Fourier-Taylor Fitting Algorithm Theory

Garrett Lim, garrett.m.lim@gmail.com

Last Update: August 2, 2021

1 Introduction

This document describes the formulation of an algorithm to select parameters for a function that is fitted to a measured data set. The family of functions that are generated by this algorithm are a linear combination of polynomial and harmonic terms. The function generated by this algorithm may be considered as a hybrid of Taylor series and a Fourier series.

2 Parameters

Parameter	Description	Units
$c_h [q, h]$	Artificial output function harmonic coefficient of dimensions q and h	[output unit]
$c_p [q, p]$	Artificial output function polynomial coefficient of dimensions q and p	[output unit] / [input unit q]^p
$f (t [q]) \forall q \in [1, Q]$	Artificial output function	[output unit]
H	Maximum degree of harmonic fitting	Unitless
h	Artificial output function harmonic coefficient degree dimension index	Unitless
N	Number of measurements	Unitless
n	Index of measurement instance	Unitless
P	Maximum degree of polynomial fitting	Unitless
p	Artificial output function polynomial coefficient degree dimension index	Unitless
Q	Input parameter dimension count	Unitless
q	Input paramater dimensional index	Unitless
$t [q]$	Artificial input parameter of dimension q	[input unit q]
$x [q, n]$	Measured input parameter of dimension q at instance n	[input unit q]
$y [n]$	Measured output parameter at instance n	[output unit]
$\delta [q, h]$	Phase offset of harmonic fitting of dimensions q	[radians]
$\varepsilon [n]$	Error of fitted function at instance n	[output unit]
$\omega [q]$	Fundamental frequency of harmonic fitting of dimension q	[radians] / [output unit]

3 Theory

Assume we are given a set of synchronous measurements. We may arbitrarily select one dimension of these measurements as our output parameter set, $y [n]$. The remainder are the input parameter set, $x [q, n]$.

The goal of the algorithm is to determine the coefficients of the function that best match the measure dataset. The function has the form described in equation (1).

$$f\left(t\left[q\right]\right)=f_h\left(t\left[q\right]\right)+f_p\left(t\left[q\right]\right) \tag{1a}$$

$$f_h\left(t\left[q\right]\right)=\sum_{q=1}^{q=Q}\sum_{h=1}^{h=H}c_h\left[q,h\right]\cdot\cos\left(h\cdot\omega\left[q\right]\cdot t\left[q\right]+\delta\left[q,h\right]\right) \tag{1b}$$

$$f_p\left(t\left[q\right]\right)=c_p\left[1,0\right]+\sum_{q=1}^{q=Q}\sum_{p=1}^{p=P}c_p\left[q,n\right]\cdot\left(t\left[q\right]\right)^n \tag{1c}$$

The error between the fitted function and the measured data is defined as in equation (2).

$$\varepsilon\left[n\right]=\left(\sum_{q=1}^{q=Q}f\left(x\left[q,n\right]\right)\right)-y\left[n\right] \tag{2}$$

The algorithm minimizes the net square error.

$$0=\frac{\partial}{\partial c_h\left[q,h\right]}\left(\varepsilon^2\left[n\right]\right)=\varepsilon\left[n\right]\cdot\frac{\partial\varepsilon\left[n\right]}{\partial c_h\left[q,h\right]} \tag{3a}$$

$$0=\frac{\partial}{\partial\delta\left[q,h\right]}\left(\varepsilon^2\left[n\right]\right)=\varepsilon\left[n\right]\cdot\frac{\partial\varepsilon\left[n\right]}{\partial\delta\left[q,h\right]} \tag{3b}$$

$$0=\frac{\partial}{\partial\omega\left[q\right]}\left(\varepsilon^2\left[n\right]\right)=\varepsilon\left[n\right]\cdot\frac{\partial\varepsilon\left[n\right]}{\partial\omega\left[q\right]} \tag{3c}$$

$$0=\frac{\partial}{\partial c_p\left[q,p\right]}\left(\varepsilon^2\left[n\right]\right)=\varepsilon\left[n\right]\cdot\frac{\partial\varepsilon\left[n\right]}{\partial c_p\left[q,p\right]} \tag{3d}$$

Each of the partial derivatives in equation (3) are expanded in equation (13). Note that each of the partial derivates in equation (13) is two-dimensional or three-dimensional.

$$\frac{\partial\varepsilon\left[n\right]}{\partial c_h\left[q,h\right]}=\cos\left(h\cdot\omega\left[q\right]x\left[q,n\right]+\delta\left[q,h\right]\right) \tag{4a}$$

$$\frac{\partial\varepsilon\left[n\right]}{\partial\delta\left[q,h\right]}=-c_h\left[q,h\right]\cdot\sin\left(h\cdot\omega\left[q\right]\cdot x\left[q,n\right]+\delta\left[q,h\right]\right) \tag{4b}$$

$$\frac{\partial\varepsilon\left[n\right]}{\partial\omega\left[q\right]}=-c_h\left[q,h\right]\cdot h\cdot x\left[q,i\right]\cdot\sin\left(h\cdot\omega\left[q\right]\cdot x\left[q,i\right]+\delta\left[q,h\right]\right) \tag{4c}$$

$$\frac{\partial\varepsilon\left[n\right]}{\partial c_p\left[q,p\right]}=\begin{cases} p=0 & : \quad 1 \\ p\neq 0 & : \quad p\cdot\left(x\left[q,n\right]\right)^{(p-1)}\end{cases} \tag{4d}$$

The partial derivatives of the harmonic series are problematic because they incorporate coefficients. For this reason, it is necessary to (a) determine the fundamental frequencies, $\omega[q]$, in advance, and (b) dissociate the phase and amplitude coefficients. The algorithm assumes that these are the first q fundamental frequencies observed in y, as computed from a Fourier transformation.

The algorithm associates each dimension with its fundamental frequency by computing the minimum error of the simplified fitted function in equation (5).

$$f\left[n\right]=A\cdot\cos\left(\omega\cdot x\left[q,n\right]\right)+B\cdot\sin\left(\omega\cdot x\left[q,n\right]\right) \tag{5}$$

Multiple dimensions may share a fundamental frequency.

Note that computation of the dimension-associated fundamental frequencies is computationally expensive. If you, the reader, can find a way to improve this aspect of the algorithm, please contact the author.

The phase and amplitude coefficients can be separated via trigonometric identity.

$$C\cos\left(\theta+\delta\right)=C\cos\theta\cos\delta-C\sin\theta\sin\delta=A\cos\theta+B\sin\theta \tag{6}$$

$$C = \sqrt{A^2 + B^2} \tag{7}$$

$$\tan\delta = -\frac{B}{A} \tag{8}$$

The partial derivatives in equation (3) can be adjusted to those in equation (9). Remember that the fundamental frequencies, $\omega\left[q\right]$, are determined by a subprocess of the main algorithm.

$$\frac{\partial\varepsilon\left[n\right]}{\partial a_h\left[q,h\right]} = \cos\left(h\cdot\omega\left[q\right]x\left[q,n\right]\right) \tag{9a}$$

$$\frac{\partial\varepsilon\left[n\right]}{\partial b_h\left[q,h\right]} = \sin\left(h\cdot\omega\left[q\right]x\left[q,n\right]\right) \tag{9b}$$

$$\frac{\partial\varepsilon\left[n\right]}{\partial c_p\left[q,p\right]} = \begin{cases} p=0 & : & 1 \\ p\neq 0 & : & p\cdot\left(x\left[q,i\right]\right)^{(p-1)} \end{cases} \tag{9c}$$

The coefficients can be reorganized into a one-dimensional parameter, γ, that is constructed by the algorithm.

$$\gamma\left[s\right] = \left\{\begin{array}{c} c_p\left[q=1,p=0\right] \\ c_p\left[q=1,p=1\right] \\ \vdots \\ c_p\left[q,p=P[q]\right]:q=1 \\ c_p\left[q,p=1\right]:q=2 \\ \vdots \\ c_p\left[q,p=P\left[q\right]:q=2\right] \\ \vdots \\ c_p\left[q,p=P\left[q\right]\right]:q=Q \\ a_h\left[q=1,h=1\right] \\ b_h\left[q=1,h=1\right] \\ \vdots \\ a_h\left[q,h=H\left[q\right]\right]:q=1 \\ b_h\left[q,h=H\left[q\right]\right]:q=1 \\ \vdots \\ a_h\left[q,h=H\left[q\right]\right]:q=Q \\ b_h\left[q,h=H\left[q\right]\right]:q=Q \end{array}\right\} \tag{10}$$

Note that in equation (10) we allow that the maximum fitting degrees, $H[q]$ and $P[q]$, may vary by dimension, q. The number of coefficients must not exceed the number of measurement instances.[1]

The variables can also be reorganized into a conformal two-dimensional parameter, ξ, as in equation (11).

[1] i.e., $1+\sum_{q=1}^{q=Q}\left(1+2H\left[q\right]+P\left[q\right]\right)\leq I$.

$$
\xi [s,n] = \left\{ \begin{array}{c} 1 \\ x^1 [q,n] : q = 1 \\ \vdots \\ x^{P[q]} [q,n] : q = 1 \\ x^1 [q,n] : q = 2 \\ \vdots \\ x^{P[q]} [q,n] : q = 2 \\ \vdots \\ x^{P[q]} [q,n] : q = Q \\ \cos\theta [q,h,n] : h = 1, q = 1 \\ \sin\theta [q,h,n] : h = 1, q = 1 \\ \vdots \\ \cos\theta [q,h,n] : h = H[q], q = 1 \\ \sin\theta [q,h,n] : h = H[q], q = 1 \\ \vdots \\ \cos\theta [q,h,n] : h = H[q], q = Q \\ \sin\theta [q,h,n] : h = H[q], q = Q \end{array} \right\} : \theta [q,h,n] = h \cdot \omega [q] \cdot x [q,n] \tag{11}
$$

We can use these linearized coefficients and variables to compute the fitted functional values by a different manner.

$$
\sum_{q=1}^{q=Q} f(x[q,n]) = \gamma [s] \cdot \xi [s,n] \tag{12}
$$

The partial derivatives of the error can be conveniently redefined as well.

$$
\left(\frac{\partial \varepsilon [n]}{\partial \gamma [s]} : \gamma [s] = a_h [q,h] \right) = \cos (h \cdot \omega [q] \cdot x [q,n]) \tag{13a}
$$

$$
\left(\frac{\partial \varepsilon [n]}{\partial \gamma [s]} : \gamma [s] = b_h [q,h] \right) = \sin (h \cdot \omega [q] \cdot x [q,n]) \tag{13b}
$$

$$
\left(\frac{\partial \varepsilon [n]}{\partial \gamma [s]} : \gamma [s] = c_p [q,p] \right) = \left\{ \begin{array}{ccc} p = 0 & : & 1 \\ p \neq 0 & : & p \cdot x^{(p-1)} [q,n] \end{array} \right. \tag{13c}
$$

These can be used to compute the coefficients associated with minimum net error.

$$
y[n] \cdot \frac{\partial \varepsilon [n]}{\partial \gamma [s_2]} = (\gamma [s_1] \cdot \xi [s_1, n]) \cdot \frac{\partial \varepsilon [n]}{\partial \gamma [s_2]} \tag{14}
$$

At this point, the algorithm filters the data to exclude instances with incomplete data. Such a filtration will exclude instances, n, with null output measurements, $y[n]$, and null input measurements, $\xi [s,n]$.

$$
\left[\xi [s_1, n] \cdot \frac{\partial \varepsilon [n]}{\partial \gamma [s_2]} \right] \cdot \gamma [s_1] = \frac{\partial \varepsilon [n]}{\partial \gamma [s_2]} \cdot y[n] \tag{15}
$$

Note that $\left[\xi \cdot \frac{\partial \varepsilon}{\partial \gamma} \right]$ is a square matrix. The algorithm then uses Gaussian elimination to compute the coefficients, γ.

Finally, the amplitudes, c_h, and phase offsets, δ_h, of the harmonic series are computed from the isolated amplitudes, a_h and b_h.

$$
c_h [q,h] = \sqrt{a_h^2 [q,h] + b_h^2 [q,h]} \tag{16a}
$$

$$
\tan (\delta [q,h]) = \frac{b_h [q,h]}{a_h [q,h]} \tag{16b}
$$