

# Simulation d'un ordinateur mécanique en scriptant sous FreeCAD ou comment dessiner des carrés avec des ronds

J.-M Friedt<sup>1</sup>, O. Testault<sup>2</sup>, É. Carry<sup>1</sup>

<sup>1</sup> Institut FEMTO-ST, Université de Franche-Comté, Besançon

<sup>2</sup> FabLab des Fabriques, Besançon

11 décembre 2020

## 1 Introduction

L'évolution du traitement du signal est une histoire fascinante largement déroulée par David Mindell dans ses divers ouvrages [1] et citations [2]. Partant de l'ordinateur mécanique avec ses rouages, poulies, bielles et crémaillères, le passage à l'électrique au début du 20ème siècle puis à l'électronique intégrée avec l'avènement du transistor et les circuits intégrés (VLSI) nous ont fait oublier les stades initiaux qui ont amené à notre statut actuel d'ordinateurs infiniment puissants, précis et compacts. Alors que cette histoire semble s'accompagner du passage de l'analogique au numérique – de la manipulation de grandeurs continues en grandeurs discrètes avec son gain en stabilité et reproductibilité – il n'en est en fait rien : un boulier fournit déjà les bases du calcul discrétisé mécanique, tandis que [3] *introduit* les concepts du calcul mécanique avec les traitements numériques avant de passer aux traitements analogiques.

De tels dispositifs ont été utilisés en pratique jusqu'à la seconde guerre mondiale, tel qu'en atteste [4] dans le contexte de l'étude des méthodes cryptographiques en mentionnant "The navy had asked the practical engineers at the Navy Yard under Don Seiler for help fighting code systems before the war began. By 1940 he built a fifty-wheel cam and gear device that was driven by hand cranks. It tested cipher and additives for divisibility by three."

Notre curiosité sur les ordinateurs mécaniques a été attisée par l'analyseur harmonique de Michelson (le même qui a monté l'interféromètre optique pour démontrer l'indépendance de la vitesse de la lumière avec la direction de déplacement de la Terre dans l'Espace). De façon générale, tout système linéaire peut s'exprimer dans le domaine spectral comme somme de composantes harmoniques par série de Fourier, fournissant une méthode de résolution en régime permanent bien plus efficace qu'une résolution dans le domaine temporel.

Pour rappel, l'analyse de Fourier s'appuie sur le fait que les combinaisons linéaires de fonctions périodiques conservent nécessairement les composantes spectrales du signal incident puisque  $A \exp(j\omega_1 t) + B \exp(j\omega_2 t)$  ne saurait contenir d'autres composantes spectrales que  $\omega_1$  et  $\omega_2$ . L'invariance dans le temps des coefficients  $A$  et  $B$  garantit que la solution est stationnaire : il s'agit de la classe des systèmes linéaires invariants en temps, ou LTI (*Linear Time Invariant*)

Parmi les problèmes célèbres qui se résolvent selon cette approche, la position des astres dans le ciel ou la hauteur des marées ont eu un rôle social ou stratégique (date du débarquement concluant la seconde guerre mondiale en Europe [5]) déterminant.

Nous proposons une analyse restreinte du phénomène classique de Gibbs qui est la cause des sauts en début et fin des créneaux de signaux électriques rapides observés sur un oscilloscope lent (les termes "rapides" et "lents" prendront leur sens plus tard) au moyen de l'analyseur harmonique. S'agissant d'une mise en œuvre d'un ordinateur mécanique, il serait souhaitable d'aboutir à une réalisation concrète : nous compenserons notre incapacité à réaliser quoi que ce soit de fonctionnel en mécanique par une simulation dynamique sous FreeCAD, pour conclure par une tentative de réalisation en prototypage rapide qui montrera les difficultés du passage de la simulation à la pratique.

## 2 L'analyseur harmonique

L'analyseur harmonique est un instrument mécanique conçu pour sommer des termes trigonométriques pondérés par la position d'un levier sur une tige mise en mouvement par une roue dentée. Une vidéo de son fonctionnement est disponible à [6] et l'ouvrage accompagnant la vidéo donne quelques informations techniques en s'étendant principalement sur les aspects esthétiques de la réalisation sans entrer dans les détails du fonctionnement.

Pour résumer les grands principes que nous tirons de l'analyse de ces vidéos :

- un axe fixé en bord de roue parcourt, lors de la rotation de la roue, les coordonnées  $(\cos(\omega t), \sin(\omega t))$ . Ainsi, un arbre suivant l'ordonnée de cet axe présente une position  $\sin(\omega t)$  soit une fonction trigonométrique
- deux roues adjacentes de rayon  $R_1$  et  $R_2$  tournent aux vitesses angulaires  $\omega_1$  et  $\omega_2$  telles que  $R_1\omega_1 = R_2\omega_2$  (condition de non-glissement à l'interface) mais *en sens opposé*
- un ressort étiré de  $\Delta z$  de sa position au repos induit une force  $F = k\Delta z$  avec  $k$  la raideur du ressort (loi de Hooke)
- à l'équilibre, la somme vectorielle des forces agissant sur un solide à l'équilibre s'annulent, soit  $\sum F = 0$  si toutes les forces sont orientées dans la même direction.

Fort de ces rappels de mécanique du solide, la bijection entre le signal temporel  $s(t)$  supposé de période  $T$  et sa décomposition en série de Fourier (Fig. 1) affirme qu'il existe une série de coefficients  $c_n$  tel que

$$s(t) = \sum c_n \exp(j2\pi(n/T) \times t)$$

et ces coefficients s'identifient par

$$c_n = \frac{1}{T} \int_0^T s(t) \exp(-j2\pi(n/T) \times t) dt$$

Dans le cas particulier du créneau périodique égal à -1 de  $-T/2$  à 0 puis égal à 1 de 0 à  $+T/2$ ,  $c_n$  s'identifie analytiquement en considérant que le créneau étant impair, les termes en cosinus (fonction paire) sont nécessairement tous nuls et seuls les termes en sinus (fonction impaire) subsistent. Ainsi,  $c_n = \frac{1}{T} \int_{-T/2}^{T/2} \sin(2\pi n/T \times t) dt$  et comme la primitive de  $\sin$  est  $-\cos$ , cette intégrale est égale à  $c_n = \frac{-2}{T} \left[ \frac{T}{2\pi n} \cos(t) \right]_0^{T/2} = -\frac{2}{\pi n} [\cos(n\pi) - \cos(0)]$ . Puisque  $\cos(n\pi) = 1$  si  $n$  est pair et  $\cos(n\pi) = -1$  si  $n$  est impair, nous déduisons que  $c_n = 1 - 1 = 0$  si  $n$  est pair et  $c_n \propto 1/n$  si  $n$  est impair. Nous retrouvons le développement classique de la série de Fourier du créneau comme somme des termes impairs pondérés par l'inverse du numéro de l'ordre du coefficient.

75. Before leaving for a time the subject of the composition of harmonic motions, we must, as promised in § 62, devote some pages to the consideration of Fourier's Theorem, which is not only one of the most beautiful results of modern analysis, but may be said to furnish an indispensable instrument in the treatment of nearly every recondite question in modern physics. To mention only sonorous vibrations, the propagation of electric signals along a telegraph wire, and the conduction of heat by the earth's crust, as subjects in their generality intractable without it, is to give but a feeble idea of its importance. The following seems to be the most intelligible form in which it can be presented to the general reader:—

THEOREM.—A complex harmonic function, with a constant term added, is the proper expression, in mathematical language, for any arbitrary periodic function; and consequently can express any function whatever between definite values of the variable.

FIGURE 1: Extrait de l'ouvrage de Lord Kelvin [7] sur les séries de Fourier

## 3 Modélisation sous FreeCAD

Nous commençons par dessiner la structure sous FreeCAD d'une série de roues dentées tangentes (afin de s'entraîner mutuellement), au bord de chacune desquelles nous plaçons un axe pour entraîner une pièce en forme de T inversé, barre verticale guidée qui ne peut se déplacer que verticalement (glissière), entraînée par l'axe pouvant glisser le long de la gorge usinée dans la partie horizontale de la

pièce. De ce fait, le déplacement vertical du T est le rayon de la roue multiplié par le sinus de l'angle de la roue au cours de sa révolution (Fig. 2) – ce mécanisme est nommé “manivelle-cadre” d’après Wikipedia qui en cite l’utilisation dans [https://en.wikipedia.org/wiki/Tide-predicting\\_machine](https://en.wikipedia.org/wiki/Tide-predicting_machine). Dans un premier temps, les roues dentées sont modélisées par des cylindres lisses supposés s’entraîner par frottement : nous étendrons le modèle aux roues munies de pignons plus tard (section 8). Nous supposons que chaque roue est libre de tourner selon l’axe en son centre, que l’axe en bord de roue peut glisser dans la fente horizontale, et que la barre verticale est guidée dans un cylindre (non représenté) à l’aplomb du centre de la roue pour guider selon un mouvement de glissière et interdire tout mouvement horizontal de cette barre (Fig. 2, gauche). La partie brune en sommet de tige, de longueur qui sera variable, représente un ressort qui imposera donc une force proportionnelle à son élongation.

Par condition de contact des roues de rayon  $R_i$  les unes avec les autres, leur vitesse angulaire  $\omega_i$  respecte  $R_1\omega_1 = R_2\omega_2 = R_3\omega_3$  de façon à ce que  $\omega_i = \frac{R_1}{R_i}\omega_1$  : en choisissant judicieusement le rayon de la  $i$ ème roue comme le rayon de la première roue divisé par  $(2i + 1)$ , nous avons bien des vitesses de rotation (donc des pulsations du mouvement vertical des pièces en forme de T inversé) qui croissent comme la séquence des harmoniques impairs  $(2i + 1)$ .

Tous les ressorts au sommet des pièces en T inversé sont reliées à une pièce commune guidée dans un mouvement vertical uniquement qui tient une mine en son extrémité : la position de cette mine peut tracer la séquence temporelle formée par la série de Fourier. Cette barre est par ailleurs retenue par un dernier ressort au bâti fixe de la machine pour lui éviter d’aller se coller contre le mécanisme des roues.

Les divers ressorts en parallèle agissent tous pour amener la mine d’ordonnée  $z_m$  vers le mécanisme des roues selon une somme des forces qui doit balancer, à l’équilibre, la force de rappel du ressort lié au bâti d’ordonnée  $z_b$ . Si chaque roue  $R_i$  place sa pièce en forme de T inversé en ordonnée  $z_i$ , alors la somme des forces est  $\sum k(z_i - z_m)$  qui soit contrebalancer la force de rappel  $k(z_m - z_b)$  donc  $\sum(z_i - z_m) = z_m - z_b \Leftrightarrow (N + 1)z_m = z_b + \sum z_i$  en considérant  $N$  harmoniques. Par ailleurs, les roues ayant été sélectionnées de rayon évoluant comme  $1/(2i + 1)$ , l’élongation du  $i$ ème ressort décroît comme  $i$ , vérifiant la pondération des coefficients de la décomposition en série de Fourier du créneau.

55. Those common kinds of mechanism, for producing rectilinear from circular motion, or *vice versa*, in which a crank moving in a circle works in a straight slot belonging to a body which can only move in a straight line, fulfil strictly the definition of a simple harmonic motion in the part of which the motion is rectilinear, if the motion of the rotating part is uniform. The motion of the treadle in a spinning-wheel approximates to the same condition when the wheel moves uniformly; the approximation being the closer, the smaller is the angular motion of the treadle and of the connecting string. It is also approximated to more or less closely in the motion of the piston of a steam-engine connected, by any of the several methods in use, with the crank, provided always the rotatory motion of the crank be uniform.

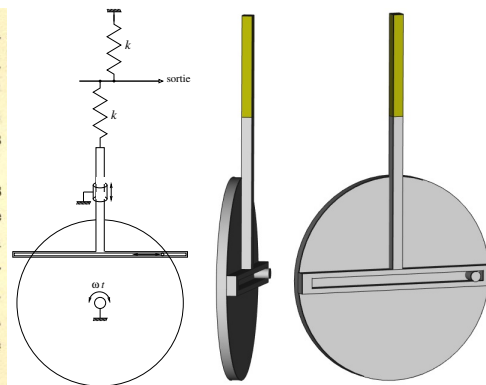



FIGURE 2 – Roue munie d’un axe en son bord guidant verticalement une pièce en forme de T inversé dont la fente horizontale permet le glissement de l’axe, tel que décrite dans [8] (gauche). Milieu-gauche : la barre verticale est supposée guidée dans un mouvement de glissière verticale interdisant tout mouvement horizontal. Droite : implémentation sous FreeCAD.

## 4 Scripter l'évolution dynamique du système en Python

Ayant observé la vidéo de [9] et les scripts fournis en pièces jointes à la vidéo, nous nous sommes inspirés de <http://linuxforanengineer.blogspot.com/2015/08/freecad-double-slider-mechanism.html> pour animer le modèle d'analyseur harmonique. Le principe de base consiste à associer un timer se déclenchant périodiquement pour appeler une fonction mettant à jour les paramètres géométriques du modèle mécanique. Puisque FreeCAD permet de scripter en Python la position et la géométrie de chaque pièce, il reste à l'auteur d'implémenter les lois de composition des divers degrés de liberté pour évaluer l'évolution des pièces mobiles du système mécanique.

