longue de 10 secondes.

La seconde remarque est que le grand nombre d'itérations vient du fait que la grille que l'on utilise dans l' équation 2.19 est optimisée pour la séquence alors encore non rétro-accélérée. Il est plus judicieux de prendre une grille dont le pas Δ' serait la moyenne de la grille précédente si elle avait été elle aussi rétro-accélérée. On propose cette méthode dans l'équation suivante :

$$\frac{\theta^{s_k} - 1}{ln(\theta)} = s + h \cdot \Delta' \qquad \forall \theta \neq 1 \tag{2.21}$$

avec

$$\Delta' = \frac{1}{P} \sum_{i=1}^{P} \theta^{s+i\Delta} \left(\frac{\theta^{\Delta} - 1}{ln(\theta)} \right) = \frac{\theta^{\Delta+s}}{P} \frac{\theta^{P \cdot \Delta} - 1}{ln(\theta)} \tag{2.22}$$

Ce qui revient finalement a résoudre l'équation.

$$\theta^{s_k} + \frac{h}{P}\theta^{\Delta+s} - \frac{h}{P}\theta^{\Delta \cdot (P+1)+s} - s \cdot ln(\theta) - 1 = 0 \tag{2.23}$$

L'utilisation de cette équation rend l'algorithme beaucoup plus rapide mais il souffre du même problème que lorsqu'on utilise l'équation 2.19 pour des séquences complexes : l'algorithme ne converge pas toujours vers une valeur de θ cohérente.

Pour ces raisons, on propose d'utiliser un simple balayage logarithmique pour la recherche

du θ optimale.

Balayage logarithmique

On souhaite utiliser un balayage logarithmique pour être en accord avec la fraction de Weber et notre perception du rythme en octave de tempo (on ressent aussi bien avec une même intensité un tempo qui double qu'un tempo qui diminue d'un facteur deux). En choisissant de manière raisonnable les bornes θ_{min} et θ_{max}, l'algorithme nécessite moins de temps de calcul que les tentatives pour converger vers une valeur optimale de θ. La distribution des facteurs θ est alors exprimée par :

$$\theta_k = \theta_{min} \cdot 2^{\frac{k}{M} \cdot \log_2\left(\frac{\theta_{max}}{\theta_{min}}\right)} \qquad \forall k = 1 \cdots M \tag{2.24}$$

On propose ainsi l'algorithme 4, dans lequel la notation $accel\_reciproque(\{t_1, \ldots, t_n\}, \theta i)$ représente l'application de l'application réciproque d'un facteur θ_i à la séquence $\{t_1, \ldots, t_n\}$. On obtient le genre de résultat de la figure 2.7. Il est facile de l'étendre, à l'image de l'algorithme 2, pour obtenir un ensemble de triplets candidats plutôt qu'un triplet unique optimal. Cette méthode est présente dans les codes issus de ce stage, mais n'a pas été étudiée dans le cas d'un traitement statistique par manque de temps d'abord et ensuite parce qu'elle n'a pas été vraiment nécessaire pour suivre la variation d'un tempo.

Algorithm 4 Minimisation de $\sigma(s, \Delta, \theta)$ par méthode de balayage

Require: $\{t_1, \ldots, t_n\}$; BPM_{max} ; ϵ ; M ; θ_{min} ; θ_{max}

for i=1 to M **do**

$\theta_i \leftarrow 2^{\log_2(\theta_{min}) + \frac{i}{M} \cdot \log_2\left(\frac{\theta_{max}}{\theta_{min}}\right)}$

$\{s_1, \ldots, s_n\} \leftarrow accel\_reciproque(\{t_1, \ldots, t_n\}, \theta i)$

$(s_i, \Delta_i) \leftarrow \min_{(s,\Delta)} \{\sigma(s, \Delta, \{s_1, \ldots, s_n\})\}$

end for

$(s*, \Delta*, \theta*) \leftarrow (s, \Delta, \theta)$ tel que $\min_i \{\sigma(s_i, \Delta_i, \theta_i)\}$

return $(s*, \Delta*, \theta*)$

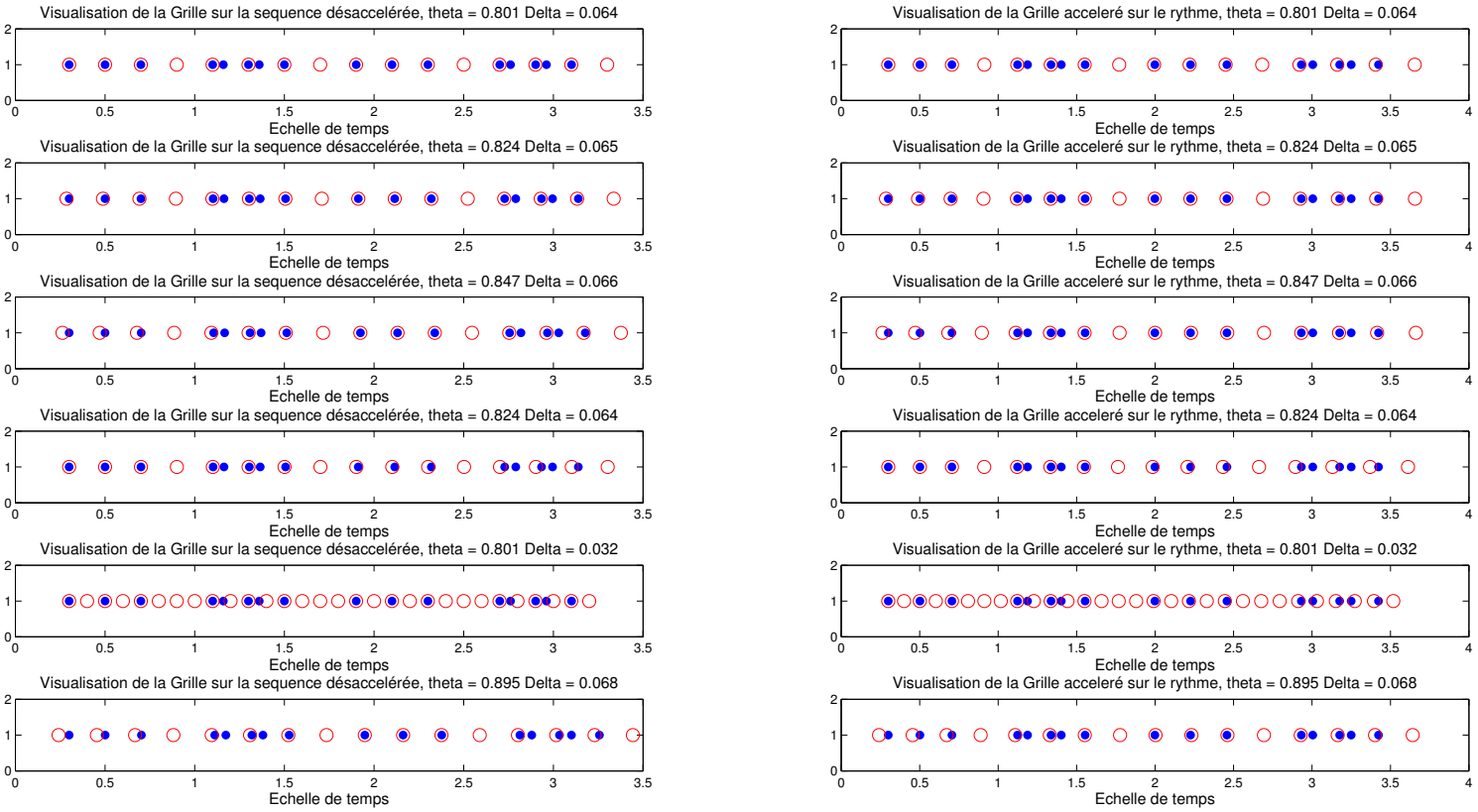

FIGURE 2.7 – Exemple de résultat des meilleurs triplets-candidats (le centre s n'est pas mentionné cependant) obtenus sur MATLAB. Les meilleurs sont en haut et les moins bons sont en bas. Le rythme est dés-accélérée à gauche, tandis que la grille est accélérée à droite. Le rythme est en bleu, la grille est en rouge.

Chapitre 3

Suivi de tatum en temps réel

3.1 Généralité

On présente dans cette section les méthodes et algorithmes associés qui ont été mis en place. Les scripts que l'on obtient sont en fait du temps réel simulé. On part donc d'une séquence d'entrée définie à l'avance et on simule du temps réel comme si la séquence était produite en temps réel. Cette méthode a l'avantage d'être plus simple à coder et de s'affranchir de nombreux problèmes relatifs à la programmation en temps réel plutôt qu'aux méthodes présentées ici. Cette façon de faire permet de comparer les différentes méthodes avec une séquence d'entrée identique a chaque fois. Les codes sont réalisés avec MATLAB. Le langage MATLAB n'est pas optimisé pour le temps réel mais sa syntaxe permet de manipuler des matrices et s'avère très pratique pour une programmation rapide. En revanche, le langage gère assez mal les boucles (en terme de rapidité d'exécution) qui sont très présentes dans les méthodes décrites. Une implémentation en temps réel dans un langage tel le C++, bien qu'un peu plus complexe à mettre en place, seraient assurément plus efficaces en terme de temps de calcul. Pour être plus réaliste, on a rajouté un paramètre T_{proc} qui représente le temps nécessaire pour effectuer les différents calculs que l'on a volontairement sur-évalué à 1 seconde. En revanche, la méthode de reconstruction de phase basée sur l'oscillateur non linéaire (voir la section 3.4.2) présente très peu de calcul et aucune boucle. On considère que son temps de calcul nécessaire est instantané.

3.1.1 Extraction de série temporelle

Comme mentionné dans le chapitre précédent, l'algorithme prend comme entrée une série temporelle d'attaques (ou "onsets" comme mentionné dans la littérature). On peut citer trois grandes catégories d'approches pour l'obtenir, données ici de la moins complexe à la plus complexe :

– On construit directement la séquence d'entrée par un algorithme. Cette approche permet notamment de tester des cas parfaits ou imparfaits où l'on contrôle l'imperfection. Citons par exemple des grilles parfaites bruitées par un bruit gaussien, ou encore des séquences accélérées ou décélérées où l'on contrôle le facteur d'accélération.

– On extrait la séquence d'entrée d'un formalisme ou d'un outil où les attaques sont directement accessibles. C'est par exemple le cas pour le formalisme MIDI, ou chaque attaque est repérée par un message "NoteOn" indiquant le début d'une note. C'est aussi le cas lorsque l'on utilise des capteurs binaires. De tels capteurs sont alors placés sur les instruments et envoie un booléen lorsque le musicien "attaque" son instrument.
– On extrait la séquence d'entrée directement de l'audio. Ces méthodes sont les plus complexes mais ont l'avantage de s'appliquer dans pratiquement n'importe quel cas. Il existe de nombreuses méthodes dans la littérature utilisées dans l'extraction de tempo. On citera la plus connus qui se base sur une décomposition en banc de filtres avec détection d'enveloppe qui a été originalement développée par Klapuri [Kla03b] et dont de nombreuses variantes sont aujourd'hui décrites.

Dans le cas des deux dernières méthodes, il est alors souvent nécessaire de rassembler plusieurs séquences en une seule. C'est la concaténation. Par exemple, dans le cas d'un formalisme MIDI, le canal de la batterie (par convention le numéro 10) contient autant de pistes que d'éléments qui la composent (grosse caisse, caisse claire, cymbales ...). Il est courant qu'un percussionniste tape en même temps sur plusieurs éléments lors de la composition du rythme musical. Mais si on le perçoit comme tel, ce n'est pas le cas dans la réalité où il existera de manière quasiment systématique un infime décalage entre les deux impacts. Dans le cas d'une détection par banc de filtre, le problème se pose de manière identique lorsque une attaque est aussi bien présente dans le domaine basse ou haute fréquence. Une concaténation brute donnerait donc un double point qui ne traduit pas la volonté du musicien et abaisserait les performances de notre algorithme, en amenant des calculs superflus.

Pour palier ce problème, on met en place un paramètre de tolérance sur la concaténation qui se définit par un temps en seconde, à partir duquel on considère que deux attaques sont identiques. Se basant sur la littérature psycho-acoustique qui décrit les plus petites différences perceptibles à l'oreille, on considèrera dans la suite une tolérance de l'ordre de 10 mili-secondes.

On précisera enfin que l'on n'a pas implémenté de méthode d'extraction basée sur l'audio, n'étant pas dans le sujet du stage. En revanche, on peut prendre comme entrée un fichier MIDI ou construire directement la séquence souhaitée.

3.1.2 Paramètrisation des fenêtres

Pour faire du suivi de tempo en temps réel, on utilise la méthode classique de fenêtre glissante sur la séquence d'entrée. L'exécution de l'algorithme 2 nécessite approximativement une vingtaine de nombres d'événements pour être efficace (voir la section 2.3.2). Cela nécessite donc une certaine taille de fenêtres d'observation qui va impliquer un retard sur la mise à jour du tempo. En regardant l'algorithme 2, il semble naturel de paramétrer ces fenêtres par le nombre d'événements qu'elles contiennent. On s'assure ainsi un temps de calcul globalement identique pour chaque fenêtre. Cependant, cette façon de faire peut s'avérer problématique dans le cas d'événements singuliers dans le morceau. On peut par exemple citer un roulement de batterie qui impliquera un rétrécissement local de la fenêtre qui peut s'avérer problématique. Cependant, si on considère

des fenêtres de taille temporelle constante, toujours dans l'exemple d'un roulement, on risque d'obtenir une fenêtre comportant un grand nombre d'événements qui va énormément rallonger le temps de calcul (Nouno calcule une complexité de $\mathcal{O}\left(\frac{n^2(n-1)P_{max}}{2}\right)$ pour l'algorithme 1, sensiblement identique à l'algorithme 2). Finalement, le problème d'une taille constante en terme d'événements ou bien en terme temporel n'est pas triviale, et dépend énormément de la séquence d'entrée. Le choix est donc laissé à l'utilisateur et les deux méthodes sont implémentées. On considérant un BPM *moderato* de $120bpm$ et une durée de note minimum moyenne située à la double croche, on obtient comme taille de fenêtre approximative pour obtenir une vingtaine d'événements : $Taille = \frac{20 * \cdot 60}{4 * 80} = 2.5s$.

On implémente aussi un recouvrement des fenêtres que l'on maintient à environ 75%. Cela nous permettra de mettre à jour notre tempo toutes les 0.625 secondes si on considère des tailles de fenêtres de 2.5 secondes, ou tous les 5 événements. Cela nous permet un plus grande réactivité pour notre estimation, et aura son utilité dans le filtrage Kalman. Par exemple, dans le cas d'un saut de tempo (passage instantané d'un pas Δ_1 à un pas Δ_2), le nouveau pas influencera le résultat final de la prédiction de tempo beaucoup plus rapidement. Un tel changement fera ressortir dans le vecteur d'état représenté par le vecteur qui associe à chaque $\Delta \in [\Delta_{min} \quad 2 \cdot \Delta_{min}]$ la valeur $1 - \sigma$ les deux valeurs de Δ_1 et Δ_2. (voir section 3.1.3). Le Filtrage de Kalman pourra ainsi s'adapter beaucoup plus rapidement a cette nouvelle valeur.

3.1.3 Construction du vecteur d'état

Densité de probabilité

Lors de l'application de l'algorithme 2, on récupère un ensemble de triplets $(s_i, \Delta_i, \sigma_i)_{\substack{i \in \mathbb{N} \\ 1 \leq i \leq N_{obs}}}$ dont on va finalement utiliser uniquement les couples $(\Delta_i, \sigma_i)_{\substack{i \in \mathbb{N} \\ 1 \leq i \leq N_{obs}}}$ pour faire notre traitement statistique. L'idée est de suivre la métrique du tatum représenté par Δ_i et dont ce paramètre est d'autant plus probable comme valeur de la périodicité que sa valeur σ_i associée est faible. On choisit ainsi d'assimiler le paramètre σ_i à une mesure indirecte de la probabilité de Δ_i d'être le bon. Comme on a $\sigma \in [0 \quad 1]$, on choisit d'effectuer la transformation $\sigma' \leftarrow 1 - \sigma \in [0 \quad 1]$. Cette transformation permet d'assimiler la valeur σ'_i à $P(\Delta* = \Delta_i)$, à un facteur multiplicatif près. On obtient alors un ensemble discret $(\Delta_i, P(\Delta* = \Delta_i))_{\substack{i \in \mathbb{N} \\ 1 \leq i \leq N_{obs}}}$ qui représente un ensemble de mesures de la fonction de densité de probabilité $f : \Delta \rightarrow f(\Delta)$ avec $\Delta \in [\Delta_{min} \quad 2 \cdot \Delta_{min}]$ à un facteur multiplicatif près. Ce facteur vient du fait que pour être réellement une densité de probabilité, on devrait avoir $\int_{\Delta_{min}}^{2 \cdot \Delta_{min}} f(\Delta) \mathrm{d}\Delta = 1$, chose que l'on ne peut pas affirmer. Cela n'a cependant pas d'influence car l'important est de pouvoir comparer les différentes probabilités $P(\Delta* = \Delta_i)$ entre elles. On veut ainsi pouvoir suivre l'évolution dans le temps de $f(\Delta)$. La valeur optimale de Δ est ainsi définie comme $\Delta* = \max_{\Delta}\{f(\Delta)\}$. Cela sera plus efficace que suivre le scalaire $\Delta*$ car dans le cadre d'un suivi de paramètres, cela autorise des variations non-continues de $\Delta*$. Les variations de $f(\Delta)$ seront par contre continues. Par exemple, dans le cadre d'un saut de tempo de Δ_1 à Δ_2, on verra la fonction $f(\Delta)$ présenter deux "pics" correspondant à Δ_1 et Δ_2. Celui relatif à Δ_1 diminuera à mesure que l'on avance dans le temps et que l'on dépasse de

la localisation temporelle du saut tandis que le pic relatif à Δ_2 augmentera.

Le problème est alors que la répartition des Δ_i que renvoie l'algorithme 2 dépend de la séquence d'entrée, et qu'il n'est, en conséquence, pas possible de comparer directement la mesure de $f(\Delta)$. On est alors dans l'obligation de reconstruire un vecteur d'état à partir de notre mesure. Concrètement, cela veut dire que l'on doit pouvoir associer à une valeur quelconque de $\Delta \in [\Delta_{min} \quad 2 \cdot \Delta_{min}]$ une valeur de $f(\Delta)$. On a développé pour cela deux méthodes décrites dans les sections suivantes. L'une d'elle est basée sur une interpolation, l'autre est basée sur une estimation par noyau. On a aussi cherché à développer une méthode de "fitting" (à l'image de celle des moindres carrés) mais le caractère incomplet de la mesure rend la mise en place d'une telle méthode complexe. Avant de décrire ces méthodes, on propose ici différentes considérations sur la mesure que l'on obtient de $f(\Delta)$ par l'algorithme 2.

Considérations sur la mesure

On cherche donc, pour une séquence d'entrée quelconque, à obtenir un valeur de $f(\Delta)$ pour une série $\{\Delta_k\}_{\substack{k\in\mathbb{N} \\ 1\leq i\leq N_f}}$ définie à l'avance de N_f points.

L'espace des Δ doit être un espace logarithmique binaire.

