

La synchronisation spontanée : des applaudissements aux métronomes

Guilhem Repetto, candidat n°20389

Dans la nature, on observe de nombreux phénomènes de synchronisation entre des individus de même espèce. On peut penser qu'ils se sont développés au cours de l'évolution pour optimiser les chances de survie du groupe. On peut par exemple citer :

- la synchronisation du chant des cigales ;
- la synchronisation du clignotement des lucioles mâles, probablement pour émettre une lumière plus forte à plusieurs et donc attirer plus de femelles ;
- la synchronisation des menstruations entre femmes ou “effet dortoir”, et mammifères en général, qui vivent suffisamment proches les unes des autres ;
- la synchronisation de la migration des oiseaux, tortues, poissons, qui permet à un plus grand nombre de ces individus de parvenir à destination en affrontant à plusieurs les aléas de la route ;
- la synchronisation de la nage de certains groupes de poissons, qui permet une plus grande vigilance vis-à-vis des prédateurs et une rapide diffusion de l'information en cas de danger ;
- la synchronisation de la fuite d'un troupeau, initiée par un individu qui entraîne toute la masse.

Dans notre univers technologique, la synchronisation est un enjeu important dans certains domaines comme par exemple celui de l'énergie électrique. Un réseau de transport et distribution comporte comme source plusieurs générateurs de natures différentes. En Europe, la fréquence choisie est de 50 Hz. Les différentes centrales sont surveillées et contrôlées pour rester en phase et synchrones, afin de prévenir des résonances destructrices [5].

C'est cependant un autre phénomène qui a motivé mon étude. On observe souvent, lors des applaudissements du public à la fin d'un spectacle, une période où tout l'auditoire applaudit en rythme. D'abord aléatoires, les applaudissements finissent parfois par se synchroniser. Ce phénomène n'a pas une origine strictement biologique comme c'est le cas chez les lucioles. On peut avancer plusieurs hypothèses d'ordre sociologique et comportementale :

- l'instinct grégaire qui pousse chacun à ne pas se distinguer et à entrer dans le rythme ;
- des spectateurs enthousiastes coopèrent volontiers et consciemment à la synchronisation afin d'optimiser le volume sonore produit et ainsi obtenir un bis ;

L'applaudissement synchronisé serait un message à l'adresse des artistes. Quand il se produit, ce phénomène peut être plus ou moins marqué et plus ou moins long. Il peut être atténué ou renforcé selon les réactions des artistes. Il peut disparaître et se produire plusieurs fois pendant la séquence d'applaudissements.

Comment ces périodes de synchronisation se développent-elles ?

Appliqué à d'autres situations, ce modèle peut présenter un intérêt réel. Par exemple, dans un réseau de machines tournantes comme des turbines, on peut imaginer que si

plusieurs d'entre elles sont mal équilibrées, il peut se produire des mouvements parasites amplifiés par une éventuelle synchronisation.

Table des matières

I	Cadre théorique	4
1	Cadre de l'étude, considérations théoriques	4
1.1	Définitions, vocabulaire	4
1.2	Le modèle général de Winfree	4
1.3	Le modèle de Kuramoto	4
2	Étude des influenceurs	5
2.1	Les couplages dissymétriques	5
2.2	Le cas particulier $N = 2$	6
2.3	Généralisation aux chaînes et arbres d'influenceurs	6
II	Expériences	7
3	Conception d'une expérience	8
4	Réalisation des expériences	8
5	Traitement et analyse des résultats	9
5.1	Fonctionnement du programme utilisé	9
5.2	La transformée de Hough circulaire	10
5.3	Format des données obtenues	10
III	Annexes	11
6	Simulation de différents couplages pour les mêmes conditions initiales	11
7	Détails des expériences	12
8	Données obtenues lors des expériences	14
9	Preuve des théorèmes	18
9.1	Stabilité du modèle de Kuramoto pour une équirépartition des phases . . .	18
9.2	Première approche naïve du théorème 1 :	18
9.3	Preuve rigoureuse du théorème 1 :	21
9.4	Preuve du théorème 2 :	22
9.5	Preuve du corollaire 2 :	22
10	Code source des programmes python	22

Première partie

Cadre théorique

1 Cadre de l'étude, considérations théoriques

1.1 Définitions, vocabulaire

Nous étudions l'influence de plusieurs individus effectuant une action répétitive, perçue par l'ensemble du groupe. Nous désignerons par le terme général **oscillateur** chaque individu. Chaque oscillateur possède une fréquence d'oscillation (ou de rotation), ainsi qu'une phase variant en fonction du temps.

Le nombre d'oscillateurs considérés est un nombre entier noté N . On parlera indifféremment des oscillateurs $1, \dots, N$ ou des oscillateurs $O_1, \dots, O_N$. Pour un oscillateur i , on considère de plus :

- θ_i sa phase instantanée
- $\dot{\theta}_i$ sa fréquence instantanée
- ω_i sa fréquence naturelle

1.2 Le modèle général de Winfree

Winfree [2] propose un modèle dans lequel la phase de chaque oscillateur est influencée par une combinaison de paramètres de l'ensemble des oscillateurs. Pour chaque $i \in \llbracket 1; N \rrbracket$, l'équation de la phase de l'oscillateur O_i est

$$\dot{\theta}_i = \omega_i + \left(\sum_{j=1}^N X(\theta_j) \right) Z(\theta_i)$$

où :

- chaque oscillateur O_j exerce une influence X sur l'oscillateur O_i . Cette influence dépend de sa phase, d'où la fonction $X(\theta_j)$;
- $Z(\theta_i)$ est la fonction représentant la sensibilité de l'oscillateur O_i , qui dépend de θ_i .

1.3 Le modèle de Kuramoto

Kuramoto [1] considère un cas particulier du modèle de Winfree :

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^N \Gamma_{ij} (\theta_j - \theta_i)$$

Ici, la sensibilité Z de chaque oscillateur est constante, égale à 1. Dans ce modèle, $\Gamma \in \mathcal{M}_N(\mathbb{R}_+^{\mathbb{R}})$ est la matrice de couplage entre les oscillateurs. Ainsi, la fonction $\Gamma_{i,j}$ représente le couple entre les oscillateurs i et j . Pour $t \in \mathbb{R}_+$, le scalaire $\Gamma_{i,j}(t)$ représente le couple entre i et j à l'instant t . Kuramoto pose que l'influence entre les oscillateurs est identique et proportionnelle au sinus de la différence des phases. Il pose donc :

$$\forall (i, j) \in \llbracket 1; N \rrbracket, \Gamma_{ij}(\theta_j - \theta_i) = \frac{K}{N} \sin(\theta_j - \theta_i)$$

où K est une constante positive qui représente le couple entre deux oscillateurs. L'utilisation de la fonction sinus a plusieurs intérêts :

- l'influence mutuelle est faible si des oscillateurs sont presque en phase, et forte s'ils sont plus décalés ;
- la fonction sinus étant impaire, il y a une attraction mutuelle : si la phase de θ_i se rapproche de celle de θ_j , alors celle de θ_j se rapproche aussi de celle de θ_i ;
- enfin, le système reste stable s'il y a parfaite opposition de phase, c'est-à-dire si $\theta_j = \theta_i + \pi$, car, dans ce cas, $\sin(\theta_i - \theta_j) = \sin(\pi) = 0$. Plus généralement, si les différentes phases sont équiréparties sur l'intervalle $[0; 2\pi[$, le système reste stable.

Cela est cohérent avec la situation étudiée : lors d'applaudissements, il se peut que certains applaudissent à contre-temps, sans que le rythme ne soit affecté. Le modèle théorique énonce que cet équilibre s'établit si les phases sont parfaitement équiréparties.

Par exemple, avec 3 oscillateurs, on obtient la matrice de couplage :

$$\Gamma(x) = \frac{K}{3} \begin{pmatrix} 0 & \sin(x) & \sin(x) \\ \sin(x) & 0 & \sin(x) \\ \sin(x) & \sin(x) & 0 \end{pmatrix}$$

En utilisant cette matrice dans l'équation (1.3), on obtient le *modèle de Kuramoto* :

$$\forall i \in \llbracket 1, N \rrbracket, \quad \dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i)$$

Paramètre d'ordre

Pour visualiser le phénomène, Kuramoto introduit un indicateur complexe de synchronisation $re^{i\psi}$:

$$re^{i\psi} = \frac{1}{N} \sum_{j=1}^N e^{i\theta_j(t)}$$

Ce "paramètre d'ordre" est interprété comme le rythme collectif produit par l'ensemble des oscillateurs. Il vérifie $r \in [0, 1]$ et $\psi \in [0, 2\pi[$.

Si r est proche de 0, alors les phases sont éparpillées dans tout l'intervalle $[0, 2\pi[$. Si r est proche de 1, alors les phases sont proches les unes des autres, et l'ensemble des oscillateurs agit comme un unique oscillateur géant. Le nombre ψ représente la phase moyenne. On montre que le modèle de Kuramoto se met alors sous la forme :

$$\dot{\theta}_i = \omega_i + Kr \sin(\psi - \theta_i)$$

Chaque oscillateur apparaît alors couplé uniquement à la moyenne de l'ensemble : la phase θ_i est reliée à la phase moyenne ψ , et le couplage devient proportionnel à la cohérence r . Cela montre une réaction de tendance à la synchronisation : plus les oscillateurs sont en phase, plus r croît, ce qui favorise la synchronisation [4].

2 Étude des influenceurs

2.1 Les couplages dissymétriques

Un influenceur ayant pour but conscient de propager son rythme, nous supposons que sa sensibilité envers les autres est très faible. Autrement dit, le couplage est dans ce cas

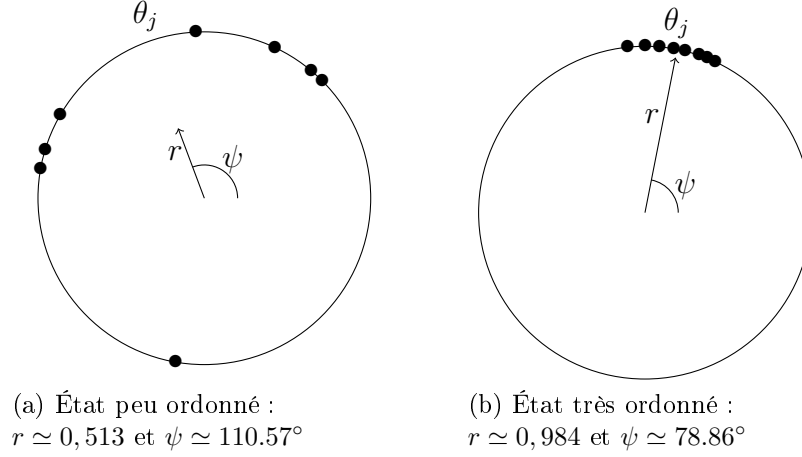


FIGURE 1 – Différentes situations avec le paramètre d'ordre associé

dissymétrique : fort de lui vers les autres et faible voire nul des autres vers lui. Cela se traduit dans le modèle par le fait que la matrice Γ d'abord supposée antisymétrique (i.e $\forall i, j \in \llbracket 1, N \rrbracket, \Gamma_{i,j} = -\Gamma_{j,i}$) ne l'est plus. Par exemple, avec 3 oscillateurs, si θ_1 est un influenceur, on peut obtenir une matrice du type :

$$\Gamma(x) = \frac{K}{3} \begin{pmatrix} 0 & 2\sin(x) & 2\sin(x) \\ 0 & 0 & \sin(x) \\ 0 & \sin(x) & 0 \end{pmatrix}$$

L'influence de θ_1 vers les autres est la plus importante : $\Gamma_{1,2}$ et $\Gamma_{1,3}$ valent tous deux $2\sin$. Les autres ne l'influencent pas : $\Gamma_{2,1}$ et $\Gamma_{3,1}$ sont égaux à la fonction nulle.

2.2 Le cas particulier $N = 2$

Plaçons-nous dans le cas $N = 2$. On suppose que O_1 est influenceur : il influence O_2 sans être influencé. On a donc : $\Gamma_{2,1} = 0$ et $\Gamma_{1,2} \neq 0$. Dans ce cas, on adoptera le symbolisme suivant :

$$\begin{aligned} O_1 &\rightarrow O_2 && \text{si } \Gamma_{2,1} = 0 \\ O_1 &\nrightarrow O_2 && \text{sinon} \end{aligned}$$

Théorème 1. *Dans le cas d'une différence de phases non-égale à π à l'origine des temps, les oscillateurs se synchronisent, c'est-à-dire :*

$$\lim_{t \rightarrow +\infty} \theta_2(t) - \theta_1(t) = 0$$

2.3 Généralisation aux chaînes et arbres d'influenceurs

On peut généraliser cette propriété à des chaînes d'influenceurs. Nous supposons que $N \geq 2$.

Chaîne d'influence Les oscillateurs sont disposés en *chaîne d'influence* si et seulement si :

- il existe un unique $r \in \llbracket 1; N \rrbracket$ tel que pour tout $i \in \llbracket 1; N \rrbracket, O_i \nrightarrow O_r$

- pour tout $i \in \llbracket 1; N \rrbracket \setminus \{r\}$, il existe un unique j tel que $O_j \rightarrow O_i$
- pour tout $i \in \llbracket 1; N \rrbracket$, l'ensemble des O_j tels que $O_i \rightarrow O_j$ est soit vide, soit réduit à un unique élément.

Arbre d'influence Les oscillateurs sont disposés en *arbre d'influence* si et seulement si :

- il existe un unique $r \in \llbracket 1; N \rrbracket$ tel que pour tout $i \in \llbracket 1; N \rrbracket$, $O_i \rightarrow O_r$
- pour tout $i \in \llbracket 1; N \rrbracket \setminus \{r\}$, il existe un unique j tel que $O_j \rightarrow O_i$
- pour tout $i \in \llbracket 1; N \rrbracket \setminus \{r\}$, il existe un entier n et $(O_{j_1}, \dots, O_{j_n})$ tels que

$$O_r \rightarrow O_{j_1} \rightarrow \dots \rightarrow O_{j_n} \rightarrow O_i$$

Dans les deux cas, on dira que O_i est un *ancêtre* de O_j si et seulement s'il existe des oscillateurs $O_{i_0}, \dots, O_{i_r}$ tels que

$$O_i \rightarrow O_{i_0} \rightarrow \dots \rightarrow O_{i_r} \rightarrow O_j$$

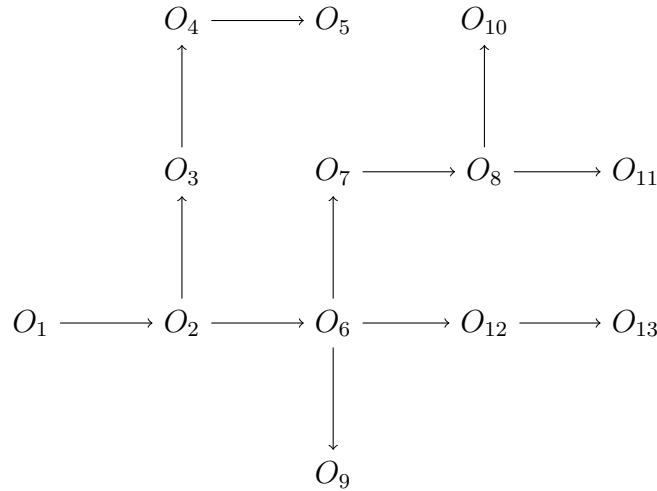
Théorème 2. Soient N oscillateurs disposés en chaîne d'influence :

$$O_1 \longrightarrow O_2 \longrightarrow O_3 \longrightarrow \dots \longrightarrow O_{N-1} \longrightarrow O_N$$

Dans cette chaîne, pour tout i de $\llbracket 2; N \rrbracket$, l'oscillateur O_i n'est influencé que par O_{i-1} . On obtient :

$$\boxed{\forall i \in \llbracket 1; N \rrbracket, \lim_{t \rightarrow +\infty} \theta_i(t) - \theta_1(t) = 0}$$

Corollaire 1. Ce résultat est encore valable lorsque les oscillateurs sont disposés en arbre d'influence, par exemple :



On obtient encore :

$$\forall i \in \llbracket 2; N \rrbracket, \lim_{t \rightarrow +\infty} \theta_i(t) - \theta_1(t) = 0$$

Deuxième partie

Expériences

3 Conception d'une expérience

Pour tenter de confirmer les résultats théoriques précédemment établis, j'ai eu deux idées :

- Tenter d'influencer les applaudissements d'une salle non-avertie, en installant des compères dans une salle qui applaudiraient en rythme dès le début et essaieraient de propager leur synchronisation ;
- Réaliser un dispositif mécanique, électronique ou informatique pour simuler des individus.

La première idée était tentante. Elle nécessitait beaucoup de préparation et de matériel : il faudrait par exemple idéalement disposer plusieurs micros pour étudier la dynamique de la salle. La pandémie actuelle rend difficile l'organisation de telles expériences car les salles de concert sont fermées.

J'ai donc opté pour le dispositif mécanique. L'objet se rapprochant le plus d'un oscillateur naturel est le métronome mécanique. J'ai choisi d'en utiliser quatre, que j'ai fixés sur une planche de bois. Pour permettre l'influence mutuelle des métronomes, j'ai placé cette planche de bois sur deux tubes métalliques identiques afin de lui laisser un degré de liberté. L'idée est d'observer le mouvement des métronomes, lorsqu'ils sont réglés sur la même fréquence et lancés au hasard, en espérant observer leur synchronisation suffisamment durable. Les métronomes mécaniques sont équipés d'une masselotte qui permet de régler leur fréquence, et qui de plus permet l'équilibre du système. J'ai retiré cette masselotte pour que chaque métronome induise un mouvement du support. Ensuite, comme les métronomes ne sont pas rigoureusement identiques à cause de défauts de fabrication, j'ai utilisé des pastilles adhésives pour les calibrer. A l'aide du logiciel Phyphox, qui possède un chronomètre sonore, j'ai ajusté les quantités de pâte que j'ai collée sur le balancier afin que chaque métronome, privé de sa masselotte, batte à environ 275 battements par minute. L'analyse des battements par ce même logiciel m'a permis de détecter un phénomène que je n'attendais pas : la régularité entre deux battements successifs évolue dans le temps. Deux battements successifs n'ont en effet jamais rigoureusement la même période à cause de la nature du mécanisme utilisé qui n'est pas symétrique. Cependant, on observe des moments où l'irrégularité est forte, et d'autres où les battements sont très réguliers. Ce phénomène semble, de plus, périodique. Je pense qu'il est dû à la rotation d'un engrenage interne défectueux (voir la figure ??).

J'ai filmé le tout et effectué un pointage de la position du sommet de chaque balancier. Comme la planche sur laquelle les métronomes reposent bouge très peu par rapport aux balanciers, j'ai décidé de négliger son mouvement. Je considère donc que chaque métronome est immobile dans le traitement des données.

4 Réalisation des expériences

J'ai utilisé une planche de $20 \times 86 \times 445$ mm, de masse 625 g et deux rouleaux en inox d'une circonférence de 16 mm, de masse 45 g. Le tout est éclairé d'une barre de leds blanches, et le fond utilisé est un voile gris uniforme. J'ai filmé avec un appareil photo

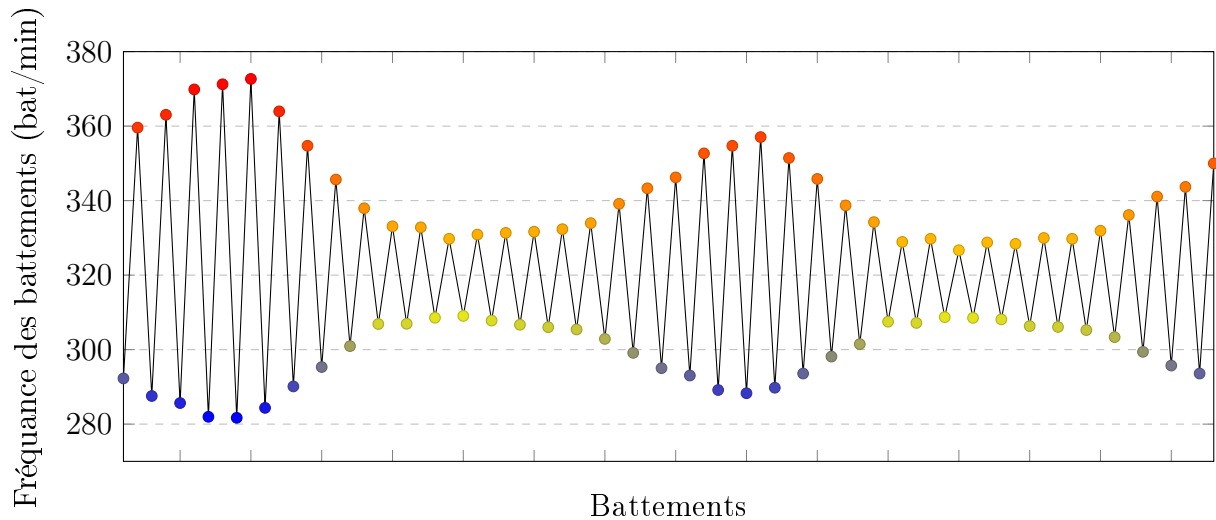


FIGURE 2 – Durée battements consécutifs d’un métronome, qui varie de au cours du temps. Il y a une alternance entre des périodes où les battements sont relativement réguliers (0,0125 secondes de différence) et des périodes où l’irrégularité est plus grande (0,052 secondes de différence), et ce phénomène semble périodique.

numérique Fuji x20 réglé sur 80 images par seconde.

5 Traitement et analyse des résultats

5.1 Fonctionnement du programme utilisé

Le traitement des images a été réalisé à l’aide d’un programme Python que j’ai mis au point qui détecte les cercles. L’idée pour suivre efficacement chaque métronome est de placer une cible circulaire sur le haut de chaque balancier, et de trouver le centre de ces cibles sur chaque image de la vidéo. Pour tenter d’obtenir du contraste, j’ai réalisé plusieurs cibles colorées plus ou moins compliquées, mais les résultats n’étaient pas satisfaisants. Je me suis finalement contenté de cibles blanches. En réalisant un fort contraste entre les cibles et le reste de l’image, on parvient, en convertissant les images en binaire (noir *ou* blanc) à ne conserver sur l’image que les cibles visibles.

Le programme utilisé convertit chaque image de la vidéo en image binaire et détermine la position des cercles qui apparaissent blancs sur un fond noir. La sensibilité de cette conversion peut être réglée de sorte à n’obtenir à la fin que les cercles. Une des difficultés est d’initialiser la recherche. Pour ce faire, une transformée de Hough circulaire est effectuée sur la première image (voir paragraphe 5.2). On obtient ainsi une liste de coordonnées qui correspondent aux centres des cercles présents. Pour les images suivantes, le traitement est plus simple : on fait la moyenne des coordonnées des points blancs trouvés dans un voisinage des centres de cercles. Ainsi, comme les parcours des balanciers ne se croisent pas, on est sûr par cette méthode de ne prendre en compte que les bons points.

La rapidité est satisfaisante : environ 2 minutes pour une vidéo de 2400 images comportant quatre métronomes, soit 80 centres de cercles calculés par seconde.

5.2 La transformée de Hough circulaire

La transformée de Hough [3] est une méthode permettant de détecter une forme précise dans une image. Le programme Python utilise ici une transformée de Hough circulaire qui permet de détecter les cercles présents sur l'image. Dans notre cas, pour une image binaire où le fond est noir et dont les éventuels cercles sont blancs, la recherche de cercles de rayon R fixé se fait selon le principe suivant :

On calcule d'abord le gradient de l'image pour ne garder que les discontinuités dans la couleur. On obtient alors des lignes fermées blanches sur fond noir. On définit une "matrice d'accumulation" de mêmes dimensions que l'image, dont toutes les valeurs sont initialisées à 0. Pour chaque point blanc dans l'image, on incrémente de 1 chaque case de la matrice dont la distance au point blanc vaut R . Cela fait, on recherche les maxima locaux dans la matrice : ce sont les centres des cercles de rayon R .

Il suffit d'appliquer cet algorithme avec plusieurs valeurs de R pour trouver tous les cercles dont le rayon est dans une certaine plage.

5.3 Format des données obtenues

Le programme Python fournit des listes de coordonnées du type :

Numéro n de l'image	$X1(n)$	$Y2(n)$	$X2(n)$	$Y2(n)$	...
0	28	16	152	17	...
1	29	17	150	18	...
2	30	17	149	17	...
3	29	17	148	17	...
4	27	16	146	16	...

Le sommet de chaque balancier décrit bien un arc de cercle, mais je néglige la mesure des hauteurs car elles ne me semblent pas importantes : elles varient peu, et je souhaite simplement obtenir un signal périodique. Je ne garde ainsi que les mesures de $x(t)$. Le graphe de $(t, x(t))$ ressemble d'ailleurs assez à une fonction sinusoïdale (voir la figure 3). Enfin, une fois que les positions sont déterminées, je calcule pour chaque métronome la moyenne globale de ses positions en x et y que je soustrais à l'ensemble des mesures. J'obtiens donc pour chaque métronome des mesures en x et y centrées en 0. Cela permet de superposer virtuellement les métronomes.

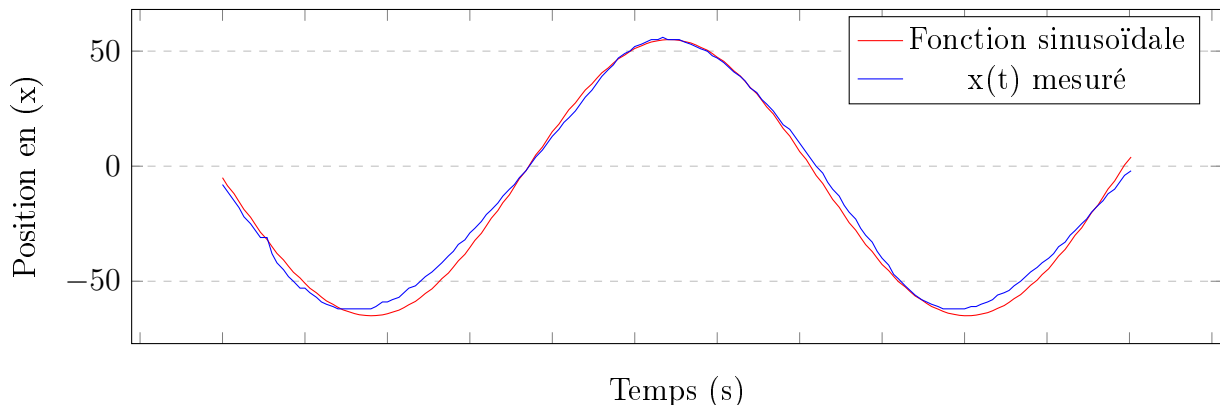
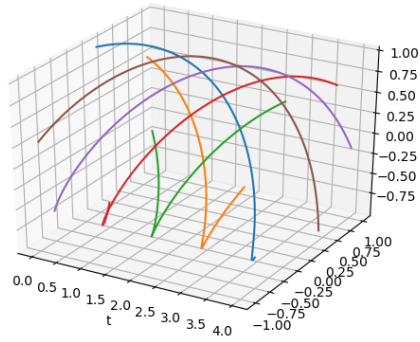


FIGURE 3 – Comparaison entre le mouvement du balancier et une fonction sinusoïdale

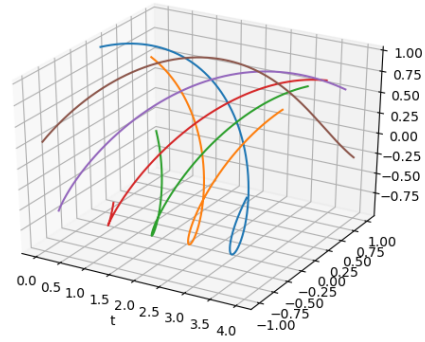
Troisième partie

Annexes

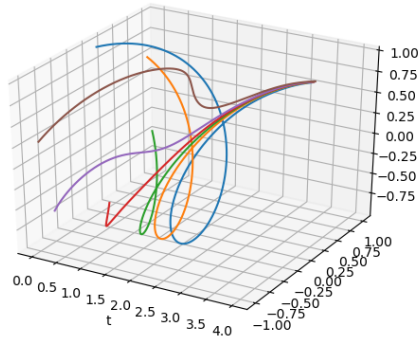
6 Simulation de différents couplages pour les mêmes conditions initiales



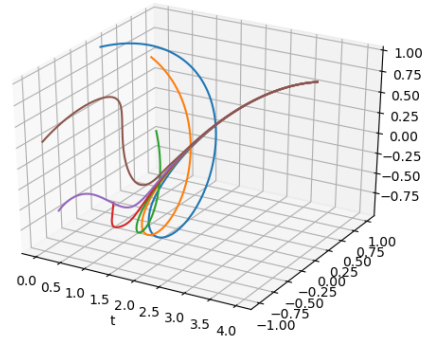
(a) $k=0$



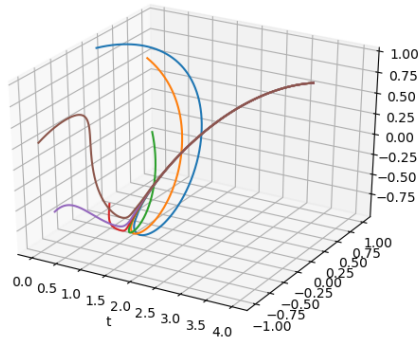
(b) $k=0,2$



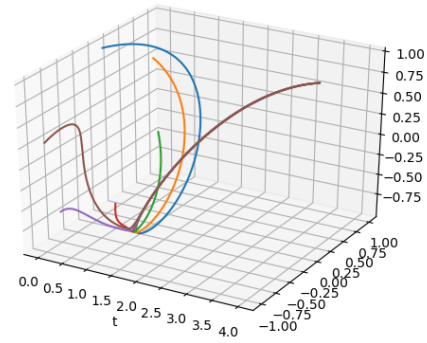
(c) $k=0,4$



(d) $k=0,6$



(e) $k=0,8$



(f) $k=1$

FIGURE 4 – Phases de départ : 0,1,2,3,4 et 5 radians, fréquence naturelle : 1 rad/s
Nous constatons que la synchronisation est d'autant plus rapide que le couplage est important. Celle-ci n'a pas lieu si le couplage est nul, les oscillateurs étant dans ce cas indépendants les uns des autres.

7 Détails des expériences

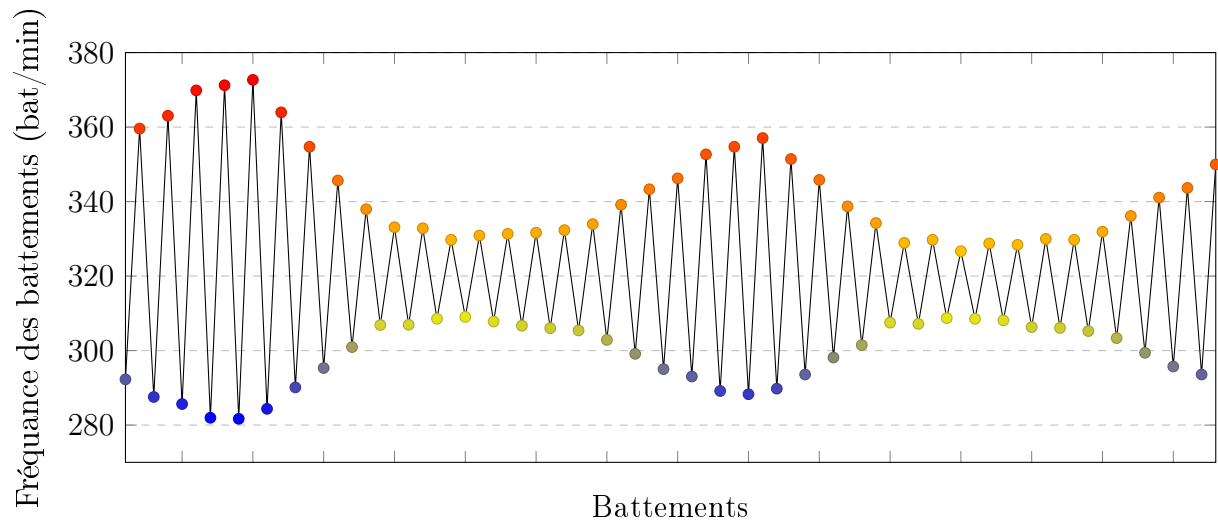


FIGURE 5 – Durée battements consécutifs d'un métronome, qui varie de au cours du temps. Il y a une alternance entre des périodes où les battements sont relativement réguliers (0,0125 secondes de différence) et des périodes où l'irrégularité est plus grande (0,052 secondes de différence), et ce phénomène semble périodique.

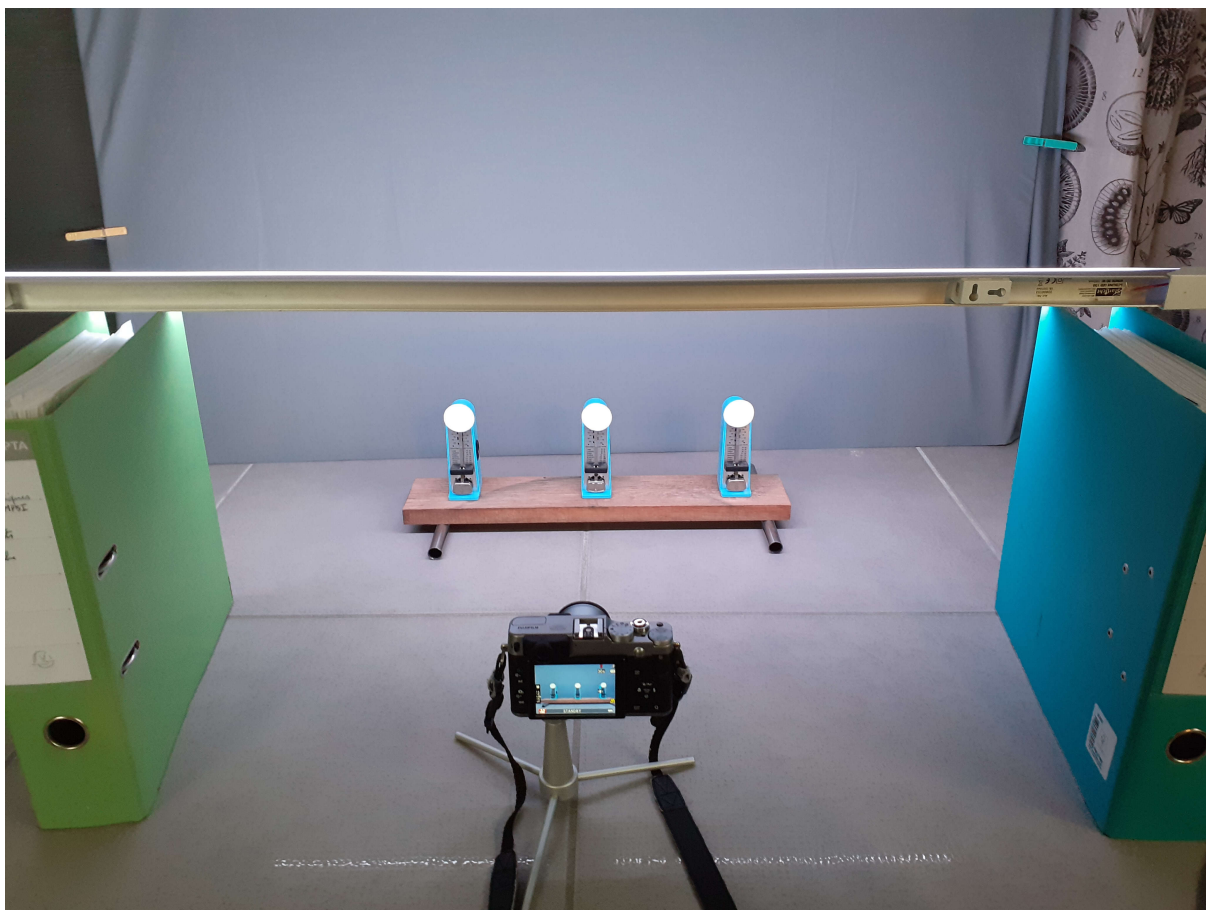


FIGURE 6 – Mise en place de l'expérience

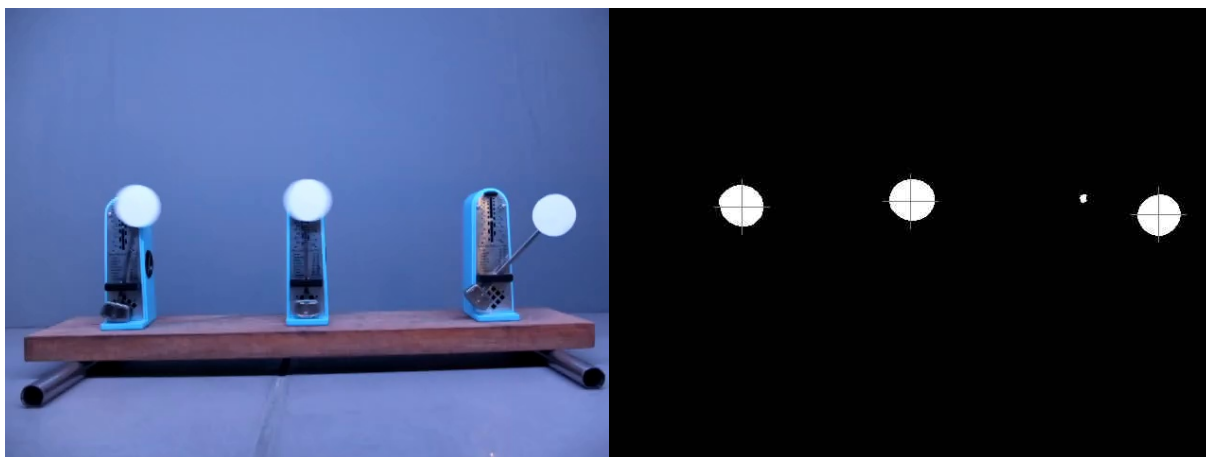


FIGURE 7 – Image initiale puis convertie en image binaire, avec les centres des cercles calculés par le programme

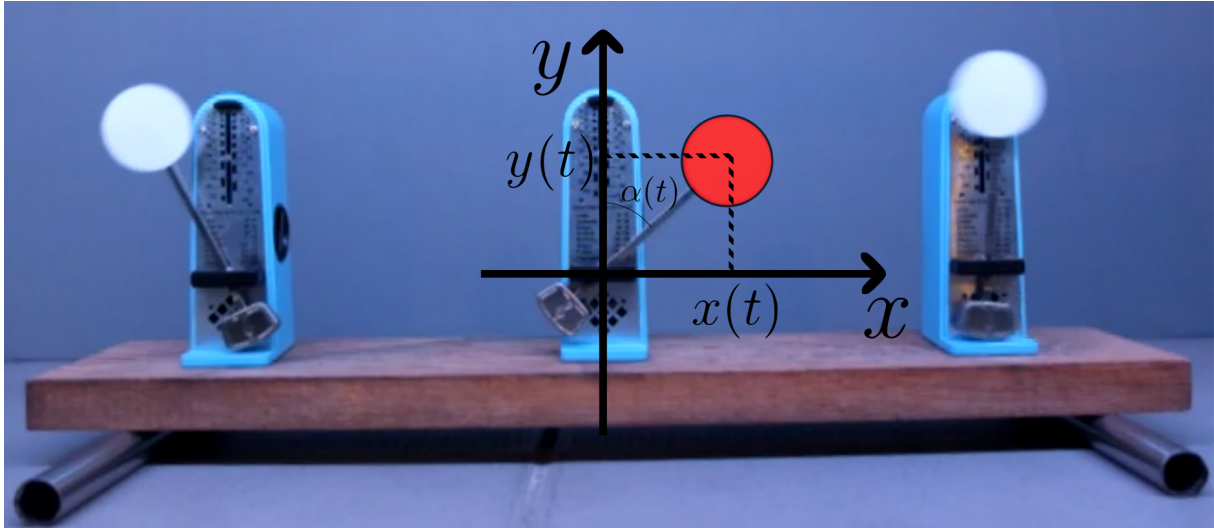


FIGURE 8 – Axes du métronome du centre

8 Données obtenues lors des expériences

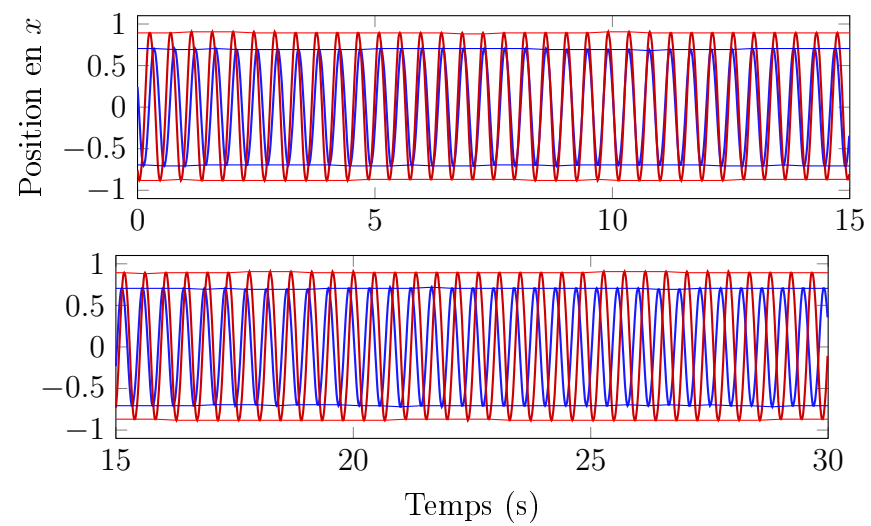
Les 3 métronomes sont réglés sur la même fréquence.

Expérience 1 : ils sont lancés au hasard et non synchronisés dès le départ. On observe une synchronisation progressive en amplitude et fréquence. A la fin, l'état de synchronisation est stable (non montré sur les graphiques).

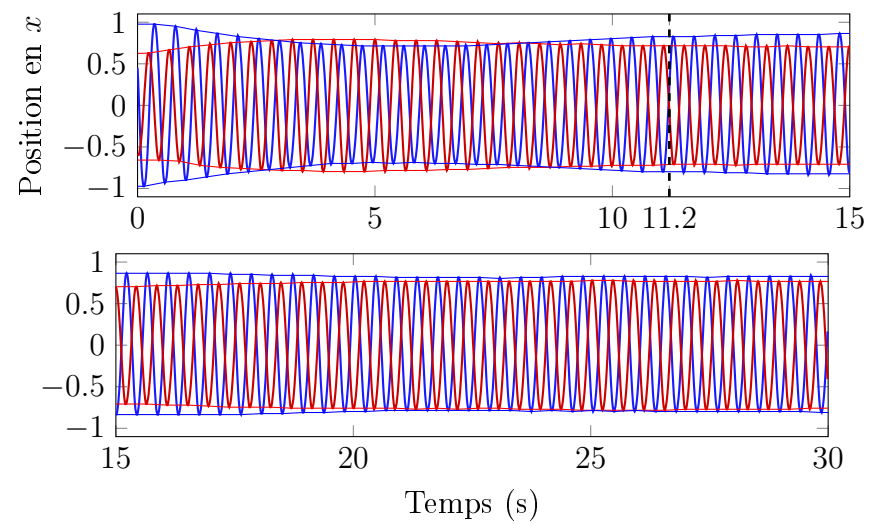
Expérience 2 : ils sont lâchés de sorte que deux soient synchronisés et que le troisième soit en opposition de phase par rapport aux autres.