Plutôt que nous lancer dans le système complexe de la roue-axe-T, considérons dans un premier temps le système simple de deux masses reliées par un ressort et guidées sur une glissière. Le premier ressort entre le bord de la glissière supposée fixe (d'ordonnée 0) et la première masse (d'ordonnée  $z_1$ ) est de même raideur et de même longueur que le second ressort qui relie cette première masse à la seconde masse dont la position  $z_2$  est supposée imposée par l'utilisateur. Ainsi, la force  $-k(z_1)$  entre le point fixe et la première masse est égale à la force  $+k(z_2 - z_1)$  entre la seconde masse et la première, et en régime statique où la somme des forces est nulle il reste  $-kz_1 + k(z_2 - z_1) = 0 \Leftrightarrow z_1 = z_2/2$  (on aurait pu s'en douter intuitivement : il s'agit de l'équivalent mécanique du pont de résistances diviseur de tension). Le programme Python de modélisation du comportement de ce système va donc lire la position de la seconde masse mobile (cm) et en déduire la position de la première masse libre c1 afin de la placer à la bonne ordonnée. Accessoirement, la masse mobile se voit imposer un mouvement périodique lent (pour vérifier l'hypothèse de conditions quasi-statiques de somme des forces nulles).



```
1 import FreeCAD
2 from PySide import QtCore
3 from math import sin, cos, radians
4
5 # retrieve the objects from the document:
6 # start typing FreeCAD.ActiveDocument and see completion
7 cm=FreeCAD.ActiveDocument.getObject("Box")
8 cf=FreeCAD.ActiveDocument.getObject("Box001")
9 slider = FreeCAD.ActiveDocument.getObject("Cylinder")
10 cf_ini=cf.Placement.Base;
11 i=0
12
13 def update():
14     global i
15     alpha = radians( i )
16     z = slider.Height/20*(cos( alpha )+1.5)
17     cf.Placement=FreeCAD.Placement(cf_ini+FreeCAD.Vector(0,0,z),cf.Placement.Rotation)
18     cm.Placement.Base.z=cf.Placement.Base.z/2
19     FreeCAD.Gui.updateGui()
20     i += 1
21
22 timer = QtCore.QTimer()           # create a timer object
23 timer.timeout.connect( update )  # connect timer event to function
24 timer.start( 10 )                # trigger every 10 ms
```

Ces opérations s'automatisent dans la console Python de FreeCAD selon le programme ci-dessus. Nous commençons par charger les bibliothèques permettant d'accéder aux objets de FreeCAD, avant d'instancier les divers objets présents dans la scène que nous avons pris soin de charger. Notre approche pour connaître les noms des objets est, dans la console Python, de commencer à taper

FreeCAD.ActiveDocument.

et d'observer les complétions proposées par l'éditeur. Nous constatons que les parallélépipèdes s'appellent Box mais les cylindres restent bien Cylinder.

Notons qu'au lieu de directement inclure les commandes Python dans la console de FreeCAD, il est plus efficace d'utiliser son éditeur favori et appeler le script par

```
import os
os.chdir('/tmp') # aller dans le repertoire contenant le script
import mon_programme
# import importlib
# importlib.reload(mon_programme)
```

Cette séquence charge le programme Python `mon_programme.py` situé dans `/tmp` et éventuellement le recharge (`importlib.reload`) lors de la phase de développement.

## 5 Généralisation à l'analyseur harmonique

Cette rapide analyse se généralise à l'analyseur harmonique par les montages présentés en Fig. 3. Dans ce montage, 3 (gauche) ou 4 (droite) composantes spectrales sont analysées. Les ratios des roues sont de 15 : 5 : 3, c'est à dire que lorsque la première roue tourne à  $\omega$  rad/s, la deuxième roue tourne à  $3\omega$  rad/s et la troisième roue à  $5\omega$  rad/s. Les roues intermédiaires ont pour vocation de changer le signe de la rotation sans changer la vitesse, étant de même diamètre que leur voisine. Finalement l'axe de chaque roue est situé en bord de rayon, de façon à ce que la force exercée par chaque ressort décroît comme le rayon, c'est à dire  $1/(2N + 1)$  pour la  $N$ ème roue dont le rayon a été ajusté pour que sa vitesse de rotation soit  $(2N + 1)\omega$ ,  $\omega$  étant la vitesse de rotation de la première roue qui entraîne les autres.

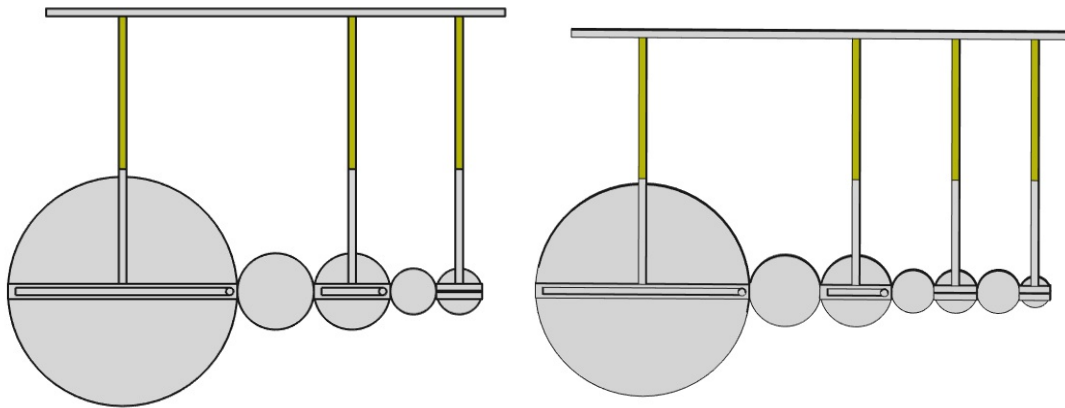


FIGURE 3 – Analyseur harmonique pour 3 composantes spectrales (gauche) et 4 composantes spectrales (droite).

Un extrait du script Python pour contrôler cette séquence est

```
1 p1 = FreeCAD.ActiveDocument.getObject("Connect")
2 h1 = FreeCAD.ActiveDocument.getObject("Connect001")
3 c0 = FreeCAD.ActiveDocument.getObject("Cylinder")
4 c1 = FreeCAD.ActiveDocument.getObject("Cylinder001")
```

```

5 spring1 = FreeCAD.ActiveDocument.getObject( "Box003" )
6 plini=pl1.Placement
7 h1ini=h1.Placement
8 springini=spring1.Placement
9 leninit1=spring1.Width
10 [... idem pour les autres roues ...]
11
12 pen=FreeCAD.ActiveDocument.getObject( "Box012" )
13 peninit=pen.Placement.Base
14 i=0
15 def update():
16     global i
17     alpha = radians( i )
18     y1 = c1.Placement.Base.x*(sin( alpha ))
19     p1.Placement=FreeCAD.Placement(plini.Base,FreeCAD.Rotation(FreeCAD.Vector(0,0,1),i) )
20     h1.Placement=FreeCAD.Placement(h1ini.Base+FreeCAD.Vector(0,y1,0) , h1ini.Rotation )
21     spring1.Placement = FreeCAD.Placement( springini.Base+FreeCAD.Vector(0,y1,0),h1ini.Rotation)
22
23 [... idem pour les autres roues : alpha = (2N+1)*i pour roue N>=0 ...]
24
25     yvar=(y1+y2+y3+y4)/3 # la force de rappel est un terme constant choisi a 0
26     spring1.Width=leninit1.Value-y1+yvar
27     spring2.Width=leninit2.Value-y2+yvar
28     spring3.Width=leninit3.Value-y3+yvar
29     spring4.Width=leninit4.Value-y4+yvar
30     pen.Placement.Base.y=peninit.y+yvar