Reprenons notre modèle de musicien parfait posé dans la section 2.3.1. On a pour le musicien parfait : $\left\{t_j^p\right\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq n}} \subset \mathbb{R}$ tous sur la grille du tatum. Donc, il existe pour tout j, des valeurs de h_j, s^p* et Δ^p* telles que $t_j^p = s^p* + h_j \cdot \Delta^p*$. On peut en déduire que pour un tel musicien, $\sigma^p(s^p*, \Delta^p*) = 0$. On constate que l'on aura alors à nouveau $\sigma^p = 0$ pour $\Delta = {}^{\Delta^p*}/_2$, puis pour $\Delta = {}^{\Delta^p*}/_4$ et ainsi de suite. De même, la meilleure mesure dans l'intervalle de type $[\Delta \quad 2\cdot\Delta]$ qui contient la valeur $2\cdot\Delta^p*$ sera certainement $\sigma^p(s^p*, 2\cdot\Delta^p*)$ et prendra la valeur ${}^l/_n$ avec l le nombre de points t_j telle que h_j soit un entier impair. En fait, les motifs que va former la fonction $f(\Delta)$ sur l'intervalle $[\Delta_{min} \quad 2\cdot\Delta_{min}]$ seront analogues au facteur ${}^l/_n$ près si on est dans une intervalle supérieure à celui contenant le tatum, à ceux formés sur l'intervalle $[2\cdot\Delta_{min} \quad 4\cdot\Delta_{min}]$ et ainsi de suite. Cela correspond aux propriétés d'un espace logarithmique binaire. C'est en conséquence dans ce genre d'espace que l'on a le plus de chance d'obtenir une distribution symétrique autour de $\Delta*$ pour la valeur $1-\sigma$. Ainsi, on prendra comme répartition pour notre série de référence $\{\Delta_k\}_{\substack{k\in\mathbb{N} \\ 1\leq i\leq N_f}}$:

$$\Delta_k = \Delta_{min} \cdot 2^{\frac{k}{N_f}} \in [\Delta_{min} \quad 2\cdot\Delta_{min}] \qquad \forall k = 1 \cdots N_f \tag{3.1}$$

Mesure incomplète et forme attendue de la distribution

Comme on l'a déjà mentionné dans la section 2.3.2, la mesure est incomplète parce que rien ne nous assure que la phase s_i associée au couple (Δ_i, σ_i) issue de l'algorithme 2 est la meilleure pour le Δ_i considéré. Ainsi, si on considère une mesure optimale en terme de phase pour un pas donné $\sigma(s*, \Delta)$, on est sûr d'avoir $\sigma(s*, \Delta) < \sigma(s, \Delta) \ \forall s$ par définition. L'ensemble des valeurs issues de la mesure sont donc toujours plus grandes que la mesure optimale. Si on considère la

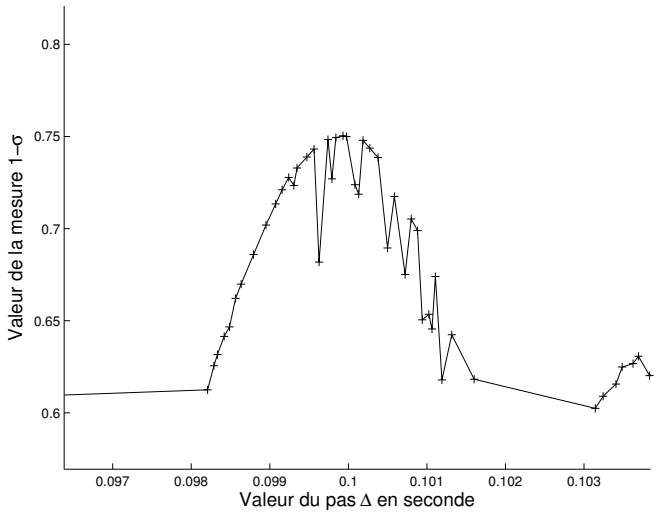

FIGURE 3.1 – Zoom sur une distribution type repartie autour de la valeur cible $\Delta* = 0.1s$. On peut reconnaître une courbe en forme de "cloche", que l'on assimilera à une gaussienne repliée sur un espace circulaire. Cette gaussienne est bruitée par des valeurs négatives uniquement, issues du caractère incomplet des mesures dû à une phase non optimale.

valeur $\sigma' = 1 - \sigma$, la densité de probabilité $f(\Delta)$ idéale que l'on cherche à construire est alors un majorant de l'ensemble des couples (s, Δ).

Il est très difficile de déterminer à partir de l'équation 2.8 la forme des distributions que l'on attend pour la densité de probabilité $f(\Delta)$ car cette équation dépend de la séquence d'entrée. Plus précisément, il faudrait pouvoir exprimer la phase optimale $s*_\Delta$ relative à un pas Δ donné. Il n'est pas possible d'obtenir une telle expression. En regardant la forme des distributions que l'on obtient de manière empirique, on fait la supposition classique d'être en présence d'un mélange de Gaussiennes dont leurs centres respectifs sont les pas Δ qui correspondent à la métrique visée du musicien, en particulier celle du tatum. Remarquons que lors de nos essais sur un corpus de jazz, il apparait souvent deux périodicités privilégiées correspondant au tatum de la croche (ou double croche), ainsi que celle du triolet. Ces deux pulsations n'ont en effet pas un rapport correspondant à une puissance de deux.

Enfin, comme nous recherchons notre périodicité dans l'intervalle $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$, nous considérons en fait un repliement logarithmique (voir section précédente) circulaire de l'espace des périodicités. Cette hypothèse n'est pas totalement exacte, puisque nous avons montré que les motifs se répétaient uniquement à un facteur multiplicatif près, et que la mesure était optimale pour le tatum (voir section 2.3.1). Mais justement parce que les motifs se répètent, et que l'on ne considère qu'une seule octave de tempo, on considèrera que cette hypothèse restera valide. Ainsi, on considère un mélange de Gaussiennes replié, pouvantt être approximées par des distributions de von-Mises, ce qui nous permettra lors d'un suivi de passer de manière continue de la valeur Δ_{min} à la valeur $2 \cdot \Delta_{min}$.

Un exemple type du type de distribution obtenue est présenté à la figure 3.1

3.2 Methode de construction du vecteur d'état

Sont présentées dans cette section les méthodes pour obtenir les valeurs de $f(\Delta_k)$ dont les Δ_k sont issus de l'équation 3.1 à partir des couples $(\Delta_i, \sigma'_i)_{\substack{i\in\mathbb{N} \\ 1\leq i\leq N_{obs}}}$ issus de l'algorithme 2. On considère maintenant que l'indice i représente les Δ_i classés dans l'ordre croissant (et non par σ'_i croissant comme précédemment). On pose ainsi comme **vecteur d'état :** $\Psi = \{\Psi_k = f(\Delta_k)\}_{\substack{k\in\mathbb{N} \\ 1\leq k\leq N_f}}$

3.2.1 Interpolation

L'approche la plus simple et la plus naïve est d'utiliser des méthodes d'interpolations. On a deux méthodes d'interpolations à notre disposition :

- **L'interpolation linéaire :** Cette méthode extrapole la valeur de Ψ_k par considération du segment de droite provenant des points (Δ_i, σ'_i) et $(\Delta_{i+1}, \sigma'_{i+1})$ avec $\Delta_k \in [\Delta_i \quad \Delta_{i+1}]$.
- **L'interpolation cubic ou méthode des Splines :** Cette méthode est semblable à la méthode précédente mais utilise des polynômes d'ordre 3, dont les coefficients sont en conséquence issus de la considération de 4 points (Δ_i, σ'_i).

Pour réaliser ces interpolations, on utilise la fonction `interp1` de MATLAB qui prend comme entrées `(X , Y , Xq, 'methode')`, ou `X` et `Y` représentent des vecteurs contenant les abscisses et ordonnées des points sur lesquels est construite l'interpolation, soit $(\Delta_i, \sigma'_i)_{\substack{i\in\mathbb{N} \\ 1\leq i\leq N_{obs}}}$; `Xq` le vecteur d'abscisse duquel on veut obtenir nos valeurs Ψ_k, soit $\{\Delta_k\}_{\substack{k\in\mathbb{N} \\ 1\leq i\leq N_f}}$; et `'methode'` une chaîne de caractères qui indique la méthode utilisée. Rien de nous permet d'affirmer que nos Δ_i contiennent les deux valeurs extrêmes Δ_{min} et $2 \cdot \Delta_{min}$. Pour pouvoir réaliser une interpolation sur l'ensemble de nos Δ_k, on rajoute donc les points $(\Delta_{N_{obs}}/2 - eps, \sigma'_{N_{obs}})$ et $(2 \cdot \Delta_1\} + eps, \sigma'_1)$ parce que l'on considère que notre espace de recherche est circulaire. La valeur *eps* représente la plus petite valeur considérée par MATLAB et permet d'éviter d'avoir deux valeurs pour Δ_{min} et $2 \cdot \Delta_{min}$ si celles-ci sont présentes. Comme l'espace considéré est logarithmique, on effectue la transformation $\Delta \leftarrow \log_2(\Delta)$ avant d'effectuer l'interpolation. Un version de l'algorithme d'interpolation est l'algorithme 5.

Algorithm 5 Algorithme d'interpolation du vecteur d'état Ψ à partir du résultat de l'algorithme 2

Require: $(\Delta_i, \sigma'_i)_{\substack{i\in\mathbb{N} \\ 1\leq i\leq N_{obs}}}$; 'methode' ; $\{\Delta_k\}_{\substack{k\in\mathbb{N} \\ 1\leq i\leq N_f}}$; eps

$\{\Delta_i\} \leftarrow \{\Delta_{N_{obs}}/2 - eps \quad \Delta_1 ..... \Delta N_{obs} \quad 2 \cdot \Delta_1 + eps\}$

$\{\sigma'_i\} \leftarrow \{\sigma'_{N_{obs}} \quad \sigma'_1 \cdots \sigma'_{N_{obs}} \quad \sigma'_1\}$

$\{\Delta_i\} \leftarrow \log_2(\{\Delta_i\})$

$\{\Delta_k\} \leftarrow \log_2(\{\Delta_k\})$

$\{\Psi_k\} =$ `interp1`$(\{\Delta_i\}, \{\sigma'_i\}, \{\Delta_k\}, \text{'methode'})$

return $\Psi = \{\Psi_k\}$

Cette approche ne pallie pas le problème de l'incomplétude des données puisqu'elles font partie intégrante de l'extrapolation, mais a l'avantage de ne pas être biaisé. Le meilleur pas issu de l'algorithme 2 reste ainsi inchangé par rapport à $\Delta_{k_{opt}}$avec k_{opt} tel que $\max_k\{\Psi_k\}$.

3.2.2 Estimation par noyau

Un solution pour pallier le problème de l'incomplétude des données est d'utiliser le principe de l'estimation par noyau. L'idée est de sélectionner les N_{ker} meilleures mesures $\{\Delta_j\}_{1\leq \overset{j\in\mathbb{N}}{j}\leq N_{ker}}$ issues de l'algorithme 2 puis de re-créer Ψ à partir de ces mesures. Concrètement, on va re-créer Ψ en utilisant l'estimation :

$$f(\Delta) = \frac{1}{N_{ker}} \sum_{j=1}^{N_{ker}} K(\Delta - \Delta*_j, \Sigma_j) \tag{3.2}$$

avec K le noyau de paramètre Σ_j utilisé. Dans notre cas, comme mentionné dans la section 3.1.3, on utilisera comme noyau une distribution de von-Mises sur un espace logarithmique circulaire. Une telle distribution a pour paramètres sa moyenne μ qui sera évidemment associée à $\Delta*_j$ et sa variance $\frac{1}{\kappa}$. Ces paramètres sont analogues aux paramètres de moyenne et de variance d'une gaussienne. Son expression est alors définie pour tout x appartenant à un intervalle de longueur $2 \cdot \pi$ par :

$$K(x, \mu_j, \kappa_j) = \frac{e^{\kappa_j \cos(x-\mu_j)}}{2\pi I_0(\kappa_j)} \tag{3.3}$$

où I_0 représente la fonction de Bessel modifié à l'ordre 0. Dans notre cas, on considère la variable Δ qui est dans l'intervalle $[\Delta_{min} \quad 2 \cdot \Delta_{min}]$. On transforme ainsi la formule par :

$$K(\Delta, \Delta*_j, \kappa_j) = \frac{e^{\kappa_j \cos\left(\frac{2\pi}{\Delta_{min}}(\Delta-\Delta*_j)\right)}}{2\pi I_0(\kappa_j)} \tag{3.4}$$

On perd alors la propriété relative à la densité de probabilité qui implique que l'aire sous la courbe est égale à 1, mais cela n'a pas d'importance dans notre cas, puisqu'il s'agit uniquement de récupérer le maximum de la courbe. Cette fonction est en fait une approximation de la projection de la gaussienne sur le cercle. Pour que cette estimation prenne en compte la valeur de la mesure σ'_i, on propose de relier les valeurs de κ_j a celle-ci. Pour cela, considérons la hauteur maximale d'une distribution gaussienne repliée qui s'exprimer par :

$$f_{WN}(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} \exp\left[\frac{-(x-\mu+2\pi k)^2}{2\sigma^2}\right] \tag{3.5}$$

Si l'on considère un écart type σ assez faible par rapport à la longueur de l'intervalle $2 \cdot \pi$, la hauteur maximale prendra son maximum en $x = \mu$ et vaudra : $H_{max} \approx \frac{1}{\sigma\sqrt{2\pi}}$. On est alors quasiment dans le cas d'une courbe de Gauss classique. Pour une variable aléatoire auquel on assigne une telle densité de probabilité, on considère que 93% des données sont contenues dans

l'intervalle $[\mu - 2\sigma \quad \mu + 2\sigma]$. On peut donc affirmer que si $4\sqrt{\frac{1}{\kappa}} < \Delta_{min}$, on aura effectivement $H_{max} \approx \frac{\sqrt{\kappa}}{\sqrt{2\pi}}$ pour une distribution de von-Mises modifiée issue de l'équation 3.4. On souhaite que cette hauteur soit proportionnelle à notre mesure σ' de telle sorte que $\frac{\sqrt{\kappa}}{\sqrt{2\pi}} = A \cdot \sigma'$, avec A une constante. Pour déterminer cette constante, on est dans l'obligation de faire une hypothèse sur notre valeur souhaitée de l'écart type $\frac{1}{\kappa}$ de notre distribution. On choisit alors comme référence la valeur pour la mesure $\sigma' = {}^1/_4$ que l'on considère arbitrairement (plus exactement de manière empirique) comme la pire valeur possible pour la mesure σ' (on montre en effet dans la section 3.2.3 qu'il est impossible de récupérer une valeur nulle pour σ). Pour une telle mesure, on choisit d'attribuer une valeur pour la variance de $\frac{1}{\sqrt{\kappa}} = \frac{\Delta_{min}}{4}$, ce qui nous permet de respecter l'hypothèse des 93% de données énoncée ci-dessus. Vient alors $A = \frac{16}{\sqrt{2\pi}\Delta_{min}}$ et finalement $\frac{1}{\sqrt{\kappa}} = \frac{\Delta_{min}}{16\sigma'}$ ou encore $\kappa = \left(\frac{16\sigma'}{\Delta_{min}}\right)^2$

En vérité, à l'image des considérations de la section précédente, on appliquera cette estimation sur un espace logarithmique binaire circulaire. On va donc aussi effectuer la transformation $\Delta \leftarrow \log_2(\Delta)$ avant d'appliquer l'estimation. La gaussienne dans un espace linaire sera donc déformée par l'espace logarithmique. Les considérations ci-dessus devrait donc être faites dans l'espace logarithmique binaire. Cela ne pose pas de problème à l'exception des considérations sur la variance. En effet, l'estimation étant appliquée dans l'espace logarithmique, il n'y a plus de symétrie de la distribution et la correspondance entre la variance souhaitée dans l'espace linaire et celle que l'on doit prendre dans l'espace logarithmique va donc dépendre de la position de la distribution sur l'octave considérée, soit du paramètre $\Delta *_j$. Cependant, comme notre intervalle est uniquement composé d'une octave, on pourra prendre comme correspondance pour l'écart-type une simple moyenne entre la partie gauche et la partie droite de notre distribution telle que :

$$\frac{1}{\sqrt{\kappa}} \rightarrow \frac{\log_2(\Delta *_j + \frac{1}{\sqrt{\kappa}}) - \log_2(\Delta *_j - \frac{1}{\sqrt{\kappa}})}{2} \tag{3.6}$$

Une version de l'algorithme d'estimation par noyau est l'algorithme 6 dans lequel $(\Delta_i, \sigma'_i)_{\substack{i \in \mathbb{N} \\ 1 \leq i \leq N_{ker}}}$ représente les N_{ker} meilleurs couples au sens de la mesure σ' issue de l'algorithme 2.

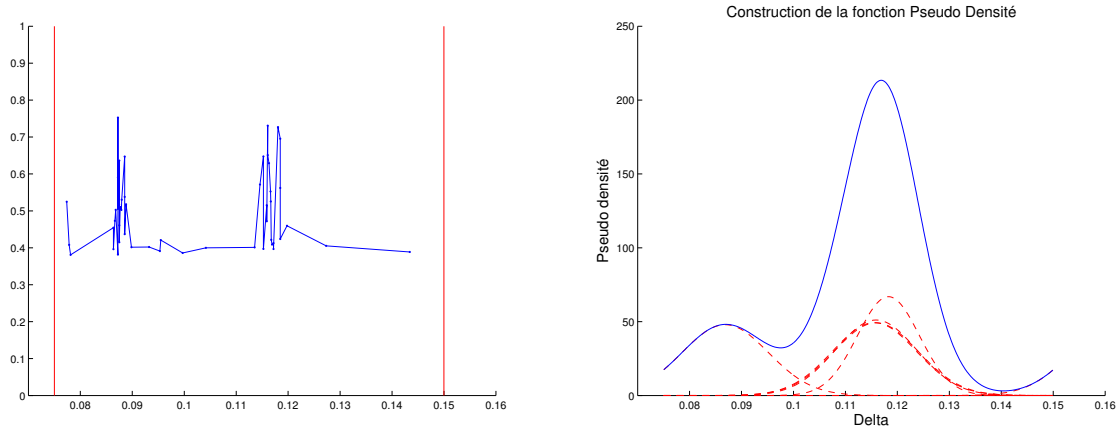

FIGURE 3.2 – illustration de la méthode d'estimation par noyau et de la méthode d'estimation par interpolation linéaire. À gauche l'estimation par interpolation linéaire et à droite celle obtenue par l'interpolation par noyau. On constate une grosse différence d'allure entre les deux méthodes, mais on ne peut affirmer qu'une est meilleure que l'autre

Algorithm 6 Algorithme d'estimation par noyau du vecteur d'état Ψ à partir du résultat de l'algorithme 2

Require: $(\Delta_i, \sigma'_i)_{\substack{i\in\mathbb{N} \\ 1\leq i\leq N_{ker}}}$; $\{\Delta_k\}_{\substack{k\in\mathbb{N} \\ 1\leq i\leq N_f}}$; N_{ker}

$\Psi_k = \{0...0\}_{N_f}$ Initialisation

$\{\Delta_k\} \leftarrow \log_2(\{\Delta_k\})$

for i=1 to N_{ker} **do**

$\{\Delta_i\} \leftarrow \log_2(\{\Delta_i\})$

$\Sigma_i \leftarrow \dfrac{\Delta_{min}}{16\sigma'_i}$

$\Sigma_i \leftarrow \dfrac{\log_2(\Delta *_j + \Sigma_i) - \log_2(\Delta *_j - \Sigma_i)}{2}$

$\kappa_i \leftarrow \dfrac{1}{\Sigma_i^2}$

$\{\Psi_k\} \leftarrow \{\Psi_k\} + \dfrac{e^{\kappa_i \cos\left(\frac{2\pi}{\Delta_{min}}(\{\Delta_k\} - \Delta *_i)\right)}}{2\pi I_0(\kappa_i)}$

end for

$\{\Psi_k\} \leftarrow \{\dfrac{\Psi_k}{N_{ker}}\}$

return $\Psi = \{\Psi_k\}$

Cette méthode présente l'avantage de pallier l'incomplétude des données. Dans la pratique, on a typiquement $N_{ker} = 10$, ce qui veut dire que l'on sélectionne les 10 meilleurs mesures. Ces mesures étant les meilleures, elles correspondent vraisemblablement à des mesures pour lesquelles la phase est optimale. En revanche, cette méthode peut présenter un biais sur le pas optimum défini par le maximum des Ψ_k. En effet, si les Δ_i ne sont pas équi-répartis autour de la valeur $\Delta*$ recherchée (on constate que c'est par exemple le cas lors d'accélération de tempo), la somme des noyaux peut ne pas avoir son maximum sur la meilleure mesure σ'_i. Cependant, ce biais peut s'avérer positif s'il est dans le "bon sens", toujours dans l'exemple de l'accélération du tempo.