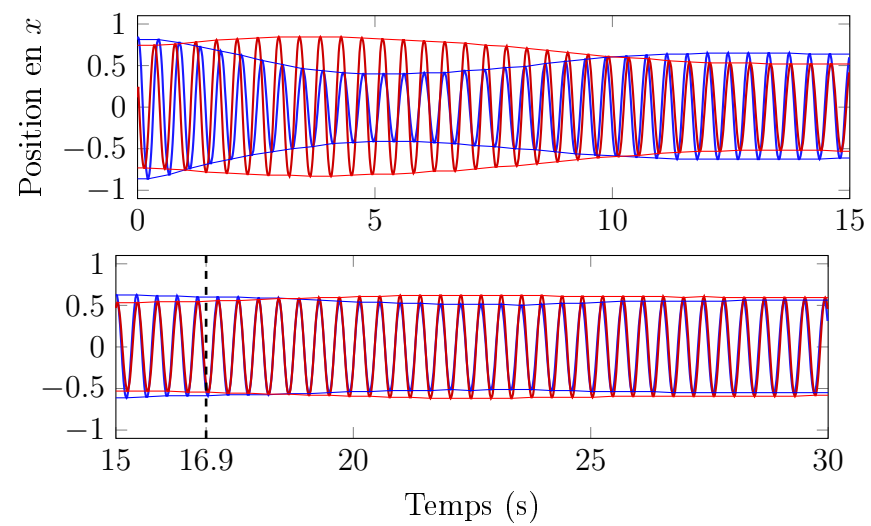
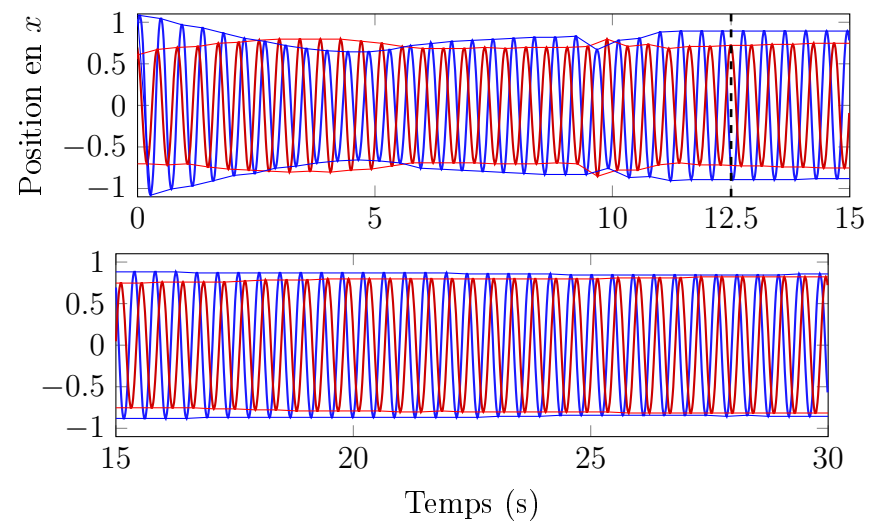
Expérience 3 : lancés au hasard, on constate que l'amplitude de l'un domine celles des autres. Au bout d'une dizaine de périodes, le métronome du milieu s'arrête complètement, puis conserve un mouvement uniquement dû aux mouvements des autres métronomes car il demeure silencieux.

Expérience 1, $N = 2$ - sans rouleaux

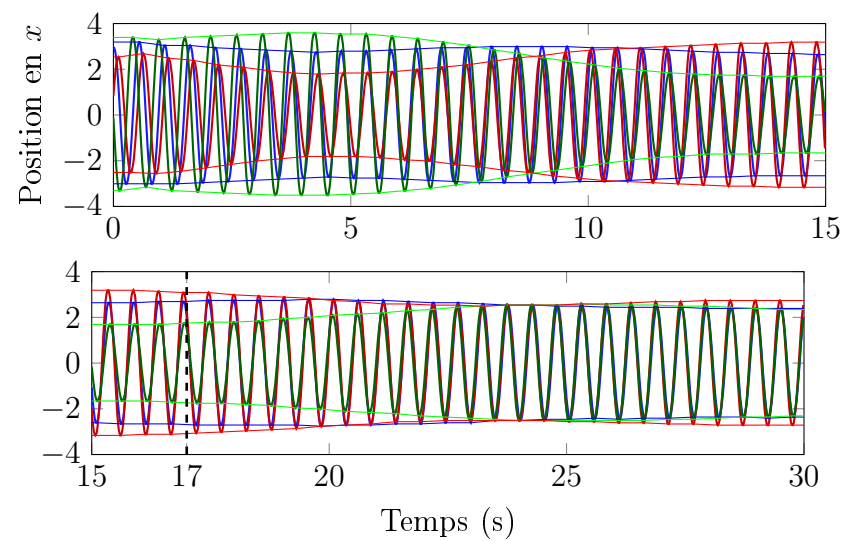


Expérience 2 : $N = 2$, rouleaux

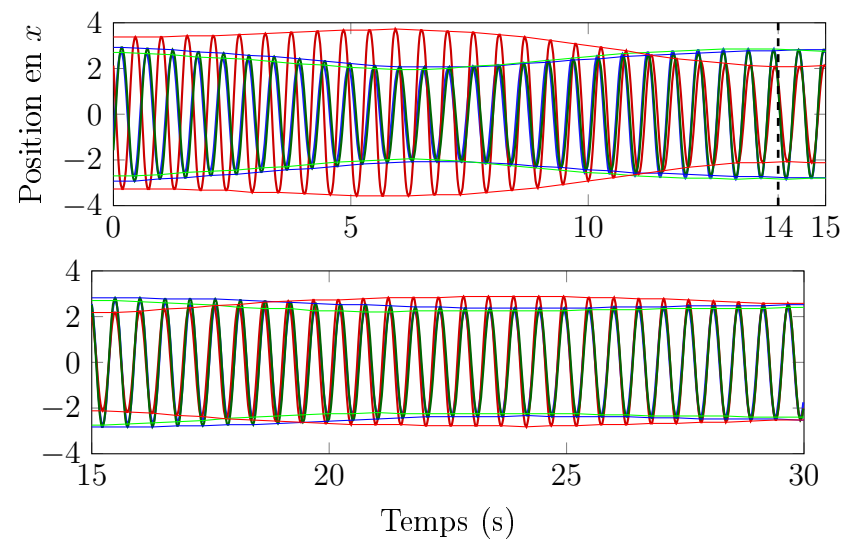


Expérience 3 : $N = 2$, rouleauxExpérience 4 : $N = 2$, rouleaux

Expérience 5 : $N = 3$, rouleaux



Expérience 6 : $N = 3$, rouleaux



9 Preuve des théorèmes

9.1 Stabilité du modèle de Kuramoto pour une équirépartition des phases

Si N oscillateurs possèdent des phases $\theta, \theta + \frac{2\pi}{N}, \dots, \theta + (N-1)\frac{2\pi}{N}$, on obtient :

$$\begin{aligned} \sum_{k=1}^N \Gamma_{ik}(\theta_k - \theta_i) &= \sum_{k=1}^N \sin(\theta + k\frac{2\pi}{N} - \theta) \\ &= \Im \left(\sum_{k=1}^N e^{i\frac{2k\pi}{N}} \right) \\ &= 0 \end{aligned}$$

9.2 Première approche naïve du théorème 1 :

D'après Kuramoto, on obtient le système suivant, en supposant que θ_1 définit la phase à l'origine :

$$\begin{cases} \dot{\theta}_1 = \omega_1 \\ \dot{\theta}_2 = \omega_2 + \frac{k}{d^2} \sin(\theta_1 - \theta_2) \end{cases}$$

Pour de petits angles, on utilise l'approximation $\sin x \approx x$, et on obtient l'équation à intégrer (on note $K = \frac{k}{d^2}$) :

$$\dot{\theta}_2 + K\theta_2 = K\omega_1 t + \omega_2$$

La résolution par la méthode de variation de la constante donne :

$$\theta_2(t) = \frac{1}{K}\omega_2 + \omega_1 t - \frac{\omega_1}{K}(1 - e^{-Kt}) + \lambda_0 e^{-Kt}$$

où λ_0 est un paramètre réel, c'est-à-dire

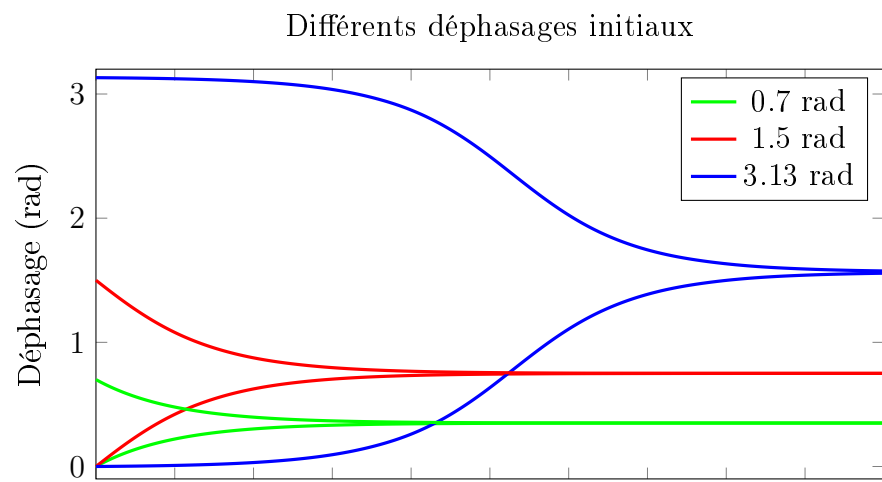
$$\theta_2(t) = \frac{\omega_2 - \omega_1}{K} + \omega_1 t + \left(\frac{\omega_1}{K} + \lambda_0\right)e^{-Kt}$$

Et donc, dans l'hypothèse où $\omega_1 \approx \omega_2$, on obtient :

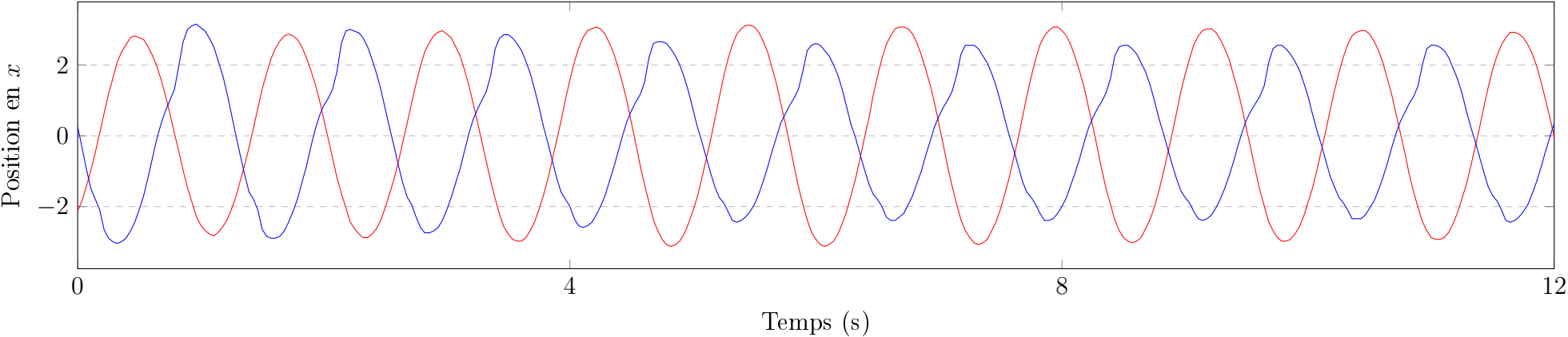
$$\lim_{t \rightarrow +\infty} \theta_2(t) - \omega_1 t = 0$$

c'est-à-dire :

$$\lim_{t \rightarrow +\infty} \theta_2(t) - \theta_1(t) = 0$$



$N = 2$ - instable 2



9.3 Preuve rigoureuse du théorème 1 :

On considère le système suivant :

$$\begin{cases} \theta_1 = \omega_1 t \\ \dot{\theta}_2 = \omega_2 + \frac{k}{d^2} \sin(\theta_1 - \theta_2) \end{cases}$$

On a supposé que θ_1 a une phase nulle à l'origine. On fait de plus l'hypothèse que les oscillateurs ont la même fréquence ω qui reste constante : $\omega = \omega_1 = \omega_2$. On note enfin $K = \frac{k}{d^2}$. Par un changement de fonction inconnue en posant

$$\alpha(t) = \theta_2(t) - \omega t$$

on obtient le résultat suivant :

$$\begin{aligned} \frac{d}{dt}(\theta_2(t) - \omega t) &= -K \sin(\theta_2(t) - \omega t) \\ \frac{d\alpha}{dt} &= -K \sin(\alpha) \\ \frac{d\alpha}{\sin \alpha} &= -K dt \\ \int \frac{d\alpha}{\sin \alpha} &= -K \int dt \\ \ln \left| \tan \frac{\alpha}{2} \right| &= -Kt + C \\ \tan \frac{\alpha}{2} &= \pm \exp(-Kt + C) \\ \alpha &= \pm 2 \arctan \exp(-Kt + C) \\ \theta_2(t) &= \omega t \pm 2 \arctan \exp(-Kt + C) \end{aligned}$$

En notant φ la phase à l'origine, qui est par hypothèse différente de π , on obtient, si ¹ $\varphi \neq 0$ l'expression de la constante d'intégration C :

$$C = \ln \tan \frac{\varphi}{2}$$

qui est donc bien réelle. Ainsi :

$$\theta_2(t) = \omega t \pm 2 \arctan \left(\tan \left(\frac{\varphi}{2} \right) \exp(-Kt) \right)$$

$$\boxed{\lim_{t \rightarrow +\infty} \theta_2(t) - \theta_1(t) = 0}$$

Le théorème est ainsi démontré. On constate la cohérence de l'expression de la fonction θ_2 : la quantité $2 \arctan \exp(-Kt + \varphi)$ est à valeurs dans $[0, \pi[$: elle représente bien une phase.

1. Dans le cas contraire, on aurait $\theta_1 = \theta_2$, les oscillateurs seraient synchronisés dès le départ et il n'y aurait rien à démontrer.

9.4 Preuve du théorème 2 :

On utilise le raisonnement par récurrence. D'après le théorème 1, O_2 se synchronise avec O_1 . Montrons que O_3 se synchronise avec O_1 , c'est-à-dire $\lim_{t \rightarrow +\infty} \theta_3(t) - \theta_1(t) = 0$. On a par hypothèse l'existence d'une fonction ϵ_2 telle que $\theta_2(t) = \theta_1(t) + \epsilon_2(t)$ avec $\lim_{t \rightarrow +\infty} \epsilon_2(t) = 0$. D'après le modèle de Kuramoto, la chaîne (θ_2, θ_3) vérifie le système

$$\begin{aligned} & \begin{cases} \theta_1(t) = \omega_1 t \\ \theta_2(t) = \omega_1 t + \epsilon_2(t) \\ \dot{\theta}_3 = \omega_3 + K \sin(\theta_2(t) - \theta_3) \end{cases} \\ \implies & \begin{cases} \theta_2(t) = \omega_1 t + \epsilon_2(t) \\ \dot{\theta}_3 = \omega_3 + K \sin(\omega_1 t + \epsilon_2(t) - \theta_3) \end{cases} \\ \implies & \begin{cases} \theta_2(t) = \omega_1 t + \epsilon_2(t) \\ \dot{\theta}_3 + K\theta_3 = \omega_3 + K\omega_1 t + K\epsilon_2(t) \end{cases} \end{aligned}$$

La résolution par méthode de variation de la constante donne

$$\begin{aligned} \theta_3(t) &= \frac{\omega_3 - \omega_1}{K} + \omega_1 t + \left(\frac{\omega_1}{K} + \lambda_0\right)e^{-Kt} + Ke^{-Kt} \int_0^t \epsilon_2(x)e^{Kx} dx \\ &= \frac{\omega_3 - \omega_1}{K} + \omega_1 t + \left(\frac{\omega_1}{K} + \lambda_0\right)e^{-Kt} + \mathfrak{o}(1) \\ &= \frac{\omega_3 - \omega_1}{K} + \omega_1 t + \mathfrak{o}(1) \end{aligned}$$

où λ_0 est un paramètre réel. Comme $\omega_3 = \omega_1$, on obtient

$$\lim_{t \rightarrow +\infty} \theta_3(t) - \theta_1(t) = 0$$

Donc il existe une fonction ϵ_3 tendant vers 0 telle que

$$\theta_3(t) = \theta_1(t) + \epsilon_3(t)$$

Cela se généralise pour toute chaîne finie, en considérant pour chaque couple d'oscillateurs (θ_i, θ_{i+1}) une nouvelle fonction ϵ_{i+1} , définie comme précédemment et tendant vers 0.

9.5 Preuve du corollaire 2 :

Pour un oscillateur O_i , on considère la chaîne de ses ancêtres qui remonte jusqu'à O_0 . Il suffit alors d'appliquer le théorème 2 à cette chaîne.

10 Code source des programmes python

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 import numpy as np
4 import copy as cp
5 import cv2
6 import os
7
```

```

8 def stamp(frame,x,y):
9     global rayon_recherche
10    frame[x][y]=128
11    for d in range(rayon_recherche):
12        frame[x-d][y]=128
13        frame[x+d][y]=128
14        frame[x][y-d]=128
15        frame[x][y+d]=128
16    return(frame)
17 def centreMoyenne(image,x,y):
18     compteur=0
19     h,l=len(image),len(image[0])
20     a,b=0,0
21     global rayon_recherche
22     n=rayon_recherche
23     #print(x,y)
24     for i in range(2*n):
25         for j in range(2*n):
26             if x+i-n>=0 and x+i-n<h and y+j-n>=0 and y+j-n<l:
27                 if image[x+i-n][y+j-n]==255:
28                     compteur+=1
29                     a+=x+i-n
30                     b+=y+j-n
31     a/=compteur
32     b/=compteur
33     return float2int([a,b])
34
35 def float2int(l):
36     return [int(x) for x in l]
37 ## Initialisation
38 path='/home/guilhem/Desktop/TIPE/Travail perso/Detection cercle/images/'
39 nomImage='expl6mai/synch1.MOV'
40 cam = cv2.VideoCapture(path+nomImage)
41 rayon_recherche=50
42 flou_gaussien=5
43 try:
44     if not os.path.exists(path+'data'):
45         os.makedirs(path+'data')
46 except OSError:
47     print ('Error: Creating directory of data')
48 ## Detection des premiers cercles par Hough
49 cam = cv2.VideoCapture(path+nomImage)
50 existeImage,frame = cam.read()
51 existeImage,frame = cam.read()
52 existeImage,frame = cam.read()
53
54
55 name = path+'data/metronomes_couleur.jpg'
56 cv2.imwrite(name, frame)
57
58 currentframe = 0
59 frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
60 th, frame = cv2.threshold(frame, 210, 255, cv2.THRESH_BINARY)
61 frame = cv2.medianBlur(frame, flou_gaussien)
62 circles = cv2.HoughCircles(frame, cv2.HOUGH_GRADIENT, 1, frame.shape
63 [0]/64, param1=200, param2=10, minRadius=20, maxRadius=50)
64 if circles is not None:
65     circles = np.uint16(np.around(circles))

```

```

65     print(circles)
66     circles=[[cercle[1],cercle[0]] for cercle in circles[0]]
67     liste_coord=[[i] for i in circles]
68     nombreCercles=len(liste_coord)
69     for c in circles:
70         frame=stamp(frame, c[0],c[1])
71     cv2.imshow('detected circles',frame)
72     cv2.waitKey(0)
73     cv2.destroyAllWindows()
74     name = path+'data/metronomes_binaire.jpg'
75     cv2.imwrite(name, frame)
76 else:
77     print("Aucun cercle detecte !")
78 ## Traitement
79 for cercle in range(nombreCercles):
80     print("Cercle numero", cercle)
81     currentframe = 1
82     cam = cv2.VideoCapture(path+nomImage)
83     existeImage,frame = cam.read()
84     existeImage,frame = cam.read()
85     compteur_images=1
86     while(existeImage):
87         if currentframe>=2*compteur_images:
88             compteur_images=currentframe
89             print('Image '+str(currentframe)+' en cours de traitement')
90             frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
91             th, frame = cv2.threshold(frame, 200, 255, cv2.THRESH_BINARY)
92             frame = cv2.medianBlur(frame, flou_gaussien)
93             liste_coord[cercle].append(centreMoyenne(frame,liste_coord[
cercle][-1][0],liste_coord[cercle][-1][1]))
94             currentframe+=1
95             # name = path+'data/image' + str(currentframe) + '.jpg'
96             # cv2.imwrite(name, frame)
97             existeImage,frame = cam.read()
98             print("Il y a", currentframe, "images !")
99 nombreImages=currentframe
100 print("Merci pour ce moment")
101 ## Marquage des images
102 currentframe=0
103 cam = cv2.VideoCapture(path+nomImage)
104 existeImage,frame = cam.read()
105 while(existeImage):
106     for cercle in range(nombreCercles):
107         coordonnee=liste_coord[cercle][currentframe]
108         if coordonnee[0]>0 and coordonnee[0]<len(frame) and coordonnee
[1]>0 and coordonnee[1]<len(frame[0]):
109             frame=stamp(frame,coordonnee[0],coordonnee[1])
110             name = path+'data/image' + str(currentframe) + '.jpg'
111             cv2.imwrite(name, frame)
112             currentframe += 1
113             print(currentframe)
114             existeImage,frame = cam.read()
115 cam.release()
116 cv2.destroyAllWindows()
117
118 ## Liste moyennisee
119 listeMoyenne=[[np.average([liste_coord[cercle][im][c] for im in range(
nombreImages)]) for c in range(2)] for cercle in range(nombreCercles)

```

```

]
120 listeCopie=cp.deepcopy(liste_coord)
121 for cercle in range(nombreCercles):
122     for im in range(nombreImages):
123         for c in range(2):
124             listeCopie[cercle][im][c]-=listeMoyenne[cercle][c]
125             listeCopie[cercle][im][c]/=80
126 ## Lissage + Maxima et minima
127 import numpy as np
128 from scipy.signal import argrelextrema
129 from scipy.signal import savgol_filter
130
131 <<
132 for c in range(nombreCercles):
133     listeY=[listeCopie[c][x][1] for x in range(len(listeCopie[0]))]
134     listeY=savgol_filter(listeY, 51, 10)
135     # Ruse pour supprimer deux maxima consecutifs : on retire 10^-10
    au second
136     for i in range(len(listeY)-1):
137         if listeY[i]==listeY[i+1]:
138             listeY[i+1]=listeY[i+1]-10**(-10)
139     listeMax.append(argrelextrema(np.array(listeY), np.greater))
140     listeMin.append(argrelextrema(np.array(listeY), np.less))
141 ## Trace un graphique
142 debut,fin=0,nombreImages
143 t = range(debut,fin)
144 s=[]
145 fig, ax = plt.subplots()
146 for c in range(nombreCercles):
147     l=[listeCopie[c][x][1] for x in range(debut,fin)]
148     ax.plot(t,l,linewidth=1)
149     # pour les maxs :
150     max=listeMax[c][0]
151     t1=[t[m] for m in max]
152     l1=[l[m] for m in max]
153     ax.scatter(max,l1)
154     # pour les mins :
155     min=listeMin[c][0]
156     t1=[t[m] for m in min]
157     l1=[l[m] for m in min]
158     ax.scatter(t1,l1)
159
160 ax.set(xlabel='Temps (numero d\'image)', ylabel='Position (x)', title=
    str(nombreCercles)+' metronomes')
161 ax.grid()
162 plt.show()
163
164
165 ### Calcul du parametre d'ordre dynamique
166 import matplotlib.pyplot as plt
167 import matplotlib.image as mpimg
168 import numpy as np
169 import copy as cp
170 import cv2
171 import os
172 import scipy.integrate as integrate
173 import math
174

```

```

175 moyenne_des_signaux=[]
176 t = range(debut,fin)
177 for x in t:
178     moyenne_des_signaux.append(listeCopie[0][x][1]-listeCopie[1][x][1])
179
180 def s(x):
181     return moyenne_des_signaux[int(x)]
182
183 fig, ax = plt.subplots()
184 ax.plot(t,moyenne_des_signaux,linewidth=1)
185
186 plt.title("Exp rience 5 : diff rence des signaux")
187
188 ax.grid()
189 plt.show()
190
191 ##
192 Tmin=5
193 Tmax=10
194 pas=10
195
196
197 # De 0 a 2pi
198 phi = [2*np.pi*i/pas for i in range(pas)]
199
200
201 pas=10
202 phi = [2*np.pi*i/pas for i in range(pas)]
203
204 a = 2
205
206 b = 1
207 T=Tmin
208 A = integrate.quad(lambda x: math.sin(2*math.pi/T*x+phi[i])*s(x), 0, 1)
209
210 I = quad(integrand, 2, 3, args=(a,b))
211
212
213
214 ### Creer une video
215 import glob
216 cam = cv2.VideoCapture(path+nomImage)
217 existeImage,frame = cam.read()
218 frameSize = (len(frame[0]), len(frame))
219 out = cv2.VideoWriter(path+nomImage+'_stamped.avi',cv2.
    VideoWriter_fourcc(*'DIVX'), 80, frameSize)
220
221 for i in range(nombreImages):
222     #print(path+'data/image'+str(i)+'.jpg')
223     img = cv2.imread(path+'data/image'+str(i)+'.jpg')
224     out.write(img)
225 out.release()
226 ### Affichage en PGFplot latex de listeCopie
227 temps_video=30
228
229 for c in range(nombreCercles):
230     chr=' '
231     for t in range(nombreImages):

```

```

232         chr+='(' +str(round(((t*temps_video)/nombreImages),3))+' ,'+str(
round(listeCopie[c][t][1],3))+' )'
233     print(chr)
234     print(" ")
235     print(" ")
236     print(" ")
237     print(" ")
238     print(" ")
239
240 ### Affichage en PGFplot LaTeX de l'enveloppe
241 ## Maxima
242 for c in range(nombreCercles):
243     chr=''
244     for t in range(len(listeMax[c][0])):
245         l=[listeCopie[c][x][1] for x in range(debut,fin)]
246         chr+='(' +str(round((listeMax[c][0][t]*temps_video)/nombreImages
,3))+' ,'+str(round(l[listeMax[c][0][t]],3))+' )'
247     print(chr)
248     print(" ")
249     print(" ")
250     print(" ")
251     print(" ")
252     print(" ")
253
254 ## Minima
255 for c in range(nombreCercles):
256     chr=''
257     for t in range(len(listeMin[c][0])):
258         l=[listeCopie[c][x][1] for x in range(debut,fin)]
259         chr+='(' +str(round((listeMin[c][0][t]*temps_video)/nombreImages
,3))+' ,'+str(round(l[listeMin[c][0][t]],3))+' )'
260     print(chr)
261     print(" ")
262     print(" ")
263     print(" ")
264     print(" ")
265     print(" ")

```

Références

- [1] Laurie CONTEVILLE : Analyse de la stabilité des réseaux d'oscillateurs non linéaires, applications aux populations neuronales. 2013.
- [2] Bryan C. DANIELS : Synchronization of globally coupled nonlinear oscillators : the rich behavior of the kuramoto model. May 2005.
- [3] V. K. Yadav ; S. Batham ; A. K. Acharya ; R. PAUL : Approach to accurate circle detection : Circular hough transform and local maxima concept. *International Conference on Electronics and Communication Systems*, 2014.
- [4] Steven H. STROGATZ : From kuramoto to crawford : exploring the onset of synchronization in populations of coupled oscillators. June 2000.
- [5] WIKIPEDIA : Electric power transmission.