```

Ce script Python fait tourner la roue (`p1.Placement` dont l'argument `FreeCAD.Rotation` prend en argument une rotation autour de l'axe de normale  $z$ ) et déplace la hauteur de la barre horizontale du T pour suivre le mouvement de la roue (`h1.Placement` dont l'argument `h1ini.Base` est modifié de `FreeCAD.Vector( 0,y1,0 )` avec  $y1$  l'ordonnée de l'axe sur la bord de chaque roue) pour finalement calculer la force appliquée à la barre horizontale reliant les divers ressorts et en déduire la position comme  $yvar=(y1+y2+y3+y4)/2$ . Cette relation est un cas particulier de la démonstration précédente qui indiquait que la barre qui porte une mine afin de tracer la courbe résultante du calcul et reliée par un ressort de même raideur  $k$  que tous les  $M$  autres ressorts au bâti statique de la machine vérifie

$$\sum_{m=1}^M k(z_m - z_{pen}) = k(z_{bati} - z_{pen}) \Leftrightarrow z_{pen} = \frac{1}{M-1} \sum_{m=1}^M z_m$$

en choisissant arbitrairement  $z_{bati} = 0$ . Nous trichons quelque peu car nous avons résolu les équations et introduisons la solution en plaçant les éléments du modèle avec le bon angle ou la bonne position : il aurait été plus satisfaisant d'imposer des contraintes dans le modèle tel que décrit à [https://www.freecadweb.org/wiki/Sketcher\\_Workbench](https://www.freecadweb.org/wiki/Sketcher_Workbench) et de ne faire évoluer que l'angle de la première roue pour voir, par conséquences des contraintes entre les pièces (entraînement, glissière, ...), les autres pièces se mettre en mouvement. Cette méthode nécessite de partir du plan 2D (*sketch*) et d'extruder les pièces contraintes entre elles au lieu, comme nous l'avons fait, d'assembler des éléments 3D.

Observer le mouvement du mécanisme décrit en Fig. 4 est hypnotisant – voir l'animation à [https://github.com/jmfriedt/harmonic\\_analyzer\\_FreeCAD/blob/master/video/FreeCAD2019-12-25\\_4harmonics.gif](https://github.com/jmfriedt/harmonic_analyzer_FreeCAD/blob/master/video/FreeCAD2019-12-25_4harmonics.gif) – car, malgré la trivialité de la démonstration, l'observation de la combinaison des forces introduites par les roues en mouvement se conclut par un mouvement selon une forme de créneau du stylo fixé en bout de barre fixe.

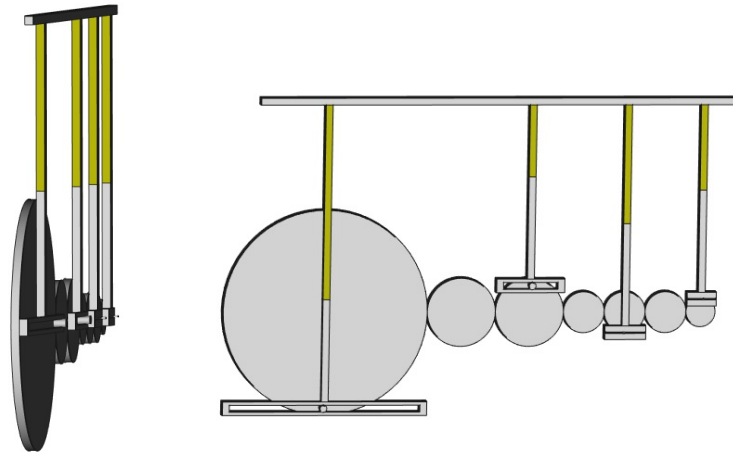


FIGURE 4 – Capture d'écran du mécanisme en mouvement.

## 6 Résultats

L'observation qualitative d'un mouvement s'apparentant à un mouvement en créneau est peu satisfaisante pour une analyse quantitative. Nous sauvons donc dans un fichier les positions successives de  $y_{var}$ . Nous constatons (Fig. 5) le parfait accord avec le modèle de somme des harmoniques impaires pondérés par l'inverse du numéro de l'harmonique. Nous avons aussi indiqué (orange) la conséquence d'oublier les roues intermédiaires : dans ce cas, l'harmonique 3 tourne dans la direction opposée des harmoniques 1 et 5 et la somme des forces résultantes ne ressemble pas à un créneau.

## 7 Automatisation de la synthèse du modèle

Atteindre l'objectif de l'analyseur harmonique de Michelson avec ses vingt composantes spectrales est passablement fastidieux si chaque pièce est réalisée manuellement. Nous pouvons automatiser la création des pièces dans le script au lieu d'instancier des pièces dessinées à la main. FreeCAD a le bon goût de fournir dans la console Python les commandes correspondant à chaque opération manuelle, à la subtilité près que les positions sont sans unité alors que les longueurs sont données en mm, deux représentations incompatibles lors des opérations arithmétiques : les unités sont éliminées en considérant la propriété `.Value` d'une longueur.

À titre d'exemple de l'assemblage de la roue principale et de son axe – variable nommée `pignon` – le code ci-dessous illustre l'instanciation de deux cylindres, la définition de leurs paramètres (hauteur, rayon et position) puis de leur fusion.

```

1 for k in range(nbharmonics):
2     tmproue=FreeCAD.ActiveDocument.addObject("Part::Cylinder", "Cylinder"+str(k))
3     tmproue.Height=1
4     tmproue.Radius=100/(2*k+1)
5     tmproue.Placement.Base.z=-1
6     pignon=FreeCAD.ActiveDocument.addObject("Part::Cylinder", "Pignon"+str(k))
7     pignon.Height=1
8     pignon.Radius=0.5
9     pignon.Placement.Base.x=tmproue.Radius.Value # -1

