

IMPLEMENTACIÓN DE FILTRO ADAPTATIVO PARA PREDICCIÓN DE SEÑAL CON RUIDO

FEDERICO FERREYRA¹, ALFREDO TAPIA²

Procesamiento Digital de Señales, Ingeniería de sonido
Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina

¹ federicoferreyra65@gmail.com, ² fredy989@gmail.com

Resumen

En este informe se documenta la implementación de un filtro adaptativo de Kalman, apuntando a la predicción del nivel de presión sonora SPL (dBSPL) a una cierta distancia de la fuente y en el cual la medición puede estar sujeta a niveles de ruidos altos. A partir del modelo físico de divergencia de ondas esféricas, del conocimiento de la ecuación de la señal utilizada y sus parámetros y de una medición considerada sin ruido a una distancia conocida de la fuente (1 metro), el filtro permite predecir el nivel de presión sonora a otra distancia de la fuente (2 metros). El desarrollo así como la implementación de dicho filtro se realiza en Python 3.7, y consta de 3 partes: la importación e implementación de mediciones y modelos matemáticos, la preparación de los parámetros iniciales del filtro y por último la implementación de la fase recursiva del mismo en orden de obtener la mejor predicción posible.

This report documents the implementation of a Kalman adaptive filter, looking for to the prediction of the sound pressure level SPL (dBSPL) at a certain distance from the source and in which the measurement may be subject to high noise levels. Based on the physical model of spherical wave divergence, on the equation of the signal used and its parameters and on a measurement considered without noise at a known distance from the source (1 meter), the filter allows to predict the sound pressure level at another distance from the source (2 meters). The development as well as the implementation of that filter is done in Python 3.7, and consists in 3 steps: importation and implementation of measurements and mathematical models, preparation of the initial parameters of the filter and finally the implementation of the recursive phase to obtain the best possible prediction.

1. INTRODUCCIÓN

El siguiente trabajo se basa en la aplicación de un filtro adaptativo de Kalman en el campo de la acústica en campo libre, es decir, sin apreciación de los efectos reverberativos de un recinto. A través de un algoritmo recursivo, se puede predecir con relativa precisión, el nivel dBSPL de una señal que por caso no se pudo o no se puede medir en campo. Los elementos a computar se basan entonces en mediciones hechas de correcta aceptación y lo que se puede establecer a partir de los modelos matemáticos/físicos de campo de la acústica. Por caso, la divergencia de ondas esféricas, tanto relativa a presión sonora o nivel de presión sonora. Además de estas herramientas que puedan establecer una predicción calculable a una determinada distancia, se conocen la ecuación y parámetros so-

bre la fuente sonora, permitiendo entonces realizar otra aproximación de diferentes características. El objetivo del trabajo es puntualmente, obtener una predicción del nivel de presión sonora a 2 metros de la fuente. Teniendo una medición de poca aceptación a dicha distancia por altos niveles de ruido, el procedimiento consiste en aplicar los ya mencionados modelos matemáticos a otra medición aceptable hecha a 1 metro de distancia para obtener una primera aproximación. Por otro lado, se logra una segunda estimación del nivel de ruido a 2 metros mediante un cálculo directo a partir de los conocimientos de la señal original (fuente), y más propiamente dicho, de sus parámetros como frecuencia fundamental, armónicos y amplitudes de estos últimos. El trabajo será entonces estructura en 3 partes subsiguientes. Primero un repaso teórico de lo necesario a la comprensión del trabajo, es decir:

explicación general de la implementación del filtro de Kalman, su implementación orientada al campo de interés y conceptos de acústica. A continuación se hará un desarrollo paso a paso de la implementación del código del algoritmo en Python. Los arreglos previos necesarios trabajando sobre las señales de interés, simulación de los modelos acústicos para finalmente llegar a los parámetros necesarios al inicio de la iteración del filtro. Se cerrará el informe con gráficos ilustrativos de los resultados obtenidos y análisis de los mismos. Una conclusión del trabajo dará las impresiones generales sobre este tipo de procedimiento y la fiabilidad de sus resultados.

2. CONCEPTOS TEÓRICOS

CONCEPTOS ACÚSTICOS

La divergencia de una onda esférica, es el caso más general de propagación de una onda desde un emisor hacia un receptor. El concepto se basa en que las ondas sonoras, van debilitándose en amplitud a medida que se van alejando del punto de origen, es lo que se conoce como atenuación de la onda. Es decir, la variación de presión sonora será más grande cerca de la fuente e irá reduciendo a medida que el punto de observación se alejara de la fuente. La onda acústica con una divergencia de tipo esférica, tiende a seguir una ley llamada “ley cuadrática inversa”: en espacio libre, al doblar la distancia el nivel de energía disminuye 6 dB. [2]

$$\Delta dB = 20 \log\left(\frac{2d}{d}\right) = 20 \log(2) = 6dB \quad (1)$$

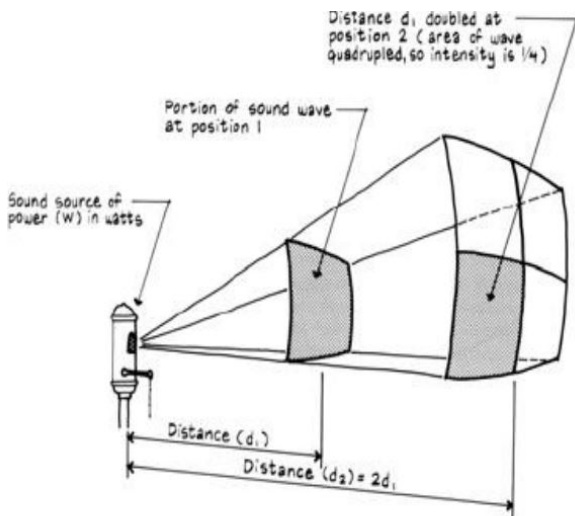


Figura 1: Divergencia esférica

FILTRO DE KALMAN

El filtro de kalman, es un algoritmo desarrollado por Rudolf E. Kalman en 1960. El algoritmo estima el estado de un sistema a partir de datos medidos, primeramente se predice el estado del sistema, luego se utiliza las mediciones de ruido para ajustar la estimación del estado del sistema. Ya que se trata de un algoritmo de carácter recursivo, puede correr en tiempo real usando únicamente las mediciones de entrada actuales, el estado calculado previamente y su matriz de incertidumbre, lo que permite un ahorro considerable de la memoria a la hora de ser implementado en un sistema digital. Actualmente se utiliza para aplicaciones que dependen de la estimación, como la visión artificial, los sistemas de guía y navegación, la econometría y el procesamiento de señales. A partir de esto es importante entender el funcionamiento, para lograr las herramientas que permitan la solución de problemas, de forma óptima.[3]

Se propone el concepto del filtro de Kalman, de manera de lograr valores más óptimos mediante el proceso. Suponiendo que se cuenta con una estimación t_2 , realizamos un nuevo estimador a priori t_3^* , utilizando el concepto de la divergencia esférica.

$$\hat{x}(t_3^-) = \hat{x}(t_2) - 6dB \quad (2)$$

Luego se calcula el estimador a priori óptimo.

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K[(z_3) - \hat{x}(t_3^-)] \quad (3)$$

Donde Z_3 representa la medición registrada a dos metros, y $K(t_3)$ es el valor de ponderación.

$$K(t_3) = \sigma_x^2(t_3^-) / [\sigma_x^2(t_3^-) + \sigma_{z_3}^2] \quad (4)$$

El valor de ponderación K , está acotada entre cero y uno.[1]

3. CÓDIGO EN PYTHON

El objetivo es estudiar la aplicación del filtro Kalman para la predicción del nivel SPL en un punto del espacio, teniendo en cuenta que el mismo se encuentra sujeta a altos niveles de ruido, para este fin se eligió Python 3.7. En una primera instancia se cargan las librerías pertinentes. La denominada librosa para poder importar archivos de audio, luego numpy, ya que si bien python cuenta con algún soporte matemático, esta extensión le agrega mayor versatilidad en ese aspecto, y finalmente matplotlib para graficar los resultados obtenidos.

Se cuenta con una medición de audio a un metro sin ruido, y una a dos metros con ruido, y una señal de calibración. Una vez cargadas las mediciones y la calibración se emparejan el largo de ambas mediciones, eligiendo la largo de la señal mas corta como

referencia. Vