

Fitting con Gnuplot

Introducción

El ajuste de parámetros (ajuste de curvas, o *curve fitting*) consiste en determinar los parámetros de un modelo a partir de datos experimentales de forma tal que el modelo se ajuste lo mejor posible a la realidad.

Un método de uso extendido para modelos lineales es el de mínimos cuadrados (por ejemplo Gauss-Newton); mientras que para modelos no lineales las soluciones más habituales consisten en una sucesión iterativa de linealizaciones locales del problema.

Siendo estos algoritmos complejos y requiriendo cálculos en varios pasos, es deseable una solución informática que permita extraer resultados de forma simple.

Gnuplot implementa el método de Marquart-Levenberg de mínimos cuadrados no lineales, que puede ajustarse a cualquier problema con las restricciones típicas:

- una cantidad de datos experimentales igual o mayor a la de variables a ajustar en el modelo, y
- una estimación inicial de la solución, de la cual depende la convergencia a la solución hallada.

Problema de ejemplo

Se desea ajustar una oscilación de amplitud exponencialmente decreciente descrita por la curva

$$f(t) = Ae^{-t/\tau} \cos(\omega t + \varphi),$$

donde A , τ , ω y φ son constantes a determinar, a partir de una serie de medidas en el intervalo $t \in [0, 1]$ s.

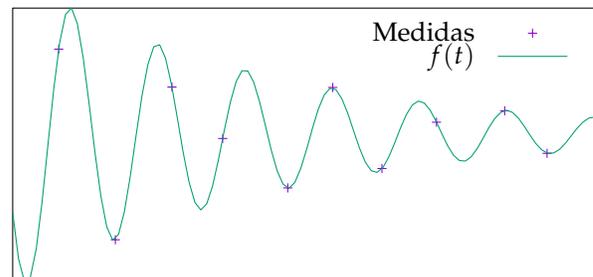
Para esto se escribe un *script* (*ejemplo.gnuplot*) donde se modela la función y se dan los datos iniciales. Se usa el comando `fit` para ejecutar el ajuste, dando como resultando los valores óptimos para los parámetros. El comando `plot` se usa para obtener un gráfico comparativo.

<pre>ejemplo.gnuplot: set dummy t # variable independiente 't' F(t) = A * exp(-t/T) * cos(w*t + phi) # Estimación inicial A = 10 w = 2*pi * 5.5 # Hz T = .500 # s phi = pi/2 # rad fit F(t) "medidas.txt" via A, w, T, phi plot "medidas.txt", F(t)</pre>	<pre>medidas.txt: .087 7.88 .194 -9.87 .301 4.35 .397 -4.38 .520 -5.04 .605 4.31 .698 -3.23 .801 1.08 .930 2.14 1.01 -1.82</pre>
---	---

Resultado:

```
$ gnuplot ejemplo.gnuplot
```

```
A      = 14.6816  +/- 0.05235 (0.3566%)
w      = 38.089  +/- 0.01611 (0.04228%)
T      = 0.490084 +/- 0.001943 (0.3964%)
phi    = 2.09174 +/- 0.007024 (0.3358%)
```



Gnuplot

Gnuplot es una utilidad de línea de comando cuya principal funcionalidad es de crear gráficas.

Como fuentes de datos para estas gráficas puede usar tanto conjuntos de puntos (típicamente datos experimentales) como funciones matemáticas explícitas (por ejemplo $f(x) = \sin t$, o combinaciones arbitrarias de las mismas). Como enlace entre estas dos funcionalidades, *Gnuplot* es capaz de resolver problemas de ajuste de parámetros (como funcionalidad secundaria).

Algunas ventajas de usar *Gnuplot* para resolver estos problemas son:

- Software *open source* de libre uso, disponible en (por ejemplo) cualquier *distro* Linux accesible.
- Lenguaje de *scripting* accesible, con lenguaje matemático similar librería "C" (`cos()`, `sqrt()`, etc.).
- Manejo de complejos, incluyendo pasaje a reales de forma idiomática (`abs()` y `arg()` para módulo y argumento, `real()` e `imag()` para partes real e imaginaria, etc.).
- Permite ejecutar rutinas de ajuste y graficar en un mismo *script* reusando el código; y
- exporta gráficos a diversos medios incluyendo imágenes y código \LaTeX embebible; siendo estas dos últimas una buena combinación a la hora de redactar un informe.

Gnuplot tiene una completa ayuda en línea, que se puede consultar desde el programa. Para cada comando, `help <comando>` imprime ayuda y ejemplos. Por ejemplo, `help fit`, `help set xlabel`.

Problema de ajuste: filtro pasabanda

Dado cierto amplificador con un filtro pasabanda, se desea hallar las frecuencias de corte y la ganancia en banda pasante a partir de una serie de medidas de amplitudes de entrada y salida a distintas frecuencias.

Éstas se disponen en un archivo de texto en tres columnas, para poder trabajarlo con *Gnuplot* (el espaciado es arbitrario, se puede usar tabulador, varios espacios, etc.):

pasabanda.txt:

f (Hz)	v_i (mV)	v_o (mV)
10300	72.7	957
12300	72.5	1040
30400	71.9	1360
102000	73.5	1480
178000	73.6	1470
302000	74.3	1420
538000	72.6	1290
1060000	74.2	1040
1770000	73.2	760
3190000	82.5	483

Entrada y salida se relacionan con la frecuencia a través de la transferencia $H(s) = \frac{V_o(s)}{V_i(s)}$, que para el filtro pasabanda para entradas sinusoidales se puede aproximar a frecuencias medias por

$$H(j\omega) \approx G \frac{j\omega}{j\omega + \omega_1} \frac{\omega_2}{j\omega + \omega_2}$$

donde G es la ganancia en banda pasante y ω_1 y ω_2 son los polos que limitan la banda; parámetros que se desean hallar mediante mínimos cuadrados no lineales usando la relación entre los datos $|H(2\pi jf)| = \frac{v_o}{v_i}$.

Se comienza describiendo la función $H(j\omega)$ en *Gnuplot*, aprovechando su manejo de números complejos:

```
j = {0,1} # Gnuplot representa a + bi como {a,b}
H(w) = G * j*w / (j*w + w1) * w2 / (j*w + w2)
```

donde se usa j para la unidad imaginaria por comodidad.

Se provee una estimación inicial para G , ω_1 y ω_2 a partir de los datos:

```
G = 1480 / 73.5
w1 = 1e4
w2 = 1e6
```

y se usa el comando `fit` para ejecutar el algoritmo:

```
set dummy f
fit abs(H(2*pi*f)) "pasabanda.txt" using ($1):($3/$2) via w1, w2, G
```

donde `set dummy f` indica que la variable independiente será f : por defecto, *Gnuplot* usa x por lo que si se quiere usar otro símbolo debe indicarse de esta forma. Es puramente estético, para remarcar que se trata de una frecuencia.

El significado de cada parámetro es:

`abs(H(2*pi*f))` Se usa la función `abs()` para obtener el módulo de la función compleja $H(w)$. Notar que como la frecuencia está en hercios en el archivo de datos, se la multiplica por 2π antes de pasarla como parámetro. Otra forma de resolver esto hubiera sido expresar $H(f)$ en función de la frecuencia en hercios.

`"pasabanda.txt"` El segundo parámetro es el nombre del archivo de datos.

`using ($1):($3/$2)` Este parámetro indica cómo obtener los datos desde el archivo, donde n indica la n -ésima columna. La expresión a la izquierda de los dos puntos indica la variable independiente en el algoritmo; para este problema la frecuencia. La expresión a la derecha indica el valor a comparar con la función `abs(H(2*pi*f))`, el primer parámetro de `fit`.

`via w1, w2, G` Finalmente se indican qué variables se deben ajustar: en este caso las frecuencias de corte y la transferencia. Pueden ser tantas o tan pocas como sea necesario, siendo el requisito que la cantidad de variables de ajuste debe ser menor o igual a la cantidad de datos contra los que se ajusta.

En suma, el primer parámetro es la función analítica a usar y el parámetro `using` indica cómo obtener los valores experimentales a partir del archivo.

Script completo, incluyendo imprimir los resultados y generar un gráfico:

pasabanda.gnuplot:

```
set fit quiet                # Evitar ruido en la salida
j={0,1}                     # unidad imaginaria
H(w) = G * j*w/(j*w + w1) * w2/(j*w + w2)
# Estimación inicial
w1 = 2*pi * 1e4
w2 = 2*pi * 1e6
G = 20
set dummy f                 # usar f como variable independiente
fit abs(H(2*pi*f)) "pasabanda.txt" using ($1):($3/$2) via w1, w2, G

# Imprimir resultados
print sprintf("Frecuencias de corte: %.3g y %.3g", w1/(2*pi), w2/(2*pi))
print sprintf("Ganancia: %.3g", G)

# Imprimir un gráfico
set terminal latex
set output "pasabanda.tex"  # Archivo de salida
dB(x) = 20*log10(x)         # Para mostrar decibeles
set logscale x              # Escala logarítmica en ambos ejes
set logscale y
set xlabel '$f$ (Hz)'      # Etiquetas en los ejes
set ylabel '\rotatebox{90}{$|H(j\omega)|$ (dB)}'
plot [1e3:1e7] "pasabanda.txt" using ($1):(dB($3/$2)) notitle, \
    dB(abs(H(2*pi * f))) title '$|H(j\omega)|$'
```

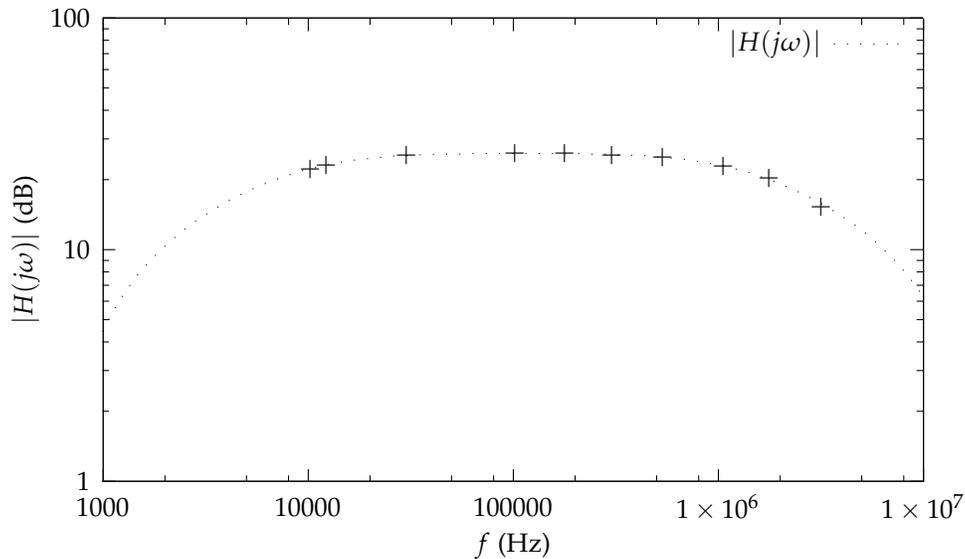


Figura 1: *Fitting* del pasabanda

La salida de ejecutar el *script* es:

```
$ gnuplot pasabanda.gnuplot
Frecuencias de corte: 1.21e+04 y 1.02e+06
Ganancia: 20.2
```

y el archivo \LaTeX `pasabanda.tex`, que se inserta en el documento (fig. 1):

```
\begin{figure}
\begin{center}
\input{pasabanda.tex}
\caption{\emph{Fitting} del pasabanda}
\label{fig:pasabanda}
\end{center}
\end{figure}
```

Referencias y enlaces

- Documentación *Gnuplot*
<http://www.gnuplot.info/documentation.html>
- Tutorial *Gnuplot* (en inglés, incluye plot y fit)
http://physics.ucsc.edu/~medling/programming/gnuplot_tutorial_1/index.html
- Módulo \LaTeX para *Gnuplot*
<http://www.gnuplot.info/files/tutorial.pdf>
- P. F. Mondragon, *A Comparison of Non-linear Regression Codes*, New Mexico Institute of Mining and Technology (2003)
<http://euler.nmt.edu/~brian/students/mondragon.pdf>