```



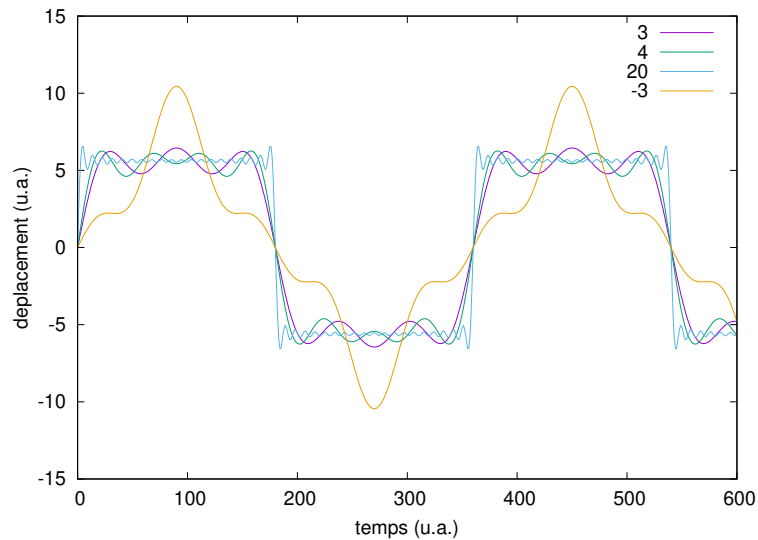


FIGURE 5 – Évolution de la position du stylo fixé à la barre sommant les forces induites par les ressorts fixés par leur autre extrémité aux roues en rotation, en fonction du temps. Le motif de créneau se dessine d'autant plus clairement (violet puis vert) que le nombre d'harmoniques sommées augmente. Se tromper de sens de rotation d'une roue (jaune) induit une réponse qui ne correspond pas au motif de créneaux attendu. Nous avons automatisé la réalisation du système à 20 harmoniques (bleu) pour nous approcher du créneau visé.

```

10 tmprouepignon=FreeCAD.ActiveDocument.addObject("Part::MultiFuse","Fusion")
11 tmprouepignon.Shapes = [tmproue,pignon]
12 rouepignon.append(tmprouepignon)
13 [...]
14
15 k=0
16 position=0
17 for r in roue:
18     if k==0:
19         position=r.Radius
20     else:
21         rouepignon[k].Placement.Base.x=position+r.Radius+2*r.Radius
22         roue2[k].Placement.Base.x=position+r.Radius
23         position=position+2*r.Radius+2*r.Radius
24     k+=1

```

L'assemblage du dispositif complet avec ses 20 composantes spectrales est illustré en Fig. 6. Évidemment avec un tel nombre de pièces et en particulier le redimensionnement dynamique de la longueur des ressorts, le modèle devient plus lent à simuler mais reste relativement fluide.

## 8 Modèle réalisable : remplacer les cylindres par des roues dentées

La découverte du greffon (*workbench*) FCGears pour la conception de roues dentées permet d'envisager de passer d'un modèle virtuel à un modèle réel. En effet, en remplaçant le rayon des cylindres



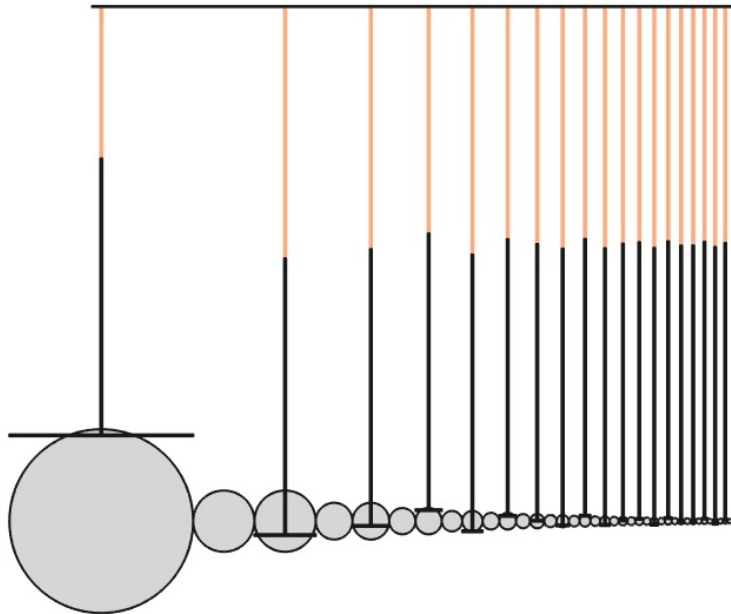


FIGURE 6 – Modèle synthétisé automatiquement pour sommer les contributions de 20 harmoniques. Les ressorts, pièces de longueur variable, sont indiqué en orange.

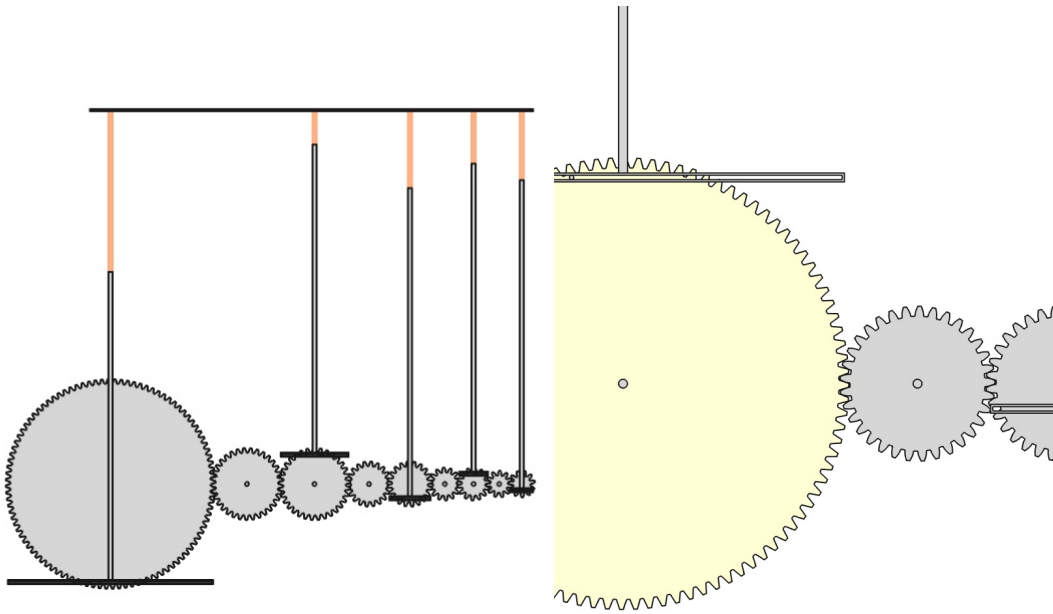
par le nombre de dents dans la roue, nous obtenons un modèle réaliste d'entraînement, qui nécessite uniquement de transformer le rayon de valeur positive quelconque en nombre entier de dents, dont nous verrons les conséquences dans la prochaine section. Les attributs définissables de la roue dentée sont son épaisseur `height` et le nombre de dents `teeth` :

```

1 import freecad.gears.commands
2 tmproue=freecad.gears.commands.CreateInvoluteGear.create()
3 tmproue.height=1
4 tmproue.teeth=math.floor(R/(2*k+1))
5 tmproue.Placement.Base.z=-1
6 [...]
7 if (k>0):
8     roueinter=freecad.gears.commands.CreateInvoluteGear.create()
9     roueinter.height=1
10    roueinter.teeth=math.floor(R/(2*k+1))
11    roueinter.Placement.Base.z=-1
12 [...]

```

En passant l'attribut "simple" de la roue dentée de "false" à "true", nous constatons qu'une roue dentée de  $M$  dents est équivalente à un cylindre de *diamètre*  $M$  mm. Par rapport à notre modèle précédent où les distances étaient paramétrées sur les rayons des cylindres, nous devons diviser toutes les dimensions par 2.



## 9 Sources d'incertitudes

Nous répétons à qui veut l'entendre la supériorité de l'électronique numérique sur les solutions analogiques, voire mécanique. Que nous apprend ce modèle mécanique sur le sujet?

- remplacer les cylindres lisses par des roues dentées élimine le problème de glissement des surfaces les unes sur les autres, problème classiquement décrit dans les intégrateurs dans lesquels une roue glisse sur un disque en rotation [11]. Ce problème met d'ailleurs en évidence pratique le problème d'adaptation d'impédance couramment mis en œuvre en électronique analogique par le montage suiveur : comment garantir l'absence de couple sur le système entraîné (la roue intégratrice) pour éviter le glissement compte tenu de la faible force de frottement avec le disque? ce problème est illustré dans [1] avec l'exemple de l'entraînement d'une tourelle de canon par un système de visée manipulé par un opérateur [11],
- l'axe en bord de roue entraînant la pièce en T inversé chargée de convertir la rotation de la roue en mouvement de translation en sinus de l'angle ne peut se trouver sur la circonférence, induisant une pondération du mouvement qui n'est pas exactement en l'inverse de la pulsation,
- de façon générale, la pondération de la  $k$ ème harmonique peut être autre que  $1/k$  tel que illustré ici en plaçant l'axe d'entraînement de la pièce en T inversé ailleurs que sur la circonférence, avec l'incertitude d'usinage de cette position,
- la condition d'engrenage peut imposer une condition de phase entre roues adjacentes qui n'est pas exactement nulle comme le voudrait la théorie.

Tous ces défauts peuvent bien entendu être corrigés par une conception mécanique adéquate, mais visent à mettre en évidence les risques d'une conception analogique face à une solution numérique, *a fortiori* implémentée sur des entiers et ne souffrant donc pas des erreurs de calculs sur les nombres à virgule flottante. En figure 7, nous sommes passés d'un modèle avec un nombre arbitraire de dents (100) sur la roue la plus grande vers un nombre multiple des trois premières harmoniques ( $105 = 3 \times 5 \times 7$ ) et des cinq premières harmoniques ( $945 = 3 \times 5 \times 7 \times 9$ ) pour finalement revenir au cas idéal. Comme les calculs des coefficients de Fourier sont des intégrales, ces erreurs s'accumulent pour donner un résultat dont l'erreur croît avec le temps.

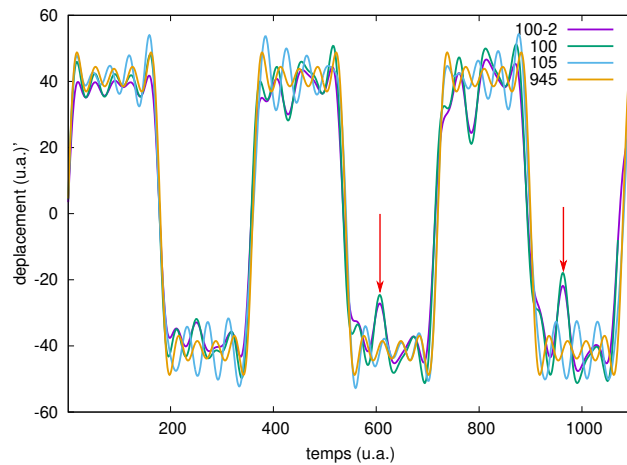


FIGURE 7 – Erreurs de calcul (flèches rouges) suite aux incertitudes sur le modèle mécanique : 100 dents et un axe d'entraînement à 2 mm du bord de la roue dentée; 100 dents et l'axe en bord de roue; 105 dents; 945 dents pour revenir au cas idéal.

## 10 Cas du triangle

Nous avons commencé cette étude par une erreur en plaçant les engrenages adjacents, oubliant de ce fait que les roues tournent en sens inverse et induisent donc un signe négatif dans l'argument de la fonction trigonométrie.

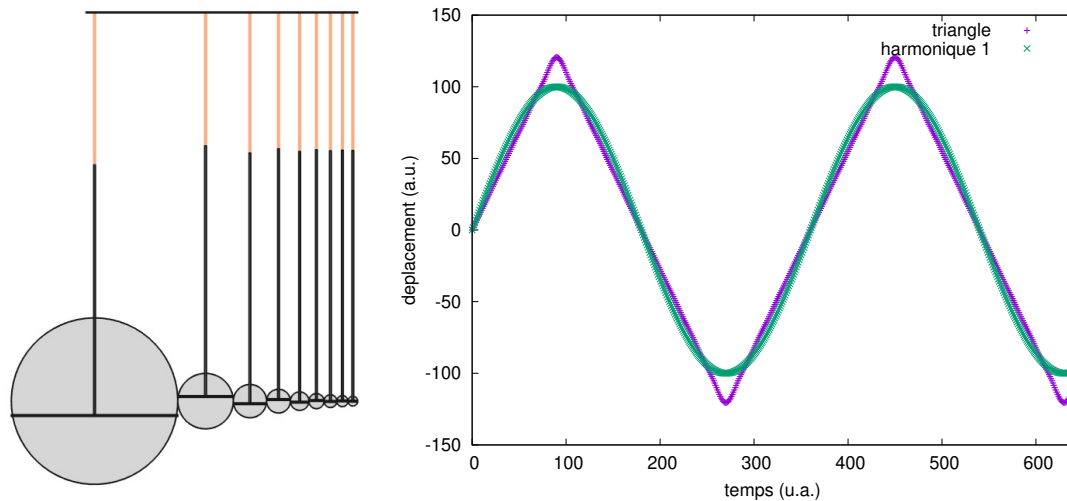


FIGURE 8 – Gauche : générateur de triangle, dans lequel l'utilisation des roues adjacentes garantit l'inversion du signe de l'argument des fonctions trigonométriques successives. Droite : tracé de la fonction résultant de la somme des contributions des termes trigonométriques qui représente bien un triangle (violet), et comparaison avec le mode principal tracé par la roue de plus grand rayon (vert).