3.2.3 Normalisation des mesures

Si on reprend notre modèle de musicien parfait posé dans la section 2.3.1 et évoqué dans la section 3.1.3., on a toujours $\sigma^p(s*, \Delta*) = 0$ et $\sigma^p(s* + {}^{\Delta*}/_2, \Delta*) = 1$ ce qui donne $\sigma'^p(s*, \Delta*) = 1$ et $\sigma'^p(s* + {}^{\Delta*}/_2, \Delta*) = 0$. En fait dans la pratique, on ne peut jamais obtenir la mesure $\sigma'^p(s* + {}^{\Delta*}/_2, \Delta*) = 0$ à partir de l'algorithme 2. En effet, le centre s est issu des points de la séquence $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq n}}$. Si on a $\sigma'^p(s* + {}^{\Delta*}/_2, \Delta*) = 0$, cela voudrait dire qu'aucun des points n'est sur la grille. Or d'après le théorème fondamental sur lequel est basé l'algorithme, au moins deux points passent par la grille. Une telle mesure est donc impossible. De même, il existe toujours un bruit dans le jeu du musicien, et il est dans la pratique impossible d'obtenir $\sigma' = 1$.

Si on reprend le modèle du musicien imparfait, c'est-à-dire qu'on prend pour hypothèse qu'il produit une série de notes $\left\{t_j^{ip}\right\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq n}} \subset \mathbb{R}$ telle que $t_j^{ip} = t_j^p + T_j$, avec T une variable aléatoire telle que $T \sim \mathcal{N}(0, \sigma_T)$. On a montré que chaque $erreur_k(s^{ip}*, \Delta^{ip}*)$ suivra une loi demi-normale de paramètre σ_T, dont la variance s'exprime par : $Var_{|T|} = \sigma_T^2\left(1 - \frac{2}{\pi}\right)$. En conséquence, et sous couvert que le musicien a un écart type tel que $\sqrt{Var_{|T|}} \ll \frac{\Delta^p*}{2}$, la valeur de la mesure optimale suivra aussi une loi demi-normale telle que $\sigma^{ip}(s^{ip}*, \Delta^{ip}*) \sim |\mathcal{N}|(0, n*\sigma_T)$. L'espérance d'une telle loi est $E(T) = \frac{\sigma_T\sqrt{2}}{\sqrt{\pi}}$.On peut faire un raisonnement similaire en considérant la plus mauvaise mesure $\sigma'^{ip}(s* + {}^{\Delta*}/_2, \Delta*)$. En conséquence, la mesure est comprise dans un intervalle plus petit que $[0 \quad 1]$ puisqu'en moyenne dans l'intervalle $[\frac{n\sigma_T\sqrt{2}}{\sqrt{\pi}} \quad 1 - \frac{n\sigma_T\sqrt{2}}{\sqrt{\pi}}]$. De plus, si on considère que on prend que les N_{obs} meilleures mesures, cet intervalle est encore plus petit.

On peut donc se poser la question de la re-normalisation des mesures avant d'appliquer les méthodes précédemment décrites. Une telle re-normalisation serait alors :

$$\sigma' \leftarrow \frac{\sigma' - \min(\sigma')}{\max(\sigma' - \min(\sigma'))} \subseteq [0 \;\; 1] \tag{3.7}$$

L'avantage de faire un telle normalisation est que l'on sera plus stable lors du filtrage. On ne risque pas confondre la mesure avec un bruit car on maintient finalement la différence entre notre meilleure et plus mauvaise mesure à 1. En conséquence, l'estimation de la variance des bruits nécessaires au filtrage (voir la section 3.3) est meilleure. En revanche, parce que chaque fenêtre voit sa valeur maximum mise à 1 et sa valeur minimum mise a 0, on perd toute notion de comparaison entre différentes fenêtres. D'un point de vue heuristique, cela revient à considérer avec le même poids la pertinence de chaque fenêtre dans le suivi de tempo. Une telle normalisation aura aussi une influence non négligeable dans le cas d'une estimation par noyau, puisque la variance des distributions est directement reliée à la valeur de σ'. En conséquence, le vecteur d'état Ψ sera composé essentiellement de mesures proches de la valeur 1 et verra donc ses distributions très concentrées autour de la valeur moyenne considérée. Cette normalisation peut finalement s'avérer positive dans certains cas mais pas du tout dans d'autres. On la laisse donc comme une option à l'utilisateur.

3.3 Filtrage de Kalman

3.3.1 Principe du Filtrage de Kalman

Le filtre de Kalman est un filtre à réponse impulsionnelle infinie qui estime les états d'un système dynamique à partir d'une série de mesures incomplètes ou bruitées. Il se prête donc bien aux mesures issues de l'algorithme 2 Le filtre de Kalman est un Estimateur récursif. C'est-à-dire que pour estimer l'état courant, seule l'estimation de l'état précédent et les mesures actuelles sont nécessaires. L'historique des observations et des estimations n'est ainsi pas requis. Il se prête donc particulièrement bien au temps réel.

L'état du filtre est représenté par 2 variables :

$\hat{\mathbf{x}}_{p|p}$ l'estimation de l'état à l'instant "p" ; $\mathbf{P}_{p|p}$, La matrice de covariance de l'erreur (une mesure de la précision de l'état estimé).

Le filtre de Kalman a deux phases distinctes : "'Prédiction"' et "'Mise à jour"'. La phase de prédiction utilise l'état estimé de l'instant précédent pour produire une estimation de l'état courant. Dans l'étape de mise à jour, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise. Ces deux phases sont alors décrites comme suit :

Prediction

$\hat{\mathbf{x}}_{p|p-1} = \mathbf{F}_p\hat{\mathbf{x}}_{p-1|p-1}$ (état prédit)

$\mathbf{P}_{p|p-1} = \mathbf{F}_p\mathbf{P}_{p-1|p-1}\mathbf{F}_p^T + \mathbf{Q}_p$ (estimation prédite de la covariance)

avec :

- $\mathbf{F}_p$: La matrice qui relie l'état précédent p-1 à l'état actuel p
- $\mathbf{P}_{p|p-1}$: La matrice d'estimation "a priori" de la covariance de l'erreur
- $\mathbf{Q}_p$:La matrice de covariance du bruit de processus.

Mise à Jour

$\tilde{\mathbf{y}}_p = \mathbf{z}_p - \mathbf{H}_p\hat{\mathbf{x}}_{p|p-1}$ (innovation)

$\mathbf{S}_p = \mathbf{H}_p\mathbf{P}_{p|p-1}\mathbf{H}_p^T + \mathbf{R}_p$ (covariance de l'innovation)

$\mathbf{K}_p = \mathbf{P}_{p|p-1}\mathbf{H}_p^T\mathbf{S}_p^{-1}$ (gain de Kalman "optimal")

$\hat{\mathbf{x}}_{p|p} = \hat{\mathbf{x}}_{p|p-1} + \mathbf{K}_p\tilde{\mathbf{y}}_p$ (état mis à jour)

$\mathbf{P}_{p|p} = (\mathbf{I}_{N_f} - \mathbf{K}_p\mathbf{H}_p)\mathbf{P}_{p|p-1}$ (covariance mise à jour)

Avec

- $\mathbf{z}_p$: observation ou mesure du processus à l'instant k
- $\mathbf{H}_p$: La matrice qui relie l'état $\mathbf{x}_p$ à la mesure $\mathbf{z}_p$
- $\mathbf{P}_{p|p}$: La matrice d'estimation "a posteriori" de la covariance de l'erreur
- $\mathbf{R}_p$: La matrice de covariance du bruit de mesure.

3.3.2 Mise en application

Décrivons maintenant comment se met en application le filtrage de Kalman sur notre vecteur d'état Ψ. On considère alors dans leur ordre d'apparition :

- p représente la numérotation de la dernière fenêtre analysée, avec laquelle on va mettre a jour notre filtre de Kalman. $p-1$ représente la fenêtre immédiatement précédente.
- $\hat{\mathbf{x}}_{p|p}$ représente le vecteur d'état $\hat{\Psi}_p = \{\hat{\Psi_{1|p}} \cdots \hat{\Psi_{k|p}} \cdots \hat{\Psi_{N_f|p}}\}$ issu de la fenêtre p après filtrage. Pour initialiser cette variable, on prend $\hat{\Psi}_1 = \Psi_1$, la première valeur non filtrée.
- $\mathbf{P}_{p|p}$ est la matrice d'estimation "a priori" de la covariance de l'erreur. Elle est calculé dans l'algorithme mais doit être initialisée pour la première fenêtre. On prendra comme valeur pour l'initialisation la valeur de référence considérée à la section 3.2.2 soit : $\mathbf{P}_{1|1} = \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}$. On considère en effet que les candidats potentiels Δ_k sont indépendants et identiquement considérés.
- $\mathbf{F}_p$ est la matrice qui relie l'état actuel à l'état précèdent. Cette matrice est utile dans lorsqu'on agit sur le processus par le biais d'une commande. Dans notre cas, notre a priori est que le tempo n'a pas changé, on a donc $\mathbf{F}_p = \mathbf{I}_{N_f}$ (matrice identité d'ordre N_f).
- $\mathbf{Q}_p$ est la matrice de covariance du bruit de processus. Dans notre cas, c'est donc la matrice de covariance de la variation de tempo. Autrement dit, cette matrice donne un a priori sur la dynamique de variation de tempo. L'estimation de ce paramètre est crucial pour le filtrage de Kalman. On considère que chacun des candidats potentiels Δ_k sont indépendants et ont la même variance. Aussi, on considère $\mathbf{Q}_p$ comme une matrice diagonale proportionnelle à l'identité. En reprenant les considérations de la section 3.2.2, on considère la valeur de référence $\frac{1}{\sqrt{\kappa}} = \frac{\Delta_{min}}{16\sigma'}$ pour la variance, avec $\sigma' = {}^1/_2$, soit $\mathbf{Q}_p = \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}$. On pourra éventuellement tester plusieurs valeurs pour comparer les performances de l'algorithme (voir section 4).
- $\mathbf{z}_p$ sera l'observation issue de la mesure, soit Ψ_p issue de la fenêtre p avant filtrage.
- $\mathbf{H}_p$ est alors l'identité $\mathbf{I}_{N_f}$, puisque l'on a déjà mis le résultat de la mesure sous un bon format avec la section 3.2.
- $\mathbf{R}_p$: La matrice de covariance du bruit de mesure. Dans notre cas, c'est donc la matrice de covariance de l'imprécision du musicien. Autrement dit, c'est notre a priori sur la précision du musicien sur la grille du tatum. À l'image de la construction de $\mathbf{Q}_p$, on supposera cette précision indépendante et identique pour chaque Δ_k. Cette valeur doit être supérieure à celle de $\mathbf{Q}_p$. En effet, la variabilité du musicien sur la périodicité est plus grande que la variation de celle-ci. Sinon, cela voudrait dire que la précision du musicien est telle que que chaque écart par rapport à la grille est inférieur à la variation de tempo de celle-ci dans le cadre d'un suivi de périodicité (saut de tempo exclu). On prendra arbitrairement comme valeur $\mathbf{Q}_p = 10 \cdot \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}$, mais on fera varier celle-ci dans la partie 4.

Une version de l'algorithme issu de cette application est l'algorithme 7. Dans cet algorithme, on considère un passage de l'état $p-1$ vers l'etat p.

Remarques sur $\mathbf{Q}_p$ et $\mathbf{R}_p$

Ces deux paramètres sont cruciaux pour le filtrage de Kalman. De la manière dont ils sont posés, nous avons en fait $\mathbf{Q}_p = \mathbf{Q}$ et $\mathbf{R}_p = \mathbf{R}$, c'est-à-dire que ces matrices ne changent pas avec le temps. La variabilité du musicien peut dépendre en fait du morceau ou improvisation considérée, tout comme celle du tactus. Il existe dans la littérature des méthodes d'estimation de ces matrices en temps réel. On en présente deux qui ont été testées sans succès dans l'annexe B.

La meilleure chose que l'on puisse faire est de développer des méthodes en amont du suivi de tempo en temps réel. On pourrait par exemple envisager, pour estimer la variation du musicien $\mathbf{R}$, de comparer la précision entre plusieurs musiciens qui sont supposés jouer une note sur un même temps (en comparant par exemple la position de l'attaque d'un musicien par rapport à une grille sur laquelle on lui a demandé de jouer). De même, la variation du tempo $\mathbf{Q}$ pourrait être estimée en demandant à un orchestre de maintenir un tempo constant sans référence et d'estimer alors la variance de celui-ci. De telles méthodes pourraient permettre d'obtenir des valeurs relatives à un orchestre ou à un musicien pour le filtrage. Ce sont des méthodes de calibrage, qui n'ont pas été développées dans ce stage puisque l'on a pas fait d'expérimentation réelle sur des orchestres ou musiciens.

Algorithm 7 Un pas de l'algorithme du filtrage de Kalman sur Ψ

Require: $\widehat{\Psi_{p-1|p-1}}$; $\mathbf{P}_{p-1|p-1}$; $\mathbf{Q}$; $\mathbf{R}$; Ψ_p

(1) Prédiction

$\widehat{\Psi_{p|p-1}} \leftarrow \widehat{\Psi_{p-1|p-1}}$

$\mathbf{P}_{p|p-1} \leftarrow \mathbf{P}_{p-1|p-1} + \mathbf{Q}$

(2) Mise à jour

$\tilde{\mathbf{y}}_p \leftarrow \Psi_p - \hat{\Psi}_{p|p-1}$

$\mathbf{S}_p \leftarrow \mathbf{P}_{p|p-1} + \mathbf{R}$

$\mathbf{K}_p \leftarrow \mathbf{P}_{p|p-1}{\mathbf{S}_p}^{-1}$

$\widehat{\Psi}_{p|p} \leftarrow \hat{\Psi_{p|p-1}} + \mathbf{K}_p\tilde{\mathbf{y}}_p$

$\mathbf{P}_{p|p} \leftarrow (\mathbf{I}_{N_f} - \mathbf{K}_p)\mathbf{P}_{p|p-1}$

(3) Determination du pas optimal

$\Delta_p* \leftarrow \max_k\{\Psi_{p|k}\}$

return $\widehat{\Psi_{p|p}}$, $\mathbf{P}_{p|p}$, Δ_p*

3.4 Reconstruction de la phase

À l'issue de l'algorithme 7, on dispose du pas optimal Δ_p* de la fenêtre d'observation référencée p. En revanche, on ne dispose pas de la phase optimale associée qui va nous permettre de re-construire la grille en temps réel. On présente ici 3 méthodes pour retrouver celle-ci. Les deux premières sont fondée sur une mise à jour de la phase par fenêtre d'observation, la dernière se base sur une mise à jour de la phase à chaque événement.

3.4.1 Mise à jour de la phase à chaque fenêtre

Phase reportée

Puisque l'on dispose de pas optimal Δ_p* issu de la fenêtre p, on commence dans une première approche à chercher la phase optimale pour ce pas dans cette fenêtre. Cela veut dire que nous allons trouver une phase comprise dans l'intervalle $[0 \quad \Delta_p*]$ de la fenêtre p. Or, à ce moment précis, on se situe dans le morceau au temps qui correspond à la fin de la fenêtre plus le temps nécessaire pour faire les calculs précédents. On devra donc reporter la phase trouvée au temps courant. Pour ce faire, on est obligé de considérer la longueur de la fenêtre p. On a défini la longueur de la fenêtre, soit par le nombre n d'éléments de la série issu de la fenêtre p $\{t_{p|k}\}_{k=1\cdots n}$ constant, soit par intervalle de temps constant. Dans le premier cas, le début de la fenêtre est donc défini par le premier élément de la série $t_{p|1}$. Dans le second cas, la phase est en théorie définie par rapport au début de la fenêtre qui ne correspond pas forcement au premier élément de la série $\{t_{p|k}\}_{k=1\cdots n}$. Dans la pratique cependant, pour rendre les méthodes compatibles, on a défini le début de la fenêtre par rapport au premier élément qu'elle contient. On considère donc que dans le cas général, la taille de la fenêtre est définie par $t_{p|n} - t_{p|1}$. Soit s_p une phase de cette fenêtre : celle-ci est donc définie relativement au premier élément $t_{p|1}$ de la fenêtre telle que sa position temporelle absolue soit $t_{p|1} + s_p$. On peut alors calculer la phase reportée $s_{p|report}$ définie comme la première phase dont la position temporelle absolue est supérieure à la valeur $t_{p|n} + T_{proc}$ avec T_{proc} le temps de calcul nécessaire aux algorithmes précédents. Une telle phase se calcule alors comme suit :

$$s_{p|report} = t_{p|n} + T_{proc} + \Delta_p * \left(1 - frac\left(\frac{t_{p|n} + T_{proc} - (t_{p|1} + s_p)}{\Delta_p*}\right)\right) \tag{3.8}$$

Une illustration de ce calcul est disponible sur la figure 3.3 sur laquelle la phase reportée est définie relativement par rapport à la fin de la fenêtre plus le temps de calcul. C'est cette phase que l'on devra prendre en compte lors d'une reconstruction de la grille du tatum.

Test des phases disponibles

L'approche la plus simple et qui respecte le plus l'esprit de l'algorithme 2 est alors de re-tester toutes les phases possibles issues des événements $\{t_{p|k}\}_{k=1\cdots n}$ dans la fenêtre p avec le pas Δ_p*. On prend alors la meilleure au sens de la mesure σ. L'algorithme qui effectue cette opération est l'algorithme 8

Algorithm 8 Algorithme de recherche de phase optimale reportée par test des phases disponibles

Require: Δ_p* ; $\{t_{p|k}\}_{k=1\cdots n}$; T_{proc} ;

$\{t'_{p|k}\} \leftarrow \{t_{p|k}\} - t_{p|1}$

for k=1 to n **do**

$$s_{p|k} \leftarrow \Delta_p * frac\left(\frac{t'_{p|k}}{\Delta_p}\right)$$

$$\sigma_{p|k} \leftarrow \frac{2}{n} \cdot \sum_{q=1}^{n} \min\left\{frac\left(\frac{t'_{p|q} - s_{p|k}}{\Delta_p*}\right), 1 - frac\left(\frac{t'_{q|k} - s_{p|k}}{\Delta_p*}\right)\right\}$$

end for

$s*_p$ tel que $\sigma*_p = \min_k \{\sigma_{p|k}\}$

$$s*_{p|report} \leftarrow \Delta_p * \left(1 - frac\left(\frac{t_{p|n} + T_{proc} - (t_{p|1} + s*_p)}{\Delta_p*}\right)\right) + t_{p|n} + T_{proc}$$

return $s*_{p|report}$

Le problème de cette méthode est qu'elle s'appuie de manière indirecte sur le théorème fondamental. Celui-ci stipule que la grille optimale est forcement issue d'une division des intervalles disponibles de la séquence d'entrée. En conséquence, cette grille passe forcement par deux points de la grille. Or ici, rien ne nous assure que le pas Δ_p* est le pas optimal de la fenêtre considérée puisqu'il prend en compte les événements passés par le biais du filtrage de Kalman. Le théorème ne s'applique donc plus et l'on est pas certains d'obtenir la phase optimale avec cette méthode. De plus, il est alors possible d'obtenir d'importants sauts de phase avec cette méthode. Une solution pour pallier ce problème est proposée dans la section suivante.

Balayage de Phase

La seconde idée est alors de tester les phases par un simple balayage de phase. On définit un nombre N_{phase} correspondant au nombre de phases que l'on va tester et on teste ensuite les phases contenues dans l'intervalle $[0 \quad \Delta*_p]$ découpé en N_{phase} sections. L'avantage de cette technique est qu'il est possible de définir une tolérance sur les sauts de phase par rapport à la phase issue de la fenêtre précédente. Remarquons qu'avec une telle méthode, on doit s'assurer que N_{phase} est suffisamment grand pour que l'on ait $\Delta_{phase} \leq \epsilon_\Delta$, défini à la section 2.3.2. L'algorithme qui effectue cette opération est l'algorithme 9, avec $tol_s \in [0 \quad 1]$ représentant le pourcentage de tolérance d'écart par rapport à la phase précédente.

Algorithm 9 Algorithme de recherche de phase optimale reportée par balayage de phase

Require: $\Delta_{p-1}*$;Δ_p* ; $s_{p-1|report}*$; $\{t_{p|k}\}_{k=1\cdots n}$; T_{proc} ; tol_s ; N_{phase}

$$\{t'_{p|k}\} \leftarrow \{t_{p|k}\} - t_{p|1}$$

La phase $s_{p-1|report}*$ est ramenée dans l'intervalle $[t_{p|n} + T_{proc} \quad t_{p|n} + T_{proc} + \Delta*_{p-1}]$

$$s_{p-1|report} \leftarrow \Delta_{p-1} * \left(1 - frac\left(\frac{t_{p|n} + T_{proc} - s_{p-1|report}}{\Delta_{p-1}*}\right)\right) + t_{p|n} + T_{proc}$$

On créer alors le vecteur des phases possibles de taille N_{phase}

$$\{s_{p|k}\}_{i=1\cdots N_{phase}} = \left\{s_{p-1|report} - \frac{\Delta_p \cdot tol_s}{2} + \frac{\Delta_p \cdot tol_s \cdot (k-1)}{N_{phase}}\right\}$$

for k=1 to N_{phase} **do**

On ramène la phase courante dans l'intervalle $[0 \quad \Delta_p]$

$$s_{p|k} \leftarrow \Delta_p * frac\left(\frac{s_{p|k} - t_{1|k}}{\Delta_p}\right)$$

Calcul de la mesure

$$\sigma_{p|k} \leftarrow \frac{2}{n} \cdot \sum_{q=1}^{n} \min\left\{frac\left(\frac{t'_{p|q} - s_{p|k}}{\Delta_p*}\right), 1 - frac\left(\frac{t'_{q|k} - s_{p|k}}{\Delta_p*}\right)\right\}$$

end for

$$s*_p \text{ tel que } \sigma*_p = \min_k \{\sigma_{p|k}\}$$

On reporte la phase dans le bon intervalle

$$s*_{p|report} \leftarrow \Delta_p * \left(1 - frac\left(\frac{t_{p|n} + T_{proc} - (t_{p|1} + s*_p)}{\Delta_p*}\right)\right) + t_{p|n} + T_{proc}$$

return $s*_{p|report}$

Une illustration de l'intervalle de recherche sur la tolérance de phase est présenté sur la figure 3.3.

Le principal reproche que l'on peut faire aux deux méthodes précédentes est que si on a une erreur sur le pas Δ_p* par rapport a la séquence d'entrée considérée à l'instant présent (c'est par exemple le cas si le tempo varie) cette erreur sera reporté m fois à l'instant présent, avec m le nombre d'éléments de la grille contenu dans l'intervalle de temps composé de la taille d'une fenêtre plus le temps de calcul nécessaire. En conséquence, il y a de fortes chances que notre grille reconstruite en temps réel soit complètement décalée par rapport à la séquence d'entre, notre estimation de Δ_p* n'étant jamais parfaite et le report d'erreurs étant très grand puisque relatif à la taille de toute une fenêtre plus le temps de calcul. Une manière de résoudre ce problème est

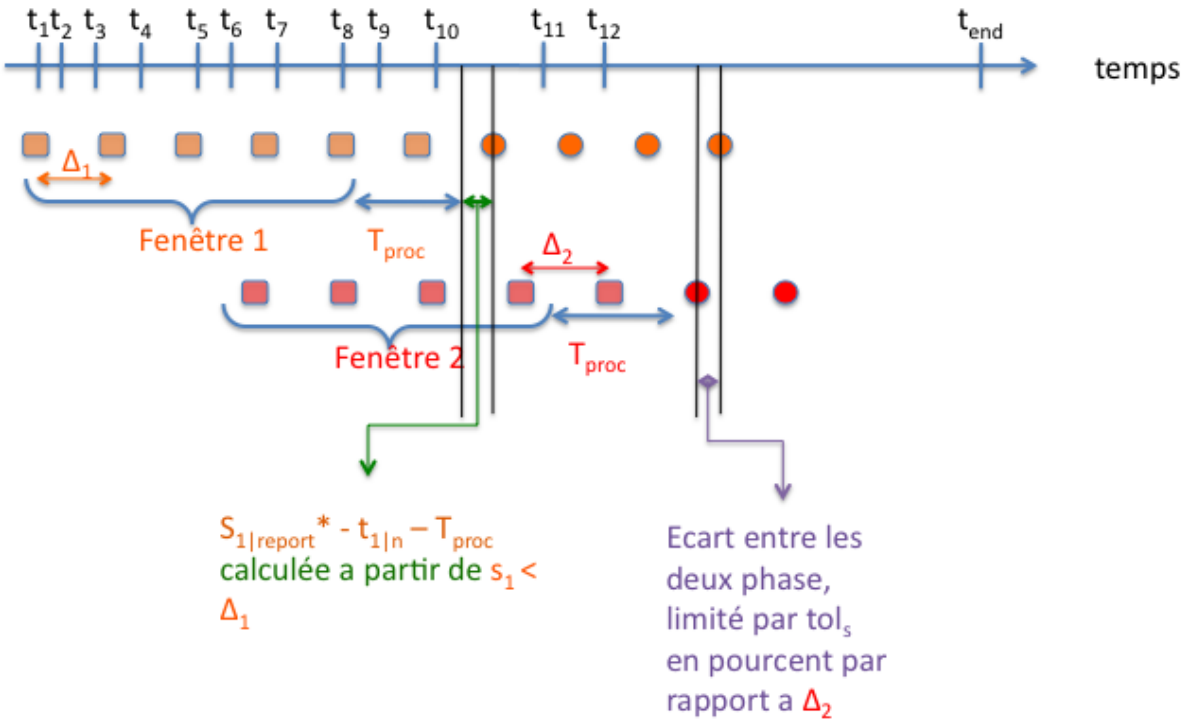

FIGURE 3.3 – Illustration du report de phase et de la comparaison des phases entre chaque fenêtre

de mettre à jour la phase par rapport à chaque événement.

3.4.2 Mise à jour à chaque événement : l'oscillateur de Larges et Jones

Principe

Pour reconstruire la phase, on propose ici une méthode basée sur l'oscillateur non linéaire auto-entretenu de Larges et Jones décrit dans [LJ99], et plus précisément la version exposée dans [LP02]. Cette oscillateur a pour base un oscillateur auto-entretenu interne qui modélise le processus de dynamique attentionnelle. Il génère ainsi une activité périodique liée à l'attente, que l'on considère comme une anticipation implicite sur la position d'un événement futur. On le couple avec une source rythmique externe et l'oscillateur tend alors à se synchroniser avec cette source. Pour ce faire, il compare la position temporelle de l'événement entrant issu de la source externe avec sa propre prédiction sur sa position, et s'ajuste ensuite en fonction de la différence entre les deux. Un tel oscillateur présente 3 paramètres que sont sa phase, sa période et son *Focus attentionnel*. Le *Focus attentionnel* représente la pondération sur la correction en fonction de l'écart entre la prédiction et l'événement entrant que va appliquer le système sur

ses deux autres variables que sont la période et la phase. Autrement dit, plus les prédictions de l'oscillateur dans le temps s'avèrent justes, moins le système s'adaptera à une erreur future, la considérant ainsi comme un bruit. En revanche, plus ses prédictions présentent des différences avec les événements entrants, plus le système sera réactif sur la variation de sa phase et de sa période interne et les modifiera rapidement.

Concrètement, c'est la phase $\Phi(t)$ qui permet de décrire quand un événement doit se produire. Elle s'exprime de manière récursive en considérant la phase $\hat{\Phi}_n$ et le pas précédent $\hat{\Delta}_n$ de l'oscillateur ainsi que la différence entre la position temporelle de l'événement entrant t_{n+1} et l'événement précédent t_n. Cette phase appartient à l'intervalle $[-0.5 \quad 0.5]$ et s'exprime par :

$$\Phi_{n+1} = \hat{\Phi_n} + \frac{t_{n+1} - t_n}{\hat{\Delta_n}} \tag{3.9}$$

Cette expression modélise la relation entre deux rythmes représentée par le couple $(\hat{\Phi}_n, \hat{\Delta_n})$ d'une part et $\{t_n\}$ d'autre part, mais ces deux rythmes reste découplés. Pour inclure la force qu'exerce le rythme externe sur l'oscillateur, il faut rajouter un terme qui correspond à une force de couplage. L'influence de cette force sera réglée par un paramètre η_Φ. On peut ainsi calculer une nouvelle phase $\hat{\Phi}_{n+1}$ de l'oscillateur qui prend en compte l'influence du rythme sur l'oscillateur par le biais de la valeur précédente de la phase. On corrige ainsi la phase Φ_{n+1} par un terme de force de couplage qui dépend d'une fonction F qui représente le focus attentionnel, et qui se calcul à partir de la phase précédemment prédit Φ_n ainsi qu'un paramètre κ_n représentant l'état du focus attentionnel.

$$\hat{\Phi}_{n+1} = \Phi_{n+1} - \eta_\Phi F(\Phi_{n+1}, k_{n+1}) \pmod{^{+0.5}_{-0.5}} \tag{3.10}$$

avec comme expression pour la fonction de focus attentionnel :

$$F(\Phi_n, k_n) = \frac{1}{2\pi \mathrm{e}^{k_n}} \mathrm{e}^{k_n \cos(2\pi\Phi_{n+1} \sin(2\pi\Phi_{n+1})} \tag{3.11}$$

Cette expression dérive en fait de l'expression d'une distribution de von-Mises présentée dans l'équation 3.3. De cette façon, la phase est mise à jour pour chaque événement entrant. Pour un événement entrant, le système met aussi à jour sa période $\hat{\Delta}_n$. Cette mise à jour s'effectue comme suit :

$$\hat{\Delta}_{n+1} = \hat{\Delta}_n(1 + \eta_\Delta F(\Phi_{n+1}, k_{n+1})) \tag{3.12}$$

Pour pouvoir calculer ces nouvelles valeurs de l'oscillateur non linéaire, il faut au préalable

avoir mis à jour l'état du focus attentionnel représenté par κ_n tel que :

$$\kappa_{n+1} = A^{-1}(r_{n+1}) \tag{3.13}$$

$$\text{avec} \quad r_{n+1} = r_n - \eta_r \cdot (r_n - \cos(2\pi\Phi_{n+1})) \tag{3.14}$$

$$\text{et} \quad A(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)} \tag{3.15}$$

Pour des raisons pratiques de recherche de solution (l'expression A n'a pas d'expression inverse analytique et on utilise une méthode d'interpolation pour retrouver la valeur de κ_{n+1}), on limite alors κ à l'intervalle [1 10].

Dans ces équations, les trois termes η_Φ, η_Δ et η_r sont des constantes qui modélisent l'influence du rythme extérieur sur l'oscillateur. Ces trois termes doivent vérifier la relation :

$$0 < \eta_r < \eta_\Delta < \eta_\Phi \leq 1 \tag{3.16}$$

Finalement, la mise à jour de l'oscillateur se déroule en 3 phases distinctes :
- Une phase de mesure, dans laquelle on évalue Φ_{n+1}
- Une phase de mise à jour, dans laquelle on évalue κ_{n+1}
- Une phase prédiction, dans laquelle on évalue $\hat{\Phi}_{n+1}$ et $\hat{\Delta}_{n+1}$

Cette mise à jour en 3 étapes est en fait l'équivalent d'un filtrage de Kalman étendu.

Application au problème

Pour appliquer cet algorithme à notre problème, il suffit de laisser de coté la mise à jour du pas Δ_n, et d'utiliser à la place le plus récent pas Δ_p* issu de l'algorithme 7 dès que celui-ci est disponible. On se contente ainsi de mettre à jour uniquement la phase de l'oscillateur, sa période étant mise à jour par nos algorithmes précédents.

La mise en place d'un tel système nécessite d'avoir imposé un fort BPM_{max} dans l'algorithme 2. En effet, si le BPM_{max} est trop faible, la période de l'oscillateur sera trop grande et pour une période correspondront plusieurs événements temporels. L'oscillateur mettra alors à jour plusieurs fois la phase pour un laps de temps d'une période et la variable κ ne sera plus représentative du focus attentionnel. L'avantage d'un système est que la phase est mise à jour pour chaque événement de manière quasi-instantanée. Ainsi, on a plus le problème de dé-synchronisation de la grille comme pour les deux méthodes précédentes. En revanche, comme la phase bouge pour chaque événement mais pas le pas, il est probable que notre grille reconstruite présente une variabilité au niveau de ses intervalles plus grande qu'avec les méthodes précédentes. Autrement dit, le tempo issu d'une telle grille n'est pas très constant.

Un version de l'algorithme qui met en place une telle méthode est l'algorithme 10

Algorithm 10 Une mise à jour de la phase $s*_{n+1}$ de la grille à l'aide de l'oscillateur non linéaire

Require: Δ_p* ; $\Phi_n : r_n \; t_n$; t_{n+1} ; η_Φ ; η_r

(1) Mesure

$$\Phi_{n+1} \leftarrow \Phi_n + \frac{t_{n+1} - t_n}{\Delta_p*}$$

(2) Mise à jour

$$r_{n+1} \leftarrow r_n - \eta_r \cdot (r_n - \cos(2\pi\Phi_{n+1}))$$

$$\kappa_{n+1} \leftarrow A^{-1}(r_{n+1})$$

(3) Prédiction

$$\Phi_{n+1} \leftarrow \Phi_{n+1} - \eta_\Phi F(\Phi_{n+1}, k_{n+1})$$

(4) Calcul de la nouvelle phase

if $\Phi_{n+1} < 0$ **then**

$s*_{n+1} \leftarrow t_{n+1} - \Phi_{n+1} \cdot \Delta_p*$

else

$s*_{n+1} \leftarrow t_{n+1} + (1 - \Phi_{n+1}) \cdot \Delta_p*$

end if

return $s*_{n+1}$; $\Phi_{n+1} : r_{n+1}$

Reste à choisir les valeurs des coefficients η. Dans la pratique, on choisit ces valeurs de manière empirique pour obtenir le meilleur résultat et on pose :

$$\eta_r = 0.85 \tag{3.17}$$

$$\eta_\Phi = 1 \tag{3.18}$$

Chapitre 4

Resultats obtenus

4.1 Mise en place d'une mesure simple

La manière la plus instinctive de tester nos différents méthodes et paramètres est l'écoute de la grille obtenue. Cette méthode est cependant très objective et relativement peu précise, les différences entre les différentes méthodes étant en effet relativement faibles. Dans le but d'effectuer la comparaison d'une manière un peu plus précise, on souhaite mettre en place une mesure nous permettant de les comparer numériquement. Pour ce faire, on se base sur la définition de la mesure Φ de l'équation 2.2. On définit alors pour une séquence d'entrée $\{t_k\}_{\substack{k\in\mathbb{N}\\1\leq i\leq n}} \subset \mathbb{R}$ et une grille de taille P $\{\mathcal{G}_j\}_{\substack{j\in\mathbb{N}\\1\leq i\leq P}} \subset \mathbb{R}$ la mesure :

$$M(\{t_k\},\{\mathcal{G}_j\}) = \sum_{k=\alpha}^{n} \min_j |t_k - \mathcal{G}_j| \tag{4.1}$$

avec α le premier entier k tel que $t_k > t_{1|n} + T_{proc}$, $t_{1|n}$ représentant le dernier événement de la première fenêtre et T_{proc} le temps de calcul.

Remarquons que pour que la mesure soit utilisable dans le cadre d'une comparaison, il faut que la séquence d'entrée soit identique, et que le BPM_{max} ne change pas d'un test à l'autre.

On dispose en tout de 4 options différentes auquel on rajoute la variation des paramètres $(\mathbf{R},\mathbf{Q})$ pour nos tests que sont :

- Taille de la fenêtre : n constant ou t_{size} constant
- Construction du vecteur d'état : Estimation par noyau ou Interpolation (on ne retiendra que l'interpolation linéaire pour nos tests)
- Normalisation du vecteur d'état : activé ou non
- Méthodes de reconstruction de la phase : Méthode de Test, Méthode par balayage (sur laquelle on maintiendra une tolérance sur la phase de 50%), et Oscillateur non linéaire.

Pour tout les tests, on testera 4 couples pour $(\mathbf{R},\mathbf{Q})$ qui sont issues des considérations dans la section 3.3. On prendra donc comme valeurs tests :

- $(\mathbf{R},\mathbf{Q}) = (10\cdot\frac{\Delta_{min}}{8}\mathbf{I}_{N_f}, \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}) = (10\sigma_{ref}, \sigma_{ref})$
- $(\mathbf{R},\mathbf{Q}) = (\frac{\Delta_{min}}{8}\mathbf{I}_{N_f}, \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}) = (\sigma_{ref}, \sigma_{ref})$

– $(\mathbf{R}, \mathbf{Q}) = (10 \cdot \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}, 0.1 \cdot \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}) = (10\sigma_{ref}, 0.1\sigma_{ref})$
– $(\mathbf{R}, \mathbf{Q}) = (\frac{\Delta_{min}}{8}\mathbf{I}_{N_f}, 0.1 \cdot \frac{\Delta_{min}}{8}\mathbf{I}_{N_f}) = (\sigma_{ref}, 0.1\sigma_{ref})$

4.2 Cas de séquence d'entrée construites

On considère ici des séquences d'entrée que l'on construit, et donc que l'on maitrise parfaitement, pour tester nos différentes méthodes. On définit un certain nombre de paramètres constants tels que :

– $BPM_{max} = 700$
– $N_f = 512$
– $N_obs = 200$
– $N_{phase} = 1024$
– $N_{ker} = 10$
– $T_{proc} = 1$

Les autres paramètres ayant déjà été définis dans le rapport.

4.2.1 Grille bruité

On considère ici pour séquence d'entrée $t_k = k\Delta + e_k$ avec $e_k \sim \mathcal{N}(0, \sigma)$. On prend pour paramètre $\Delta = 0.1s$ et $\sigma = 0.2s$. Cela correspond à une grille régulière fortement bruitée par bruit gaussien centrée en zéro. Puisque la grille est régulière, il est inutile de tester la différence entre n et t constant pour la taille de fenêtre. On gardera des fenêtres de 20 événements pour une durée totale de 30 secondes. De même, puisque l'on connait le pas minimum de la grille, la normalisation s'avère inutile puisque l'on se place dans le bon intervalle de recherche. On obtient les résultats visibles sur le tableau 4.1.

Méthodes / $(\mathbf{R}, \mathbf{Q})$	Estimation par noyau			Interpolation		
	Test	Bal	L&J	Test	Bal	L&J
$(10\sigma_{ref}, \sigma_{ref})$	5.466	5.529	4.390	5.556	5.560	4.366
$(\sigma_{ref}, \sigma_{ref})$	5.375	5.254	4.732	5.824	5.761	4.386
$(10\sigma_{ref}, 0.1\sigma_{ref})$	5.378	5.362	4.389	5.393	5.346	4.379
$(\sigma_{ref}, 0.1\sigma_{ref})$	5.321	5.323	4.379	5.435	5.451	4.355

FIGURE 4.1 – Tableau présentant les résultats de la mesure pour le cas d'une grille parfaite bruitée par un bruit gaussien en entrée. Les mentions 'Test', 'Bal' et 'L&J' représentent respectivement les méthodes de test de phase disponible, balayage de phase et oscillateur non linéaire pour la reconstruction de la phase. Les trois meilleures mesures sont respectivement les cellules rouge, orange et jaune. Les deux moins bonnes mesures sont respectivement les cellules bleu et cyan.

Commentaires sur les résultats

On remarque que les meilleures mesures sont obtenues dans le cas d'une construction du vecteur d'état par interpolation couplé à une reconstruction de phase par la méthode de l'oscillateur linéaire. Ces valeurs ne sont pas significativement différentes entre elles pour différentes valeurs du couple $(\mathbf{R}, \mathbf{Q})$ mais on remarque néanmoins que les valeurs sont meilleures pour lorsque les deux paramètres ont un rapport de 10. Les moins bonnes mesures sont obtenues lorsque les valeurs de ($\mathbf{R}$ et de $\mathbf{Q}$) sont identiques. Globalement, on constate que c'est la méthode de l'oscillateur non linéaire qui fait la différence pour différencier les mesures. Il est difficile de différencier les méthodes d'estimation par noyau ou d'Interpolation pour la reconstruction du vecteur d'état.

L'oscillateur a un poids important dans ce cas car la grille est bruitée, et l'oscillateur permet ainsi de réagir très vite pour corriger la phase. Les méthodes de balayage et de test de phase souffrent du report de l'erreur commise sur le pas lors de l'estimation. Plusieurs graphiques permettant de visualiser la plus mauvaise mesure et la meilleure sont disponibles sur les figures 4.2 et 4.3.

On constate sur ces deux graphiques que le filtrage de kalman réussit dans tous les cas à retrouver la valeur du pas original de la grille.

4.2.2 Saut de tempo

On considère ici pour entrée une séquence régulière de durée 30s qui présente un saut de tempo de $\Delta_1 = 0.1s$ à $Delta_2 = 0.15s$ en son milieu, soit un ralentissement brutal. On teste les mêmes paramètres que précédemment en rajoutant cette fois le paramètre de taille de fenêtre. Puisque le pas change. On testera ainsi $n = 20$ et $t_{size} = 2.5s$ On obtient les résultats visibles sur le tableau 4.4.

Méthodes / $(\mathbf{R}, \mathbf{Q})$	Fenêtre n constant						Fenêtre t_{size} constant					
	Estimation noyau			Interpolation			Estimation noyau			Interpolation		
	Test	Bal	L&J	Test	Bal	L&J	Test	Bal	L&J	Test	Bal	L&J
$(10\sigma_{ref}, \sigma_{ref})$	6.198	5.734	3.815	0.921	0.919	0.664	5.566	5.722	4.636	0.847	0.849	6.014
$(\sigma_{ref}, \sigma_{ref})$	6.230	6.051	4.243	0.841	0.880	6.167	5.444	5.683	4.024	0.775	0.812	6.451
$(10\sigma_{ref}, 0.1\sigma_{ref})$	5.589	5.441	3.590	1.458	1.495	1.162	6.182	6.061	5.095	1.319	1.395	1.112
$(\sigma_{ref}, 0.1\sigma_{ref})$	6.114	5.797	3.933	0.921	0.919	0.664	5.046	5.335	4.281	0.847	0.849	6.014

FIGURE 4.4 – Tableau présentant les résultats de la mesure pour le cas d'une grille parfaite présentant un saut de tempo. Les mentions 'Test', 'Bal' et 'L&J' représentent respectivement les méthodes de test de phase disponible, balayage de phase et oscillateur non linéaire pour la reconstruction de la phase. Les trois meilleures mesures sont respectivement les cellules rouge, orange et jaune. Les deux moins bonnes mesures sont respectivement les cellules bleu et cyan.

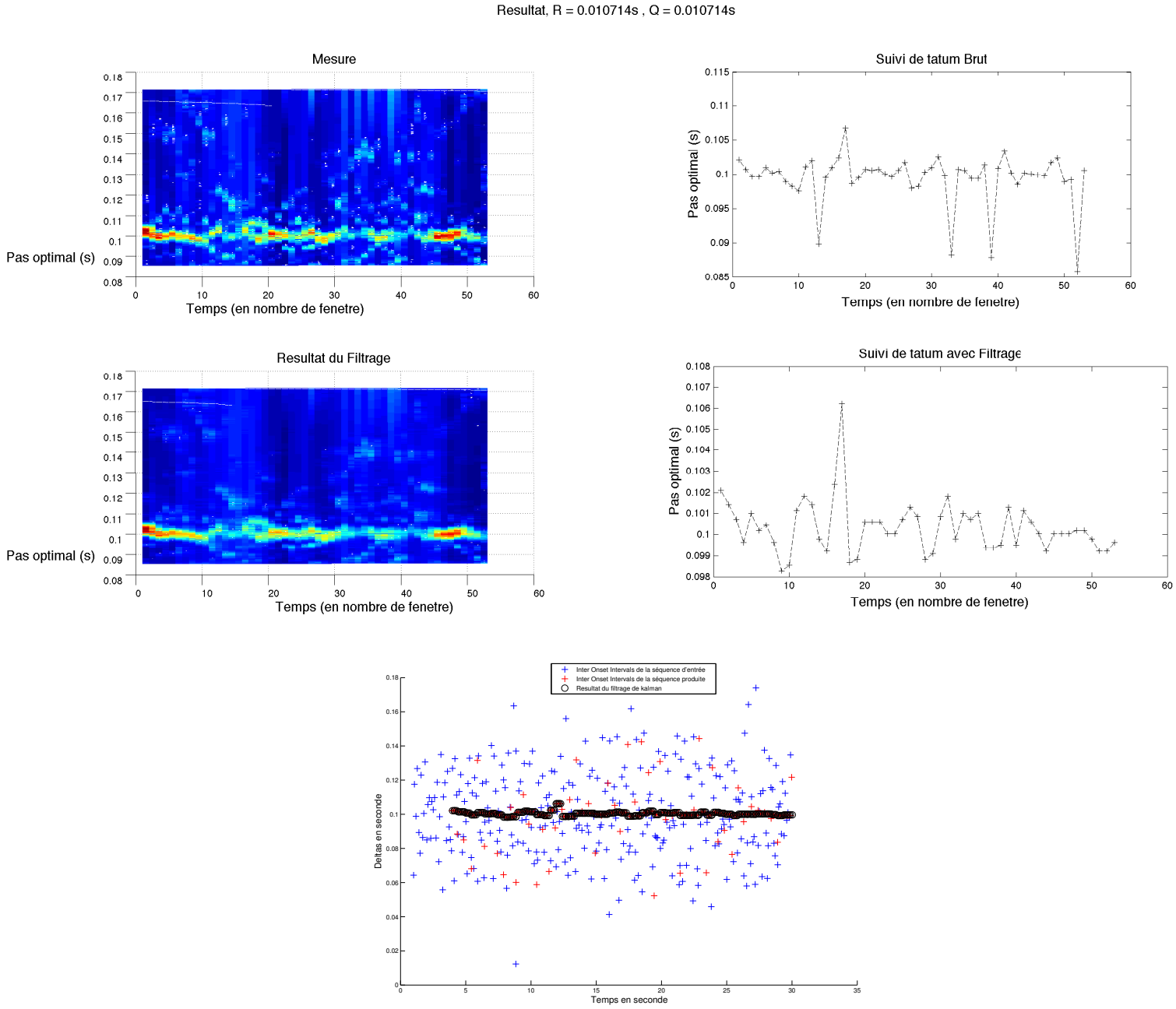

FIGURE 4.2 – Figures relatives à la plus mauvaise mesure du tableau 4.1. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure de bas représente les intervalles de la séquence d'entrée (point bleu), les intervalles prévus par le filtrage de kalman (cercle noir) et ceux que l'on obtient effectivement après reconstruction de la phase. On constate par ailleurs que la grille n'est pas régulière à cause de l'oscillateur non linéaire.

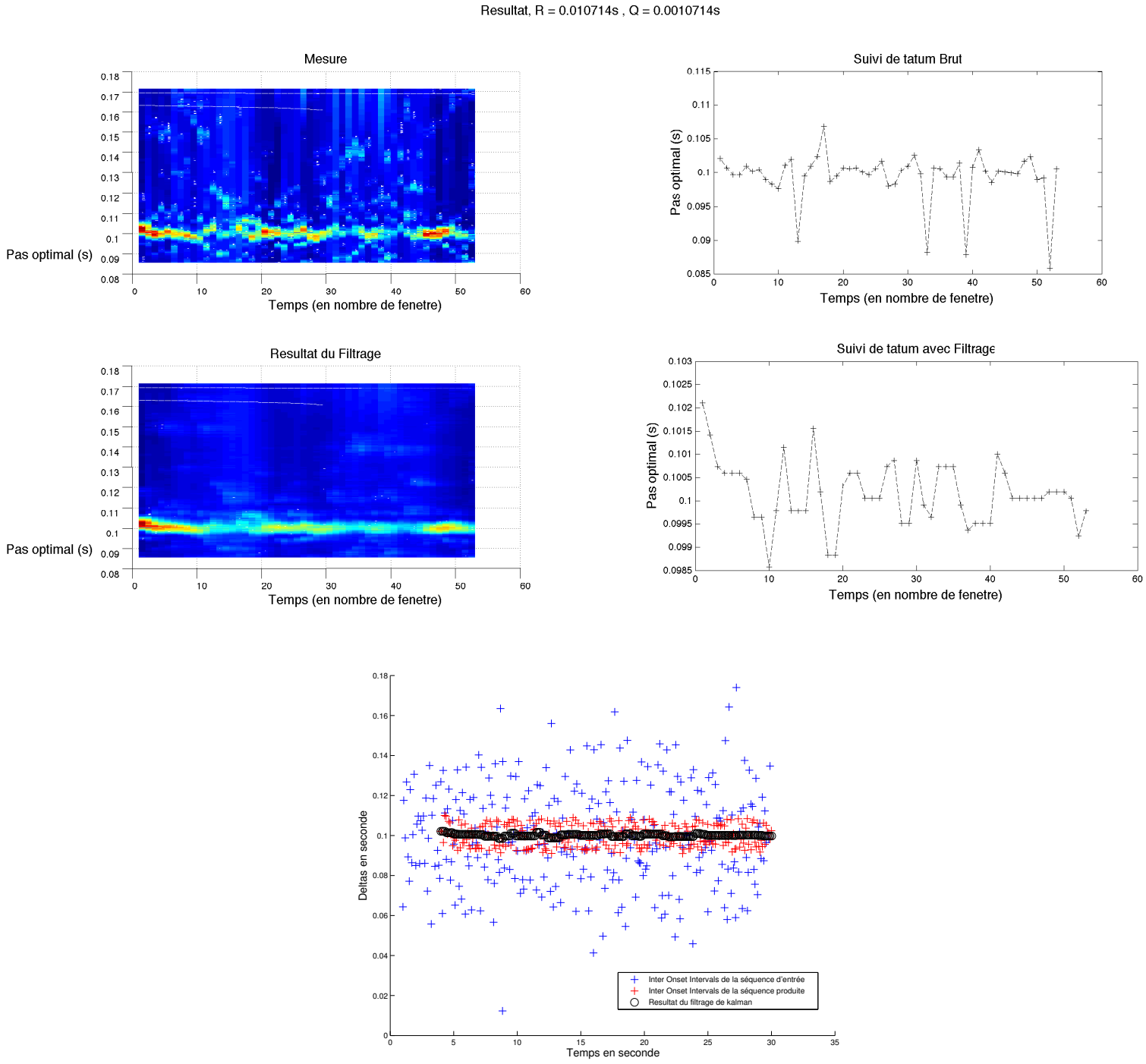

FIGURE 4.3 – Figures relatives à la meilleure mesure du tableau 4.1. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure de bas représente les intervalles de la séquence d'entrée (point bleu), les intervalles prévus par le filtrage de kalman (cercle noir) et ceux que l'on obtient effectivement après reconstruction de la phase.

Commentaires sur les résultats

On remarque qu'encore une fois, les meilleurs résultats proviennent de l'oscillateur non-linéaire. Cependant pour cette fois, c'est aussi avec cette technique que l'on obtient le plus mauvais résultat. Cela est dû à un effet visible sur la figure 4.5 où l'oscillateur n'arrive pas à se re-synchroniser avec la pulsation car sa phase est en permanence en opposition de phase. Cela est dû au fait que le saut de tempo amène la phase courante en parfaite opposition de phase. L'oscillateur ne parvient pas a se re-synchroniser car dans un tel cas pour l'oscillateur original décrit par Large, c'est la combinaison de la mise à jour du pas et de la phase qui le lui permet. Dans ce cas précis, la phase reste bloquée a la valeur 0.5, qui correspond à un "noeud" de la fonction du focus attentionnel. En conséquence, l'oscillateur ne met pas sa phase à jour. Cela illustre bien une des limitations de ce modèle.

On constate aussi que dans ce cas, la méthode d'interpolation semble plus efficace, car l'estimateur par noyau entraine un changement trop tardif dans le pas, ne faisant pas apparaître suffisamment rapidement le second "pic". Les algorithmes n'arrivent plus à retrouver la phase que ce soit avec la méthode de l'oscillateur pour les raisons évoquées précédemment, ou bien avec les méthodes de test qui n'ont plus à disposition la phase optimale. Remarquons que si l'on avait pas maintenu une tolérance de changement sur la phase à 50% dans le cas du balayage, on aurait surement obtenu des résultats comparables à ceux de l'oscillateur dans ce cas. Cette tolérance sur les sauts de phase, si elle peut trouver une justification perceptive, est un facteur limitant ici.

La différence entre une taille de fenêtre définie par son nombre d'événements, ou par sa durée n'est pas significative dans ce cas pour pouvoir tirer une conclusion, bien que la définition temporelle semble légèrement plus efficace. Les figures relatives à la meilleure mesure sont présentés sur la figure 4.6

Les meilleures mesures sont encore une fois obtenues lorsque le couple $(\mathbf{R}, \mathbf{Q})$ a un rapport de 10. En revanche, les mesures ne sont pas mauvaises lorsque ces deux paramètres sont identiques, car cela correspond à une plus grande variation de tempo autorisée par le filtrage de kalman.

4.2.3 Décélération de tempo

On considère ici pour entrée une séquence régulière de durée 30s qui est accélérée linéairement d'un facteur d'accélération $\theta = {}^1/_2$ sur l'ensemble de sa durée. C'est-à-dire une décélération d'un facteur 2 entre le début et la fin de la séquence. On teste les mêmes paramètres que précédemment. On obtient les résultats visibles sur le tableau 4.7

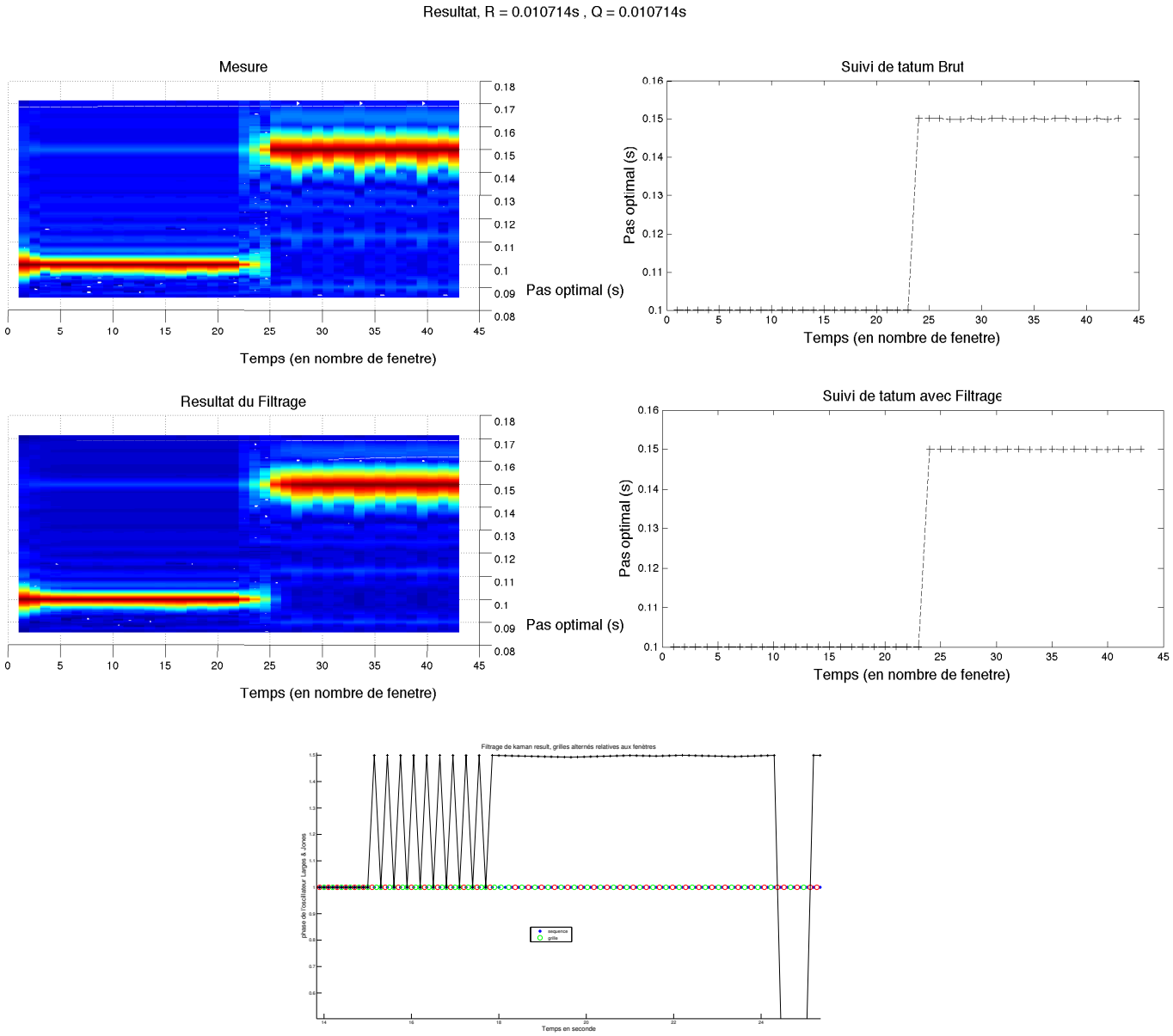

FIGURE 4.5 – Figures relatives à la plus mauvaise mesure du tableau 4.4. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure du bas est un zoom sur la séquence d'entrée sur laquelle on a superposé la grille issue de la reconstruction. Est également présente la phase Φ de l'oscillateur en noir. On constate que l'oscillateur n'arrive pas a ré-accrocher sur la bonne phase et reste ainsi en permanence en opposition de phase, d'où la plus mauvaise mesure

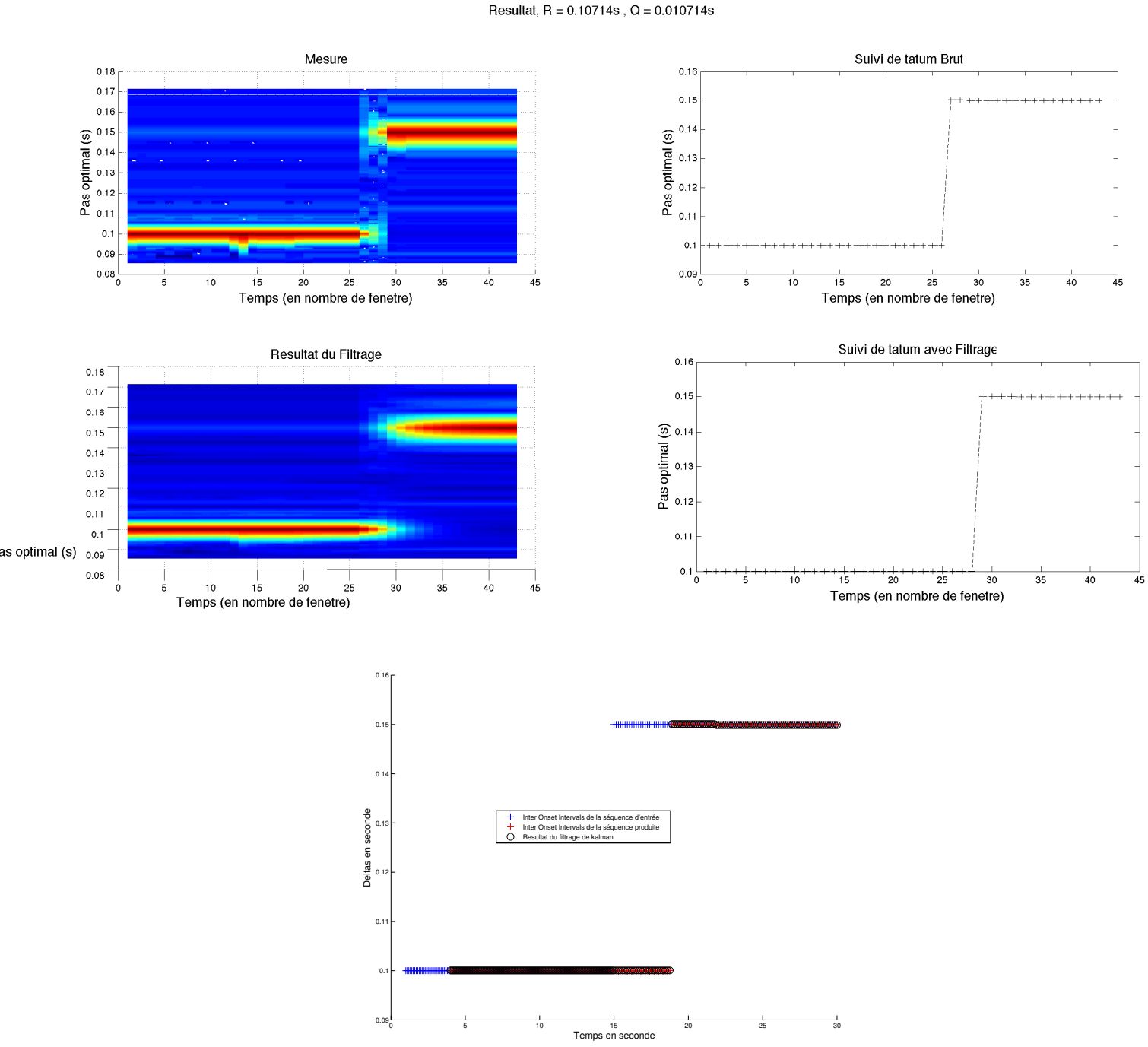

FIGURE 4.6 – Figures relatives à la meilleure mesure du tableau 4.4. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure du bas représente les intervalles de la séquence d'entrée (point bleu), les intervalles prévus par le filtrage de kalman (cercle noir) et ceux que l'on obtient effectivement après reconstruction de la phase.