Nous retournons cette erreur à notre avantage en considérant les coefficients de la série de Fourier du triangle. La fonction triangle est une fonction linéaire croissante sur la première moitié de sa période et linéaire décroissante sur sa seconde moitié. Ainsi, les coefficients de la série de Fourier impliquent le calcul de la primitive de  $x \sin(x)$  dont on se rappellera, lors de l'intégration par parties qui nous indique que  $\int u'v = [uv] - \int v'u$  et en prenant  $u = x$  et  $v = -\cos$  (donc  $u' = 1$  et  $v' = \sin$ ), que ces intégrales sont de la forme  $[\sin(x) - x \cos(x)]$  pour retrouver la solution classique que le coefficient de la  $n$ ème harmonique,  $n$  impair, est  $(-1)^{(n-1)/2}/n^2$ . Le terme  $-1^{(n-1)/2}$  représente le changement de direction des roues adjacentes, tandis que la pondération en  $1/n^2$  est respectée en plaçant l'axe dirigeant la pièce en T inversé non sur la circonférence de la roue mais à  $1/n$  de son rayon (qui lui même décroît comme  $1/n$  du rayon de la première roue). Une telle conception est illustrée en Fig. 8.

```

1  tmproue.Radius=100/(2*k+1) # composante harmonique impaire (pulsation angulaire)
2  [...]
3  pignon.Placement.Base.x=tmproue.Radius.Value/(2*k+1) # ponderation = 1/n*1/n=1/n^2
4  tmprouepignon.Shapes = [tmproue,pignon]

```

## 11 Réalisation

Les roues dentées sont réalisées en découpe laser sur PMMA (Plexiglas®) de 3 mm d'épaisseur (Fig. 9).

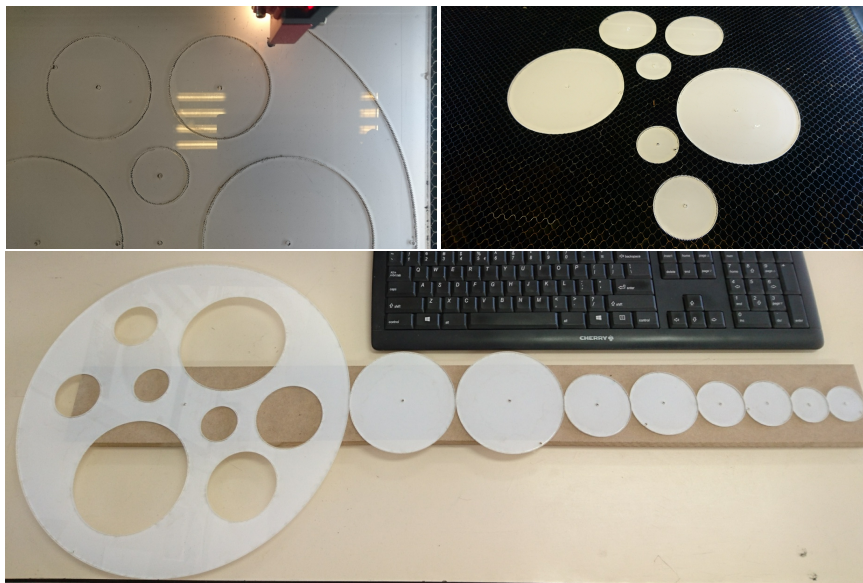


FIGURE 9 – En haut à gauche : découpe laser des pièces dans du PMMA de 3 mm d'épaisseur. La plus grande roue, qui n'a aucune raison d'être pleine, a fourni la matière pour les roues plus petites. En haut à droite : malgré les structures fines des roues dentées, les pièces se séparent sans problème de la plaque de matière. Bas : assemblage des pièces.

La contrainte sur la résolution des crans impose la dimension des roues. Partant d'un cran d'au moins 1 mm de longueur/largeur, la roue de 945 crans doit présenter une circonférence de 945 mm soit un rayon de 150 mm. Ce diamètre de 300 mm va donc imposer l'extension du ressort tiré par la

barre en T inversé fixée en périphérie. Or un ressort ne peut être étiré excessivement sous peine de perdre sa propriété d'élasticité (relation linéaire entre force et extension au travers de la constante de raideur du ressort) mais atteindre un régime de déformation plastique irréversible. Ces contraintes déterminent les dimensions de l'instrument assemblé.

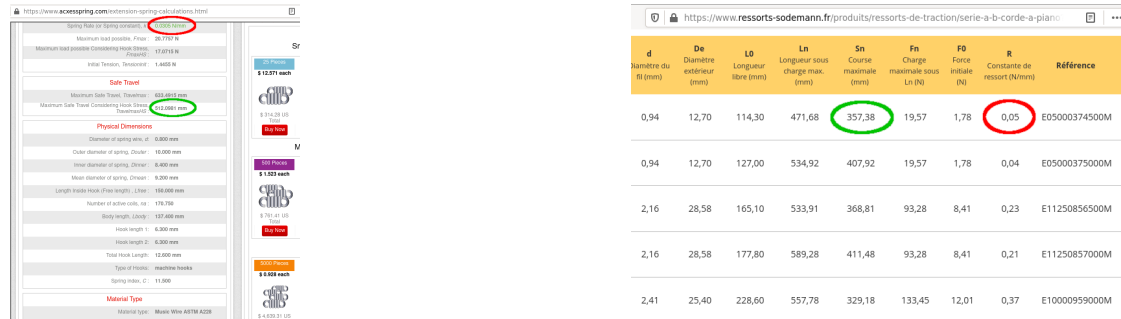


FIGURE 10 – Deux fournisseurs de ressorts avec des caractéristiques à peu près identiques : que ce soit corde à piano ou inox, les raideurs minimales sont de l'ordre de 0,03 à 0,05 N/mm (contrainte rouge) pour une extension d'au moins 30 cm (contrainte verte).

Par ailleurs, le couple raideur du ressort  $k$  et extension  $\delta x$  vont déterminer la force  $F = k \times \delta x$  appliquée sur la barre en T inversé ou les axes des roues et donc dimensionner ces éléments. Nous constatons en consultant les divers catalogues de fournisseurs de ressorts qu'il y a une relation entre le diamètre extérieur du ressort et le diamètre du fil : plus un ressort est de diamètre important, plus sa raideur est faible, mais aussi plus le diamètre du fil composant le ressort augmente, augmentant donc la raideur. Nous n'avons pu trouver de ressort de raideur inférieure à 0,05 N/mm, soit une force en pleine extension (30 cm) de 15 N ou l'équivalent exercé par une masse de 1,5 kg dans le champ de gravité terrestre (Fig. 10). Les pièces en T inversé aussi réalisées en PMMA de 3 mm d'épaisseur sont incapables de résister à une telle force, et une conception plus robuste que la découpe laser de plastique est nécessaire pour ces pièces.

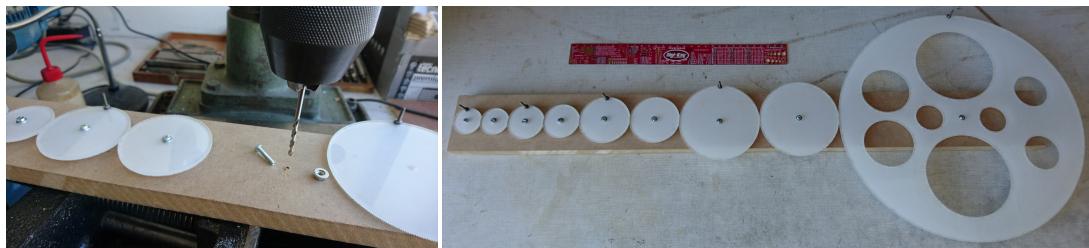


FIGURE 11 – Gauche : positionnement des pièces tangentes pour engrener les pignons. Les roues sont surélevées d'une hauteur d'écrou et de rondelle pour éviter que les guides des barres en T ne butent sur le support. Droite : montage achevé. La règle de 30 cm donne l'échelle.

L'assemblage de l'instrument démontre les limites du prototypage rapide : le jeu excessif et la petite dimension des crans imposent une précision d'assemblage bien meilleure que le montage grossier abordé (Fig. 11). Néanmoins, avec bien des frottements et des passages grippés lorsque les vis guidant les barres en T doivent passer à côté des axes de guidage de la roue dentée voisine, l'assem-

blage veut bien tourner en entraînant à bon rythme les diverses composantes de la série de Fourier. Cette expérience induit à d'autant mieux respecter les montages incroyablement complexes des derniers ordinateurs mécaniques [4], avant le passage aux ordinateurs électro-mécaniques pour aboutir aux systèmes intégrés que nous connaissons actuellement.

## 12 Conclusion

L'intérêt de l'ordinateur mécanique, peu précis, encombrant et lent, peut paraître discutabile à une époque de concentration quasi-infinie de portes logiques dans un circuit intégré. Au-delà de son intérêt historique, l'ordinateur mécanique reste d'actualité dans les environnements hostiles dans lesquels la technologie de dopage de silicium ne saurait survivre [12], voir pour des traitements de signaux radiofréquences dont les vitesses ne sont pas (encore) accessibles aux traitements numériques en temps discret [13]. Ces principes s'abordent sans mettre en marche une fraiseuse en prototypant dans le monde virtuel de FreeCAD et ses scripts Python pour manipuler les positions et dimensions des pièces du modèle mécanique.

Le système de Michelson est passablement plus complexe par l'ajout de leviers intermédiaires qui permettent de dynamiquement pondérer la contribution de chaque roue (coefficient de la série de Fourier) sans s'imposer une contribution égale au rayon de la roue. Nous laissons le soin au lecteur de développer le modèle proposé ici pour atteindre la flexibilité de l'instrument de Michelson, voire de prédire par ce biais la prochaine marée.

Tous les scripts et vidéos des systèmes mis en œuvre dans ce document sont disponibles à [https://github.com/jmfriedt/harmonic\\_analyzer\\_FreeCAD](https://github.com/jmfriedt/harmonic_analyzer_FreeCAD)

## Références

- [1] D.A. Mindell, *Between Human and Machine – Feedback, Control and Computing Before Cybernetics* Johns Hopkins University Press (2004) puis D.A. Mindell, *Digital Apollo : Human and Machine in Spaceflight*, MIT Press (2008)
- [2] S. Bennett, *A History of Control Engineering, 1930–1955*, IET (1993)
- [3] E.J. Murray, *The theory of mathematical machines, revised edition*, Kong's Crown Press (1948)
- [4] C.B. Burke, *It Wasn't All Magic : The Early Struggle to Automate Cryptanalysis, 1930s–1960s*, Cryptologic History **6**, National Security Agency (2002), p.153
- [5] B. Parker, *The tide predictions for D-Day*, Physics Today **64** (9), 35–40 (2011) ou, plus complet, P. Kennedy, *Engineers of Victory : The Problem Solvers who Turned the Tide in the Second World War*, Penguin (2014)

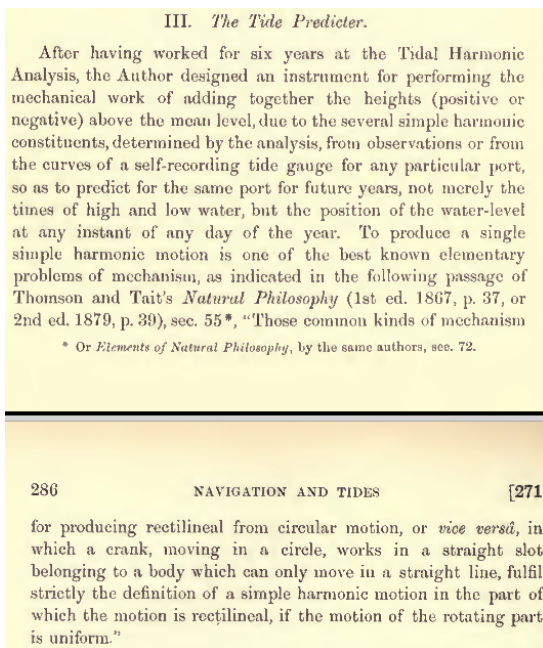


FIGURE 12: Description du calculateur analogique de marée comme somme des coefficients de Fourier de la série périodique identifiés empiriquement du niveau de la mer [7].

- [6] B. Hammack, *An Old Mechanical Computer : The Harmonic Analyzer* à <https://www.youtube.com/watch?v=GyYf1zRVu6M> (YouTube, accédé Dec. 2019)
- [7] W. Thomson Baron Kelvin, *Mathematical and physical papers, vol. 6* Cambridge University Press, 277- (1911) disponible à <https://ia802700.us.archive.org/1/items/mathematicalphys06kelvuoft/mathematicalphys06kelvuoft.pdf> avec un exemple de coefficients de Fourier en p.295
- [8] Lord Kelvin & P.G. Tait, *Treatise on Natural Philosophy, vol.1*, Cambridge University Press (1912), section 55, disponible à <https://archive.org/details/treatisnatphil01kelvrich/page/n61>
- [9] J.G. Egas Ortuño, *FreeCAD animation how to* à <https://www.youtube.com/watch?v=BHbn01BvpLI> (YouTube, accédé Dec. 2019)
- [10] B. Hammack, P. Ryan & N. Ziech, *Albert Michelson's Harmonic Analyzer : A Visual Tour of a Nineteenth Century Machine That Performs Fourier Analysis*, Articulate Noise Books (2014), disponible à <http://www.engineerguy.com/fourier/pdfs/albert-michelsons-harmonic-analyzer.pdf>
- [11] A.B. Clymer, *mechanical analog computers of Hannibal Ford and William Newell*, IEEE Annals of the History of Computing **15** (2) pp.19–34 (1993)
- [12] L'Automaton Rover for Extreme Environments (AREE) de la NASA, décrite à <https://www.nasa.gov/feature/automaton-rover-for-extreme-environments-aree/>, est un automate exclusivement mécanique conçu pour survivre aux températures extrêmes de Venus où les sondes électroniques n'ont fonctionné que quelques dizaines de minutes à la surface de la planète.
- [13] C. Ruppel & al., *A review of optimization algorithms for the design of the SAW transducers*, Proc. IEEE Ultrasonics Symposium pp.73–83 (1991) cite S. Fushimi & M. Morimoto, *A design method of FIR filters suitable for SAW filters*, Electronics and Communications in Japan (Part I : Communications) **71**(2), pp.61–70 (1988). Plus formel, C. Campbell, *Surface Acoustic Wave Devices and Their Signal Processing Applications* section 8 "Using the Remez Algorithm for SAW filter design", Academic Press (1989)