$(\mathbf{R}, \mathbf{Q})$ \ Méthodes	Fenêtre n constant						Fenêtre t_{size} constant					
	Noyaux			Interpolation			Noyaux			Interpolation		
	Test	Bal	L&J	Test	Bal	L&J	Test	Bal	L&J	Test	Bal	L&J
$(10\sigma_{ref}, \sigma_{ref})$	4.957	4.945	5.259	9.692	9.681	4.434	7.309	7.272	4.819	11.07	11.04	4.110
$(\sigma_{ref}, \sigma_{ref})$	14.04	14.03	3.791	13.77	13.77	3.801	14.12	14.11	3.575	14.42	14.42	3.592
$(10\sigma_{ref}, 0.1\sigma_{ref})$	11.27	11.24	8.108	5.573	5.522	6.599	7.551	7.696	7.861	5.361	5.422	6.417
$(\sigma_{ref}, 0.1\sigma_{ref})$	5.203	5.235	5.225	10.07	10.09	4.381	7.398	7.368	4.778	11.23	11.22	4.091

FIGURE 4.7 – Tableau présentant les résultats de la mesure pour le cas d'une grille parfaite désaccélérée linéairement. Les mentions 'Test', 'Bal' et 'L&J' représentent respectivement les méthodes de tests de phase disponible, balayage de phase et oscillateur non linéaire pour la reconstruction de la phase. Les trois meilleures mesures sont respectivement les cellules rouge, orange et jaune. Les deux moins bonnes mesures sont respectivement les cellules bleu et cyan.

Commentaires sur les résultats C'est encore une fois l'oscillateur non linéaire qui remporte les meilleurs résultats. Il peut cette fois corriger sa phase rapidement et ainsi contrer la prédiction du pas optimal qui reste stable sur une période donnée, alors que le pas réel varie avec chacun des nouveaux événements. Sans surprise, c'est donc les méthodes de balayage et de test de phase qui donnent les moins bons résultats. Cela s'illustre très bien sur la figure 4.8, où l'erreur de phase dûe au retard du filtrage crée des intervalles qui n'ont plus de sens sur la séquence de sortie.

C'est cette fois l'estimateur par noyaux par rapport à la méthode d'interpolation qui possède les meilleurs résultats. Cela est probablement due à un biais positif de cette estimateur.

Dans ce cas, bien que les meilleures mesures se trouvent dans le cas où une fenêtre est définie par un paramètre temporel, les résultats semblent meilleurs en moyenne pour une fenêtre définie en terme de nombre d'événements.

Notons enfin que les résultats sont les meilleurs lorsque le couple $(\mathbf{R}, \mathbf{Q})$ possède des valeurs identiques pour ses deux paramètres. Le couple $(\mathbf{R}, \mathbf{Q})$ correspond à la plus mauvaise valeur en moyenne car cela correspond à une grande rigidité dans la variation de tempo, qui n'est pas adapté à une décélération de celui-ci. Si les paramètres $\mathbf{R} = \mathbf{Q}$ impliquent les meilleurs résultats, ils contiennent aussi les plus mauvais. Cette paramètrisation s'avère peu stable, comme on pouvait s'y attendre.

4.3 Cas réel

On s'intéresse maintenant à des morceaux réels. Le premier exemple est un morceau issu de studio, qui possède une pulsation parfaite en théorie, le second est issu d'un enregistrement d'une performance live. Puisque c'est l'oscillateur qui semble donner les meilleurs résultats, on ne garde que cette méthode dans la suite de nos tests. On testera par contre le paramètre de

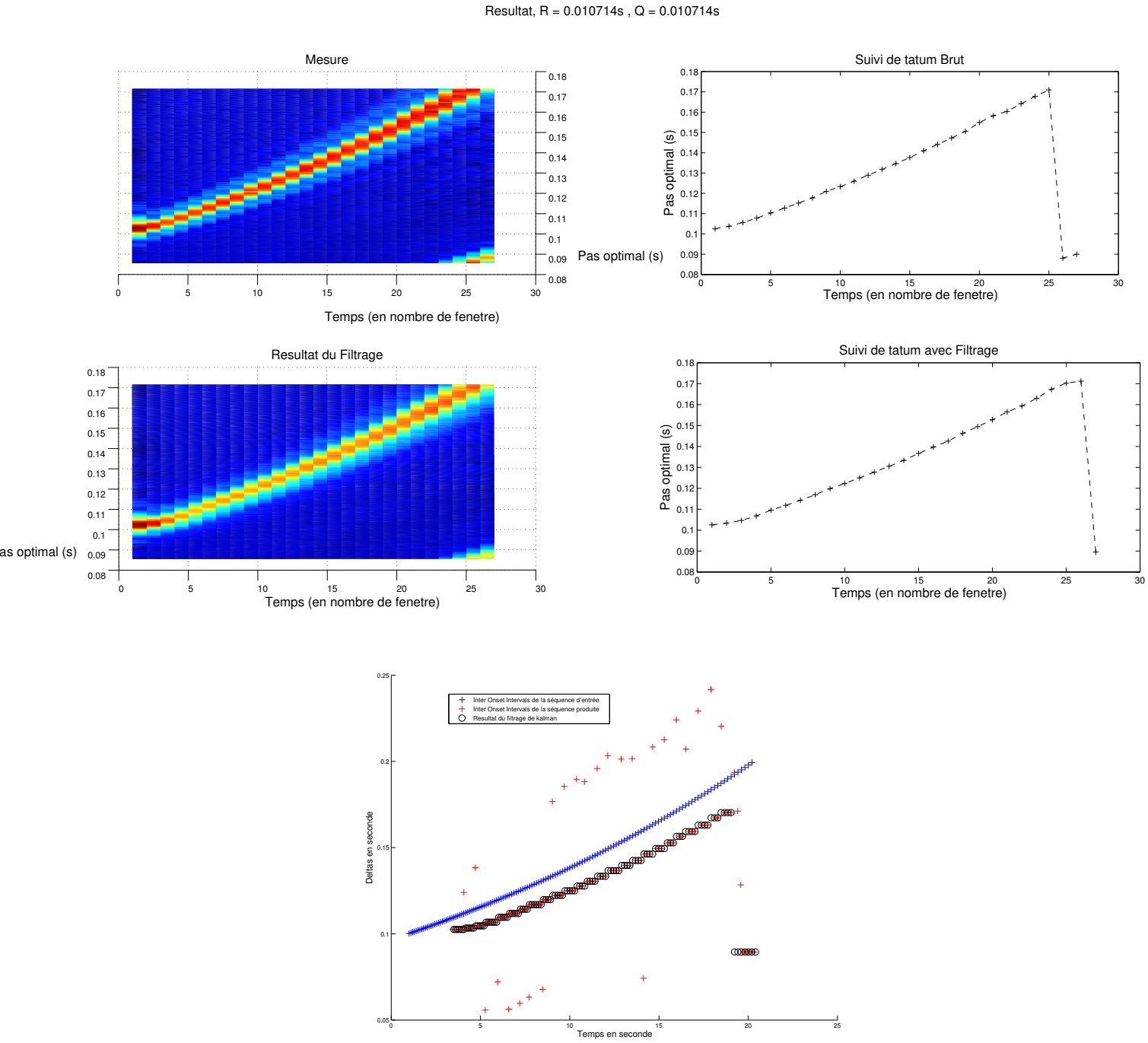

FIGURE 4.8 – Figures relatives à la plus mauvaise mesure du tableau 4.7. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure du bas représente les intervalles de la séquence d'entrée (point bleu), les intervalles prévu par le filtrage de kalman (cercle noir) et ceux que l'on obtient effectivement après reconstruction de la phase.

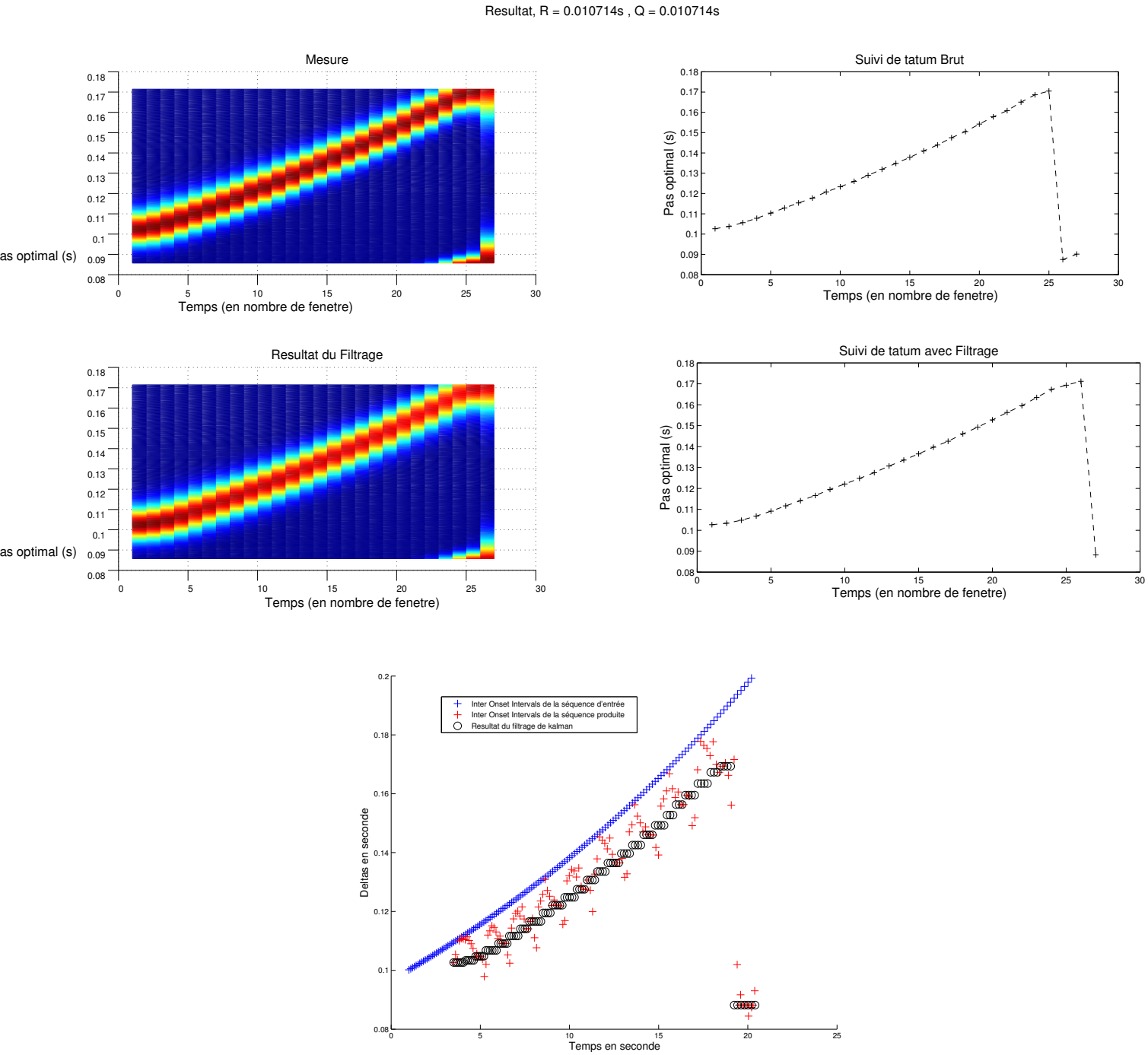

FIGURE 4.9 – Figures relatives à la meilleure mesure du tableau 4.7. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. La figure de bas représente les intervalles de la séquence d'entrée (point bleu), les intervalles prévu par le filtrage de kalman (cercle noir) et ceux que l'on obtient effectivement après reconstruction de la phase.

normalisation des vecteurs d'état, puisque l'on ne sait pas dans quel intervalle chercher le tactus. On change aussi la valeur du BPM_{max} à 600bpm, ainsi que la taille de la fenêtre temporelle t_{size} qui est alors prise à 3 secondes. On justifie ces choix par la valeur moyenne des intervalles des morceaux. On obtient les résultats visibles sur le tableau 4.10

4.3.1 Cas studio (pulse parfaite)

Le morceaux considéré est une interprétation de Miles Davis (*'four smiles'*), issue de la transcription d'une partition en MIDI. C'est en conséquence un morceau qui est sensé avoir une pulsation parfaite. On récupère le canal de la batterie du fichier midi pour séquence d'entrée. Ce morceau, bien que possédant une pulsation théorique parfaite, est d'une bonne complexité rythmique et a la particularité de présenter des passages à la croche et d'autres au triolet.

Méthodes / $(\mathbf{R}, \mathbf{Q})$	Fenêtre n constant				Fenêtre t_{size} constant			
	Noyaux		Interpolation		Noyaux		Interpolation	
		Norm		Norm		Norm		Norm
$(10\sigma_{ref}, \sigma_{ref})$	11.95	12.058	11.87	11.87	11.80	11.77	11.74	11.54
$(\sigma_{ref}, \sigma_{ref})$	12.51	12.11	12.55	12.65	13.41	13.38	13.02	12.77
$(10\sigma_{ref}, 0.1\sigma_{ref})$	11.44	11.49	10.77	10.77	11.81	11.87	10.85	10.88
$(\sigma_{ref}, 0.1\sigma_{ref})$	11.74	11.85	11.87	11.87	11.86	11.82	11.71	11.53

FIGURE 4.10 – Tableau présentant les résultats de la mesure pour morceau *four smiles* de Miles Davis issu d'une transcription de partition en MIDI. Les trois meilleures mesures sont respectivement les cellules rouge, orange et jaune. Les deux moins bonnes mesures sont respectivement les cellules bleu et cyan.

Commentaires sur les résultats :

Les meilleurs résultats sont obtenus ici dans le cadre de fenêtre défini par leur nombre d'éléments, couplés avec le couple $(\mathbf{R}, \mathbf{Q})$ qui possède le plus grand écart, correspondant à la plus grande rigidité possible concernant la variation de tempo. C'est donc logique que l'on obtienne les meilleurs résultats avec un tel couple. Les résultats sont toujours meilleurs si on prend la méthode d'interpolation. Cela est dû au fait que l'estimateur par noyau est biaisé. Et c'est d'ailleurs de manière assez attendue que l'on obtient les plus mauvais résultats pour un couple $(\mathbf{R}, \mathbf{Q})$ tel $\mathbf{R} = \mathbf{Q}$ couplé à une estimation par noyaux.

L'avantage de définir les fenêtres en termes d'éléments constants prend tout son sens ici puisque on s'affranchit du manque de précision dû à un "break" par exemple, au contraire de période trop intenses au niveau rythmique qui ne représentent plus la grille. La mesure est donc plus stable dans ce cas.

La normalisation de la mesure n'a pas un effet significatif pour pouvoir être interprétée.

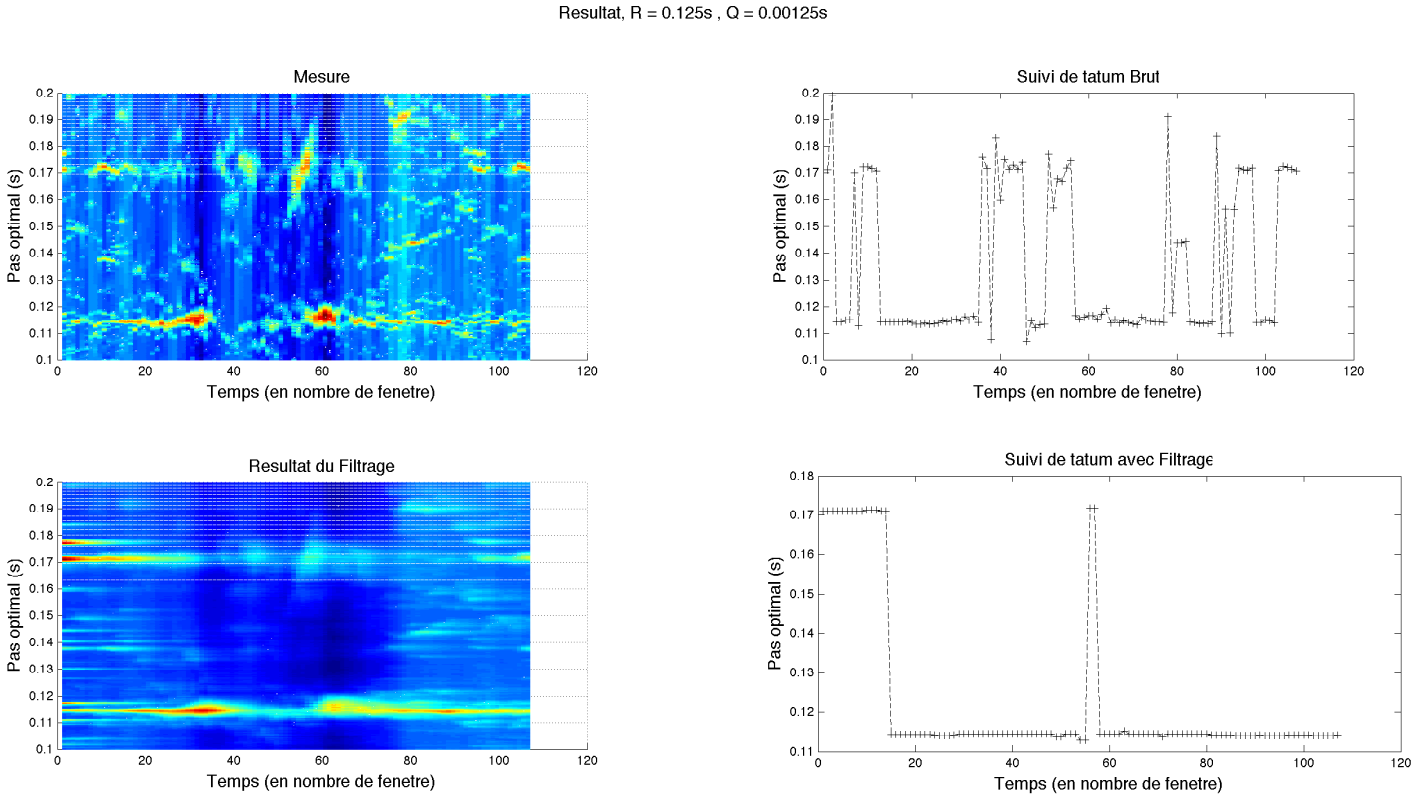

FIGURE 4.11 – Figures relatives à la meilleure mesure du tableau 4.10. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état. On constate que le pas optimal a tendance à osciller entre la valeur de la croche et celle du triolet, qui n'ont pas un rapport entier.

4.3.2 Cas improvisé

Le morceau considéré est un extrait interprété par Bernard Lubat, lors d'une séance de travail avec le logiciel *ImproteK* qu'ont mené Jérôme Nika et Marc Chemillier avec lui en 2013. C'est une improvisation qui à la particularité de présenter un changement brutal de tempo d'un facteur 2 en plein milieu de l'extrait. Ce changement doit passer inaperçu pour nous, puisque l'on considère un espace replié sur $[\Delta_{min} \quad]$. Cette improvisation "swing" énormément et peut passer du 3 temps au 4 temps sur des périodes de quelques secondes seulement. On obtient les résultats visibles sur le tableau 4.12. Cet extrait représente l'aboutissement souhaité du travail de Gilbert Nouno (application aux musiques improvisées) et donc par extension celui qu'on a mené dans ce stage.

Méthodes / $(\mathbf{R}, \mathbf{Q})$	Fenêtre n constant				Fenêtre t_{size} constant			
	Noyaux		Interpolation		Noyaux		Interpolation	
		Norm		Norm		Norm		Norm
$(10\sigma_{ref}, \sigma_{ref})$	11.23	11.07	11.25	11.27	11.87	12.48	11.41	11.34
$(\sigma_{ref}, \sigma_{ref})$	10.78	10.80	11.01	10.89	12.78	12.35	11.41	11.59
$(10\sigma_{ref}, 0.1\sigma_{ref})$	11.40	10.74	10.66	10.98	12.04	12.02	12.19	11.56
$(\sigma_{ref}, 0.1\sigma_{ref})$	11.24	11.09	11.25	11.27	11.75	12.34	11.55	11.44

FIGURE 4.12 – Tableau présentant les résultats de la mesure pour le cas d'un extrait improvisé interprété par Bernard Lubat, en entrée. Les trois meilleures mesures sont respectivement les cellules rouge, orange et jaune. Les deux moins bonnes mesures sont respectivement les cellules bleu et cyan.

Commentaires sur les résultats :

On retrouve les meilleurs résultats pour le couple $(10\sigma_{ref}, 0.1\sigma_{ref})$, associé à une interpolation et un paramètre de fenêtre de taille n constante. Pour cette improvisation, prendre une fenêtre constante dans le temps donne les plus mauvais résultats, c'est dû à la disparité et le saut de tempo de l'extrait. Cette fois, l'estimateur par noyaux n'est pas beaucoup désavantagé par rapport à la méthode d'interpolation. La normalisation a encore une fois un poids trop faible pour être interprété. Il est difficile d'en dire plus sur cet extrait en considérant les résultats du tableau 4.12.

Une visualisation du suivi de tempo avant puis après filtrage est visible sur la figure **??**. On constate que la mesure est très disparate, mais que le filtrage arrive malgré tout à faire ressortir uniquement deux pulsations, dont le rapport laisse penser qu'il s'agit de la croche et du triolet. Il semble donc que nos algorithmes arrivent effectivement à retrouver une réelle métrique du morceau. De plus, une écoute de la grille reconstruite se trouve être satisfaisante et correspond à la battue que pourrait faire un musicien. On est donc satisfait de ce résultat.

A l'heure actuelle, on considère donc que les meilleurs résultats sont obtenus pour les paramètres :

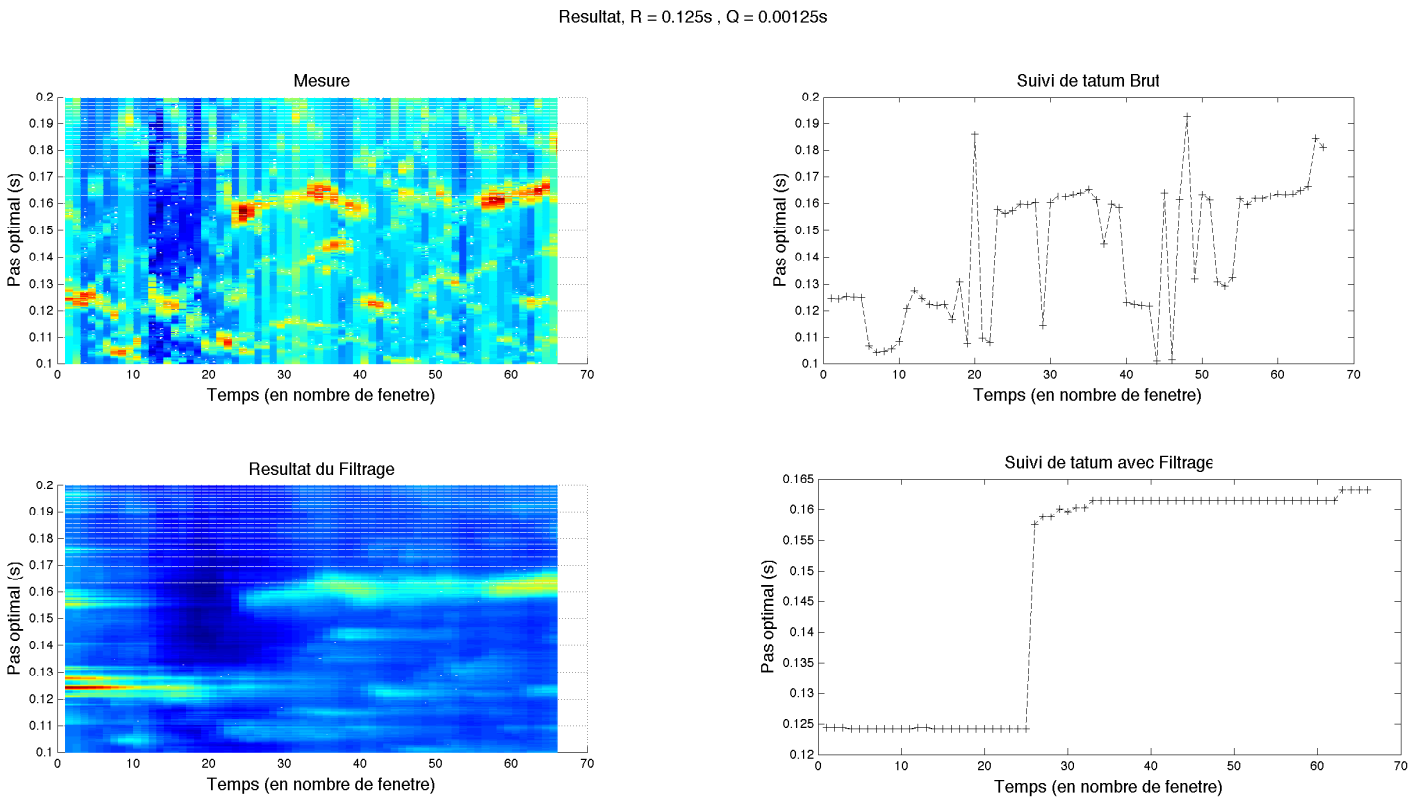

FIGURE 4.13 – Figures relatives à la meilleure mesure du tableau 4.12. La figure du haut représente une comparaison des résultats avant et après filtrage de Kalman. On peut ainsi voir le suivi en temps réel du vecteur d'état avant (en haut à gauche) et après (en bas à gauche) filtrage. Plus la couleur tend vers le rouge, plus la valeur de la mesure σ' est élevée. La partie gauche de cette figure représente le pas optimal (en haut avant et en bas après filtrage) issu du vecteur d'état.

- Paramètre du filtre de kalman : $(\mathbf{R}, \mathbf{Q}) = (10\sigma_{ref}, 0.1\sigma_{ref})$
- Méthode de construction du vecteur d'état : Interpolation
- Méthode de reconstruction de la grille : Oscillateur non linéaire
- Paramètrisation des fenêtres : En terme de nombre d'éléments constants.
- Normalisation : Peu utile, autant garder la notion de comparaison entre chaque fenêtre.

Chapitre 5

Conclusion et Perspectives

Durant ces 5 mois de stage, il s'agissait de se familiariser avec la méthode d'extraction de tempo de Gilbert Nouno afin de pouvoir la tester dans un contexte temps réel. Pour cela, on a commencé par analyser et s'approprier la méthode autant par l'intermédiaire du formalisme mathématique que par l'expérience. En effet, au début du stage il n'y avait pas de code disponible de la méthode, et l'implémentation de celle-ci nous a mené à mieux comprendre son principe et fonctionnement. Ce premier travail nous a aussi amené à proposer aussi des extensions de la méthode en se concentrant sur l'objectif final du stage qu'est le suivi de tatum en temps réel, que des nouveaux formalismes à l'image de l'annexe A lorsque ceux présentés dans la thèse de Nouno n'étaient pas considérés comme suffisamment convaincants. Cependant, la véritable valeur ajoutée de notre travail a été de fournir plusieurs traitements "post-mesure" (au sens de Nouno) pour permettre l'exploitation en temps réel du formalisme étudié. Ces traitements passent aussi bien par la construction de variables exploitables pour comparer différentes mesures entre elles que l'ajout de traitement statistique d'apprentissage automatique ou encore de développement de méthodes permettant une reconstruction en temps réel de la grille du tatum. Plusieurs pistes ont étés abordées, et si nombre d'entre elles ont été abandonnés fautes de résultats, il n'en reste pas moins plusieurs méthodes fournissant des résultats acceptables pour le suivi de tatum et la reconstruction de la grille.

Pour développer ces méthodes, il a alors été nécessaire de faire un état de l'art pour s'inspirer des méthodes existantes dans le suivi de tempo, mais aussi et surtout faire preuve d'imagination car le formalisme posé par Nouno n'a pas encore à notre connaissance été repris dans la littérature. On s'est ainsi concentré sur le développement de plusieurs méthodes que l'on a testées dans un cadre de temps réel simulé plutôt que sur le développement d'un programme temps réel comme il avait été annoncé en objectif du stage. De même, si on a présenté et comparé plusieurs façons de faire, le temps nous a manqué pour pouvoir comparer nos algorithmes à ceux de la littérature. En conséquence, on est conscient du manque d'ancrage dans la communauté de suivi de tempo sur ces méthodes. Cependant, puisqu'il résulte de ce stage un script final commenté dans le langage MATLAB et dont toutes les méthodes présentées dans ce rapport sont accessibles à haut niveau (concrètement, il suffit d'activer différentes options relatives aux différentes méthodes), il sera très facile d'effectuer une comparaison dans un travail ultérieur.

Les résultats de ce stage sont donc positifs, mettant à disposition plusieurs applications concrètes de la méthode de Gilbert Nouno, et montre que celle-ci s'avère efficace dans le cadre d'un suivi de tempo en temps réel, laissant la place à de nombreuses améliorations futures. Plusieurs perspectives sont ainsi évoquées dans la partie suivante.

Perspectives

A propos de la mesure de Nouno : Dans ce stage, nous avons uniquement considéré une séquence "d'onsets" pour la séquence d'entrée. Or, Poven a montré dans [PE85] par un ensemble de règles l'importance de la durée et de la position des notes. Ainsi, une meilleures estimations du tempo pourrait être obtenue en prenant en compte l'intensité des notes produites, la durée de celle-ci, et leurs placements harmoniques. A titre d'exemple, on pourrait par exemple pondérer chaque terme $erreur_k$ de la mesure par l'intensité des notes rencontrés, et ainsi effectuer un barycentre qui donne plus de poids aux notes de forte intensité. On pourrait aussi inclure des a priori sur la musique que l'on écoute, en intégrant par exemple les notions musicales classique de tonalité, fondamentale, quinte, tierce ... qui ont une influence sur notre perception des temps forts.

Le modèle de Gilbert Nouno est basé sur le théorème fondamental qui stipule que le musicien joue au moins deux notes sur le temps, sans quoi le tempo ne nous est pas accessible. Ce modèle suppose implicitement une absence de bruit du musicien. En effet, l'être humain est capable de déceler le tempo malgré l'imprécision du musicien. Aussi, il serait intéressant de rajouter un modèle de bruit dans le formalisme introduit par Gilbert Nouno. Quelques considérations ont étés faites en ce sens dans ce rapport par le biais des modèles posés du musicien parfait et imparfait, mais une véritable étude sur les conséquences d'une telle considération sur le théorème fondamental pour s'avérer très intéressant.

A propos des méthodes développées : Les deux méthodes que nous avons introduites dans ce stage concernant la reconstruction du vecteur d'état souffrent de défauts. En effet, la méthode d'interpolation ne pallie pas le problème des mesures incomplètes, et la méthode d'estimation par noyaux peux s'avérer biaisée pour certaines répartitions des mesures. On a beaucoup cherché dans ce stage à développer une méthode de "fitting" pour des mesures bruitées. La principale difficulté vient du fait que les mesures sont bruitées uniquement par des valeurs négatives, comme illustré sur la figure 3.1. Dans nos recherches actuelles, après avoir écarté les modèles probabilistiques classiques ou M-estimateur qui ne prennent pas en compte le coté uni-dimensionnel de l'erreur, on s'est intéressé aux méthodes de détections d'enveloppe. Instinctivement, en regardant la figure 3.1, on voit bien que c'est bien ce paramètre que l'on cherche à estimer. On pense pouvoir obtenir une meilleure construction du vecteur d'état en utilisant de telle technique issue de la littérature. Il faut cependant s'assurer que la phase de l'estimation de l'enveloppe (qui représente le pas optimal recherché) soit non biaisé.

Pour les méthodes de reconstruction de la grille, et notamment au niveau de paramètre de la phase l'oscillateur non linéaire semble le plus prometteur. Cependant, cet oscillateur est

très perturbé par un événement qui ne tombe pas sur la grille. Les musiciens ont tendance ne pas suivre exactement la grille en raison de leur expressivité. Souvent, une note avec une forte intensité sera au contraire sur la grille. Il est alors envisageable de faire varier dans le temps les coefficients η_Φ et η_r en les pondérants par exemple avec l'intensité de la note. De cette façon, notre oscillateur sera plus stable par rapport à des notes qui ne sont pas exactement sur la grille, ou autrement dit, aux effets de style du musicien.

Dernière remarque : Toutes les méthodes décrites ici ont pour objectif le suivi du tatum, et plus précisément le suivi d'une métrique dans un intervalle particulier $[\Delta_{min} \quad 2{\cdot}\Delta_{min}]$. Pour pouvoir faire un vrai suivi de tempo, il faudrait pouvoir décimer cette métrique. Pour ce faire, il faut pouvoir repérer le temps fort de la mesure, ainsi que la métrique du morceau. Il existe de nombreuses méthodes pour mettre en oeuvre une telle opération dans la littérature.

Bibliographie

[AHG12] J.R. Zapata J.L. Oliveira A. Holzapfel, M.E.P. Davies and F. Gouyon. Selective sampling for beat tracking evaluation. *Audio, Speech, and Language Processing, IEEE Transactions, 20(9) :2539–2548*, November 2012.

[AK99] Mohamed A. and Schwarz K. Adaptive kalman filtering for ins/gps. *Journal of Geodesy, Vol.73, 193-203*, 1999.

[CD04] Nick Collins and Cb Dp. Beat induction and rhythm analysis for live audio processing : 1st year phd report. Technical report, University of Cambridge, 2004.

[Cem01] A. T. Cemgil. Tempo tracking and rhythm quantization by sequential monte carlo. *MIT Press, In Proc. NIPS*, 2001.

[Cem03] A. T. Cemgil. Monte carlo methods for tempo tracking and rhythm quantization. *Journal of Articial Intelligence Research, 18 :4581*, 2003.

[Con08] Arshia Cont. ANTESCOFO : Anticipatory Synchronization and Control of Interactive Parameters in Computer Music. In *International Computer Music Conference (ICMC)*, pages 33–40, Belfast, Irlande, August 2008.

[DcB93] Carolyn Drake and Marie claire Botfe. Tempo sensitivity in auditory sequences : Evidence for a multiple-look model. *Perception & Psychophysics 54(3),2 77-286*, 1993.

[Dix01] Simon Dixon. An empirical comparison of tempo trackers. *Proceedings of the 8th Brazilian Symposium on Computer Music, pages 832–840*, 2001.

[DWR07] Wang J. Ding W. and C. Rizos. Improving adaptive kalman estimation in gps/ins integration. *The Journal of Navigation, Vol.60, 517- 529*, 2007.

[FLS85] Ray. Jackendoff F. Lerdahl and W. Slawson. A reply to peel and slawson's review of "a generative theory of tonal music". *Journal of Music Theory, 29(1) :145–160*, 1985.

[GM99] Masataka Goto and Yoichi Muraoka. Chord change detection for musical decisions. *Speech Communication, 27(3) :311–335*, 1999.

[Got01] Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research, 30(2) :159– 171*, 2001.

[J00] Wang J. Stochastic modeling for real-time kinematic gps/glonass position. *Navigation, Vol. 46, No. 4, 297-305*, 2000.

[JAHF09] Marcelo M Wanderley Jason A Hockman and Ichiro Fujinaga. Real-time phase vocoder manipulation by runner's pace. *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME)*, 2009.

[Kla03a] Anssi P Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *TSpeech and Audio Processing, IEEE Transactions on, 11(6) :804–816*, 2003.

[Kla03b] Anssi P Klapuri. Sound onset detection by applying psychoacoustic knowledge. in acoustics. *Acoustics, Speech, and Signal Processing, 1999 IEEE International Conference on, volume 6, pages 3089–3092*, 2003.

[LJ99] E. W. Large and M. R. Jones. The dynamics of attending : How people track time-varying events. *Psychological Review, vol 106, no 1, p119-159*, 1999.

[LP02] E. W. Large and C. Palmer. Perceiving temporal regularity in music. *Cognitive Science, (26) :1–37*, 2002.

[May79] P. S. Maybeck. Stochastic models, estimation, and control. *volume 141 of Mathematics in Science and Engineering. Academic Press*, 1979.

[Maz94] Guerino Mazzola. Tempo curves revisited : Hierarchies of performance fields. *Computer Music Journal, 18 :40–52*, 1994.

[Mur02] K. P. Murphy. *Stochastic models, estimation, and control.* Phd thesis, UC Berkeley, 2002.

[Nou08] Gilbert Nouno. *Suivi de Tempo Appliqué aux Musiques Improvisées : à la recherche du temps perdu...* Thèse de doctorat, UPMC IRCAM, 2008.

[Par94] Richard Parncutt. A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception, pages 409–464*, 1994.

[PE85] Dirk-Jan Povel and Peter Essens. Perception of temporal patterns. *Music Perception, pages 411–440*, 1985.

[Pee05] Geoffroy Peeters. Time variable tempo detection and beat marking. In *Proc. ICMC*. Citeseer, 2005.

[Pee06] Geoffroy Peeters. Template-based estimation of time-varying tempo. *EUR- ASIP Journal on Advances in Signal Processing*, 2007, 2006.

[R70] Mehra R. On the identification of variance and adaptive kalman filtering. *IEEE Trans Automat Contr 4C-15(2), 175-184*, 1970.

[R71] Mehra R. Online identification of linear dynamic systems with applications to kalman filtering. *IEEE. Trans Automat Contr AC-, Vo.16, No.1, 12-21*, 1971.

[Sch98] E.D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America, 103 :588*, 1998.

[Sep01] Jarno Seppanen. *Computational models of musical meter recognition.* Master's thesis, Tampere University of Technology, 2001.

Annexe A

Modification de l'espace temporel pour une courbe de Tempo donnée

A.1 Définition du problème

Soit $\mathscr{E}$ un sous ensemble de $\mathbb{N}$ noté $\mathscr{E} = \{e_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq N}}$ et tel que $\text{PGCD}(IO) = 1$

Avec $IO = \{io_j = e_j - e_{j-1}\}_{\substack{j\in\mathbb{N} \\ 2\leq j\leq N}}$ l'ensemble des intervalles associés à l'ensemble $\mathscr{E}$.

On définit alors $\mathscr{D} \subseteq \mathbb{N}$ noté $\mathscr{D} = \{d_i\}_{\substack{i\in\mathbb{N} \\ 1\leq i\leq S}}$, fini, comme la plus partie de $\mathbb{N}$ tel que :

- $\mathscr{E} \subseteq \mathscr{D}$
- $d_{i+1} - d_i = C, \qquad \forall i \in \mathbb{N} \quad \& \quad 1 \leq i \leq S-1$
- $d_1 = e_1$ et $d_S = e_N$

On peut alors définir la fonction d'index σ tel que :

$$\begin{aligned} \sigma : \quad \mathbb{N} \quad &\to \mathbb{N} \\ j \quad &\mapsto \sigma(j) \qquad \text{tel que} \qquad d_{\sigma(j)} = e_j \end{aligned}$$

Montrons que $d_{i+1} - d_i = C = 1$:

Comme $\text{PGCD}(IO) = 1$, les io_j sont premiers entre eux deux à deux, on peut donc appliquer le théorème de Bezout et affirmer que :

$$\text{PGCD}(IO) = 1 \iff \exists \quad \{a_j\}_{\substack{j\in\mathbb{N} \\ 1\leq j\leq N-1}} \in \mathbb{Z} \qquad \text{tel que} \qquad \sum_{j=1}^{N-1} a_j \cdot io_{j+1} = 1 \tag{A.1}$$

Soit :

$$\sum_{j=1}^{N-1} a_j \cdot io_{j+1} = 1 \quad \text{(A.2)}$$

$$\sum_{j=1}^{N-1} a_j \cdot (e_j - e_{j-1}) = 1 \quad \text{(A.3)}$$

$$\sum_{j=1}^{N-1} a_j \cdot (d_{\sigma(j)} - d_{\sigma(j-1)}) = 1 \quad \text{(A.4)}$$

$$\sum_{j=1}^{N-1} a_j \cdot (d_{\sigma(j)} - d_{\sigma(j)-1} + d_{\sigma(j)-1} \cdots - d_{\sigma(j-1))+1} + d_{\sigma(j-1))+1} - d_{\sigma(j-1)}) = 1 \quad \text{(A.5)}$$

comme $d_1 = e_1$ et $d_S = e_N$ et $d_{i+1} - d_i = C$, on peut réécrire l'équation A.5 ci-dessus par :

$$\sum_{j=1}^{N-1} a_j \cdot N_j \cdot C = 1 \quad \text{(A.6)}$$

$$C \cdot \sum_{j=1}^{N-1} a_j \cdot M_j = 1 \quad \text{(A.7)}$$

avec M_j le nombre d'intervalles $d_{\sigma(j)} - d_{\sigma(j)-1}$ nécessaire pour former l'intervalle $e_j - e_{j-1}$.

Dans l'equation A.7, le terme $\sum_{j=1}^{N-1} a_j \cdot M_j$ est une constante qui appartient à $\mathbb{Z}$, et C appartient à $\mathbb{N}$. La seule solution possible pour cette égalité est donc $C = 1$ ainsi que $\sum_{j=1}^{N-1} a_j \cdot M_j = 1$.

$\mathscr{E}$ représente l'ensemble de la partition rythmique dont le tempo n'est pas encore défini, c'est-à-dire avec l'absence de notion de temps réel. Ainsi, chaque événement e_j correspond dans le modèle à la position d'une note en temps symbolique dans la partition. De plus, à chaque e_j peut être associé un d_i et l'on sait que $\mathscr{D}$ est la plus partie de $\mathbb{N}$ qui contient l'ensemble $\mathscr{E}$. $\mathscr{D}$ représente donc par définition le tatum c'est-à-dire la plus petite pulsation de la partition. On dit alors que $\mathscr{E}$ et $\mathscr{D}$ sont dans l'espace pulsationnel.

On associe alors respectivement aux ensembles $\{d_i\}_{\substack{i\in\mathbb{N} \\ 1\leq i\leq S}}$ et $\{e_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq N}}$ les quantités $\{\theta_i\}_{\substack{i\in\mathbb{N} \\ 1\leq i\leq S}}$ et $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq N}}$ représentant l'instant absolu (en seconde par exemple) où est joué la note. On a donc $\{\theta_i\}_{\substack{i\in\mathbb{N} \\ 1\leq i\leq S}}$, $\{t_j\}_{\substack{j\in\mathbb{N} \\ 1\leq i\leq N}} \in I_1$ qui est un intervalle de $\mathbb{R}$.

Remarquons que l'on a encore, si $e_j = d_{\sigma(j)}$, $\quad t_j = \theta_{\sigma(j)}$.

Le lien entre les θ_i et les d_i est alors exprimé par un pseudo-tempo τ que l'on définit tel que :

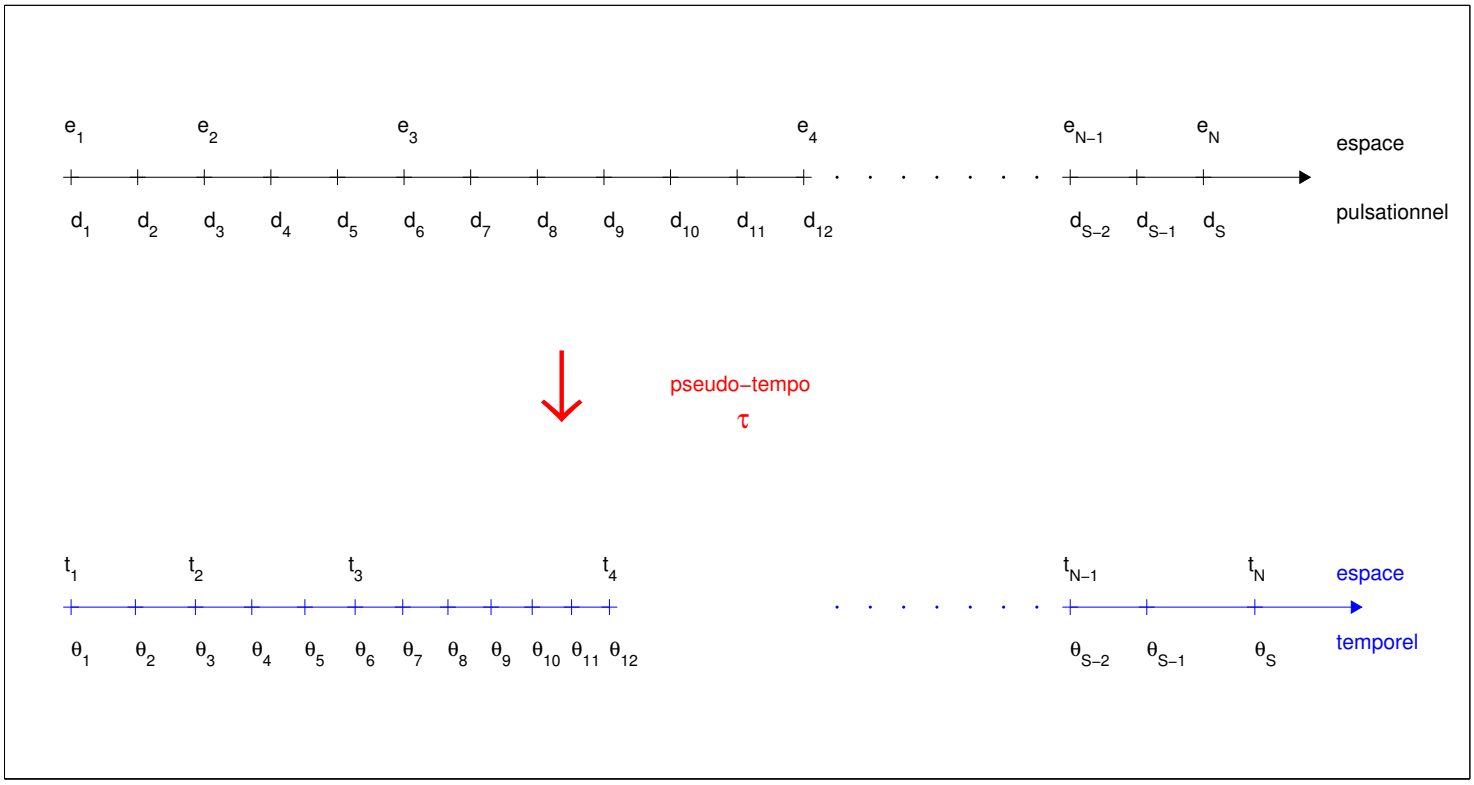

FIGURE A.1 – Schema du formalisme utilisé

$$\begin{aligned} \tau : \quad \mathscr{D} &\to I_2 \subset \mathbb{R} \\ d_i &\mapsto \tau(d_i) = \frac{d_i - d_{i-1}}{\theta_i - \theta_{i-1}} = \frac{1}{\theta_i - \theta_{i-1}} \end{aligned} \tag{A.8}$$

Remarque 1 : On appelle τ pseudo tempo car c'est le tempo du tatum, et non le tempo dans le sens commun du terme. On remarquera que la valeur qu'associe la fonction τ est multiple de la valeur qu'associe la "vrai" fonction du Tempo T. En effet, la pulsation associée a un certain tempo est un sous ensemble de la pulsation du tatum. On peut passer de l'un à l'autre par une simple décimation d'un facteur $M \in \mathbb{N}$, qui se traduit tout aussi simplement par $M * T(d_i) = \tau(d_i), \forall i \in \mathbb{N}$ et $2 \leq i \leq S$.

Remarque 2 : Le pseudo tempo τ tel qu'il est définit est une fonction mathématique qui associe à un endroit de la partition une vitesse d'exécution particulière de celle-ci. Dans le lexique habituel, τ est plutôt la courbe de tempo, et $\tau(d_i)$ est la valeur locale du tempo de dimension $\frac{evenement}{temps}$.

Remarque 3 : Cette définition implique un tempo local, et prend donc en compte non seulement les indications du compositeur concernant le tempo (*72 à la noire, Accelerando, etc ...*) mais aussi les erreurs de l'interprète qui ne va pas jouer la partition de manière exacte. La variation dans le jeu du musicien peut être considérée d'un point de vue local comme une variation de tempo (même infime).

Remarque 4 : La définition est intuitive mais n'en reste pas moins arbitraire. Notamment remarquons que le tempo n'a de sens que si l'on considère au moins deux éléments (puisque $i \geq 2$) et que la définition que l'on a choisi est causale.

Le problème est donc de savoir comment la fonction τ influe sur la répartition des t_j en prenant en compte l'espace pulsationnel.

A.2 Résolution du problème, exprimer t_j en fonction de τ et d_i

Il suffit alors de calculer l'écart temporel entre deux événements à l'aide de la fonction τ :

$$\begin{aligned} t_{j+1} - t_j &= \theta_{\sigma(j+1)} - \theta_{\sigma(j)} \\ &= \underbrace{\theta_{\sigma(j+1)} - \theta_{\sigma(j+1)-1}}_{} + \underbrace{\theta_{i\sigma(j+1)-1} - \theta_{\sigma(j+1)-2}}_{} + \ldots + \underbrace{\theta_{\sigma(j)+2} - \theta_{\sigma(j)+1}}_{} + \underbrace{\theta_{i\sigma(j)+1} - \theta_{\sigma(j)}}_{} \\ &= \frac{1}{\tau(d_{\sigma(j+2)})} + \frac{1}{\tau(d_{\sigma(j+1)-1})} + \ldots + \frac{1}{\tau(d_{\sigma(j)+2})} + \frac{1}{\tau(d_{\sigma(j)+1})} \\ &= \sum_{i=\sigma(j)+1}^{\sigma(j+1)} \frac{1}{\tau(d_i)} \end{aligned}$$

Par simplification canonique, on obtient alors pour n'importe quelle t_j la formule :

$$t_j - t_1 = \sum_{i=1}^{\sigma(j)} \frac{1}{\tau(d_i)} \tag{A.9}$$

Remarque 5 : Pour retrouver le Tempo au sens commun que l'on a défini par la fonction T, la linéarité de la somme nous permet d'écrire directement :

$$t_j - t_1 = \frac{1}{M} \sum_{i=1}^{\sigma(j)} \frac{1}{T(d_i)} \tag{A.10}$$

M étant le facteur de décimation.

Remarque 6 : De cette manière, le pseudo tempo est défini uniquement dans l'espace $\mathscr{D}$, qui est une version discrétisée de la position dans la partition. C'est parfaitement logique puisque le tempo est par essence discret, prenant son origine dans le rythme musical lui aussi

discret. Dans la pratique, on veut pouvoir interpoler la valeur d'un tempo à n'importe quelle position dans la partition, ce qui inclut une position entre deux notes. Cela revient de passer de l'espace discret $\mathscr{D}$ vers un espace continu, toujours dans le domaine pulsationnel,que l'on appelle $\mathscr{D}_{bis} \subset \mathbb{R}$. Sa variable associé sera notée d_{bis}. La fonction "courbe de tempo" partant de $\mathscr{D}_{bis}$ et allant vers I_2 est alors notée T_{bis} . Il y a de fait plusieurs manières de faire pour cela et donc une infinité de fonctions T_{bis} possibles. Une manière de faire est alors de poser :

$$t_j - t_1 \quad = \quad \frac{1}{M} \int_{d_1}^{d_{\sigma(j)}} \frac{1}{T_{bis}(d_{bis})} \mathrm{d}d_{bis} \tag{A.11}$$

On peut alors définir T_{bis} comme courbe de tempo acceptable si et seulement si :

- la fonction T_{bis} restreinte sur $\mathscr{D}$ est égale à T. Cette condition implique donc l'équation A.10.
- la fonction T_{bis} est continue et intégrable telle qu'elle satisfasse l'equation A.11

Remarque 7 : Dans cette démonstration, on a imposé la grille du tatum en imposant $\mathrm{PGCD}(IO) = 1$ pour pouvoir obtenir $\tau(d_i) = \dfrac{1}{\theta_i - \theta_{i-1}}$. On est ainsi cohérent avec le sujet du stage. Cependant, prendre une métrique suffisamment grande pour contenir tous les événements est suffisant. On aurait eu alors $\tau(d_i) = \dfrac{P}{\theta_i - \theta_{i-1}}$, avec $P \in \mathbb{N}$, et l'équation A.11 serait devenue de manière plus générale :

$$t_j - t_1 \quad \propto \quad \int_{d_1}^{d_{\sigma(j)}} \frac{1}{T_{bis}(d_{bis})} \mathrm{d}d_{bis} \tag{A.12}$$

Annexe B

Filtrage adaptatif, Méthode basés sur les résidus et méthode basé sur les innovations

On présente brièvement ici les méthodes de filtrage adaptatif. Plus d'informations peuvent être trouvées dans les références données. Cette annexe se base sur le formalisme posé dans la section 3.3. Ces deux méthodes nécessitent une initialisation de la matrice $\mathbf{R}$ et $\mathbf{Q}$ qui sont alors initialisées avec les valeurs de la section 3.3

B.1 Estimation de $\mathbf{R}_p$

B.1.1 Méthode basée sur l'innovation [R70], [R71], [AK99]

L'Idée ici de d'estimer la matrice $\mathbf{R}_p$ en considérant les innovations. On pose alors pour estimé de la matrice $\mathbf{R}_p$:

$$\tilde{\mathbf{R}}_p = \tilde{\mathbf{C}}_y - \mathbf{H}_p \mathbf{P}_{p|p-1} \mathbf{H}_p^T \tag{B.1}$$

ou $\tilde{\mathbf{C}}_y$ est l'estimé de la matrice de covariance de l'innovation pouvant être estimée par :

$$\tilde{\mathbf{C}}_y = \frac{1}{m} \sum_{i=1}^{m} \tilde{\mathbf{y}}_{p-i} \tilde{\mathbf{y}}_{p-i}^T \tag{B.2}$$

correspondant à une estimation sur une fenêtre de taille m à l'époque p.

Remarques :

– Cette estimation ne peut se faire que si on a à disposition une fenêtre d'observation (au

sens de l'époque p) de taille m. Dans notre cas, puisque le filtre de Kalman est mis à jour à chaque fenêtre d'observation, cela correspond à m mesure de Ψ_p. C'est en fait une "fenêtre de fenêtre". Dans le cas où on a pas encore m observations à disposition, on maintient R_p à sa valeur initiale.

– Avec cette méthode, on est pas assuré que $\mathbf{R}_p$ est définie positive, c'est-à-dire inversible et dont les valeurs propres sont positives. Cette condition est nécessaire pour qu'elle puisse être considérée comme la matrice de covariance d'un bruit. Pour mettre en place cette méthode, on est obligé d'inclure un test vérifiant cette condition. Si ce test échoue, on reprend alors la valeur initiale de $\mathbf{R}$ pour $\mathbf{R}_p$ plutôt que la valeur calculée.

B.1.2 Méthode basée sur les résidus de la séquence, [J00]

Le concept est quasiment identique, on définit cette fois la quantité que l'on appelle "residu" par :

$$\bar{\mathbf{y}}_p = \mathbf{z}_p - \mathbf{H}_p \bar{\mathbf{x}}_{p|p} \tag{B.3}$$

et on pose alors pour estimé de $\mathbf{R}$:

$$\tilde{\mathbf{R}}_p = \tilde{\mathbf{C}}_{\bar{y}} + \mathbf{H}_p \mathbf{P}_{p|p} \mathbf{H}_p^T \tag{B.4}$$

on estime cette fois $\tilde{\mathbf{C}}_{\bar{y}}$ par

$$\tilde{\mathbf{C}}_{\bar{y}} = \frac{1}{m} \sum_{m}^{i=1} \bar{\mathbf{y}}_{p-i} \bar{\mathbf{y}}_{p-i}^T \tag{B.5}$$

$\mathbf{R}_p$ est maintenant assurément définit positive.

B.2 Estimation de $\mathbf{Q}_p$

B.2.1 Méthode Commune aux deux méthodes, [DWR07]

Si on suppose que $\mathbf{P}_{p-1|p-1}$ et $\mathbf{R}_p$ sont connus, alors $\mathbf{Q}_p$ peut être mis à jour linéairement par le ratio entre l'estimé de la covariance de l'innovation et celle prédite tel que :

$$\tilde{\mathbf{Q}}_p = \mathbf{Q}_{p-1} \sqrt{\alpha} \tag{B.6}$$

avec :

$$\alpha = \frac{trace(\tilde{\mathbf{C}}_y - \mathbf{R}_p)}{trace(\mathbf{H}_p \mathbf{P}_{p|p-1} \mathbf{H}_p^T)} = \frac{trace(\mathbf{H}_p(\mathbf{F}_p \mathbf{P}_{p-1|p-1} \mathbf{F}_p^T + \tilde{\mathbf{Q}}_{p-1}) \mathbf{H}_p^T)}{trace(\mathbf{H}_p(\mathbf{F}_p \mathbf{P}_{p-1|p-1} \mathbf{F}_p^T + \mathbf{Q}_{p-1}) \mathbf{H}_p^T)} \tag{B.7}$$

Remarques : L'estimation de $\mathbf{R}_p$ s'appuie sur $\mathbf{P}_{p|p-1}$ ou $\mathbf{P}_{p|p}$ qui est calculé à partir de la matrice $\mathbf{Q}_{p-1}$. On suppose implicitement que notre estimation précédente de $\mathbf{Q}_p$ est exacte. De même, l'estimation $\mathbf{Q}_p$ s'appuie indirectement sur la supposition que notre estimation de $\mathbf{R}_p$ est

juste. C'est en fait un schema alterné où l'on suppose d'abord que $\mathbf{Q}_{p-1}$ est juste pour estimer $\mathbf{R}_p$, puis que cette estimation de $\mathbf{R}_p$ est juste pour estimer $\mathbf{Q}_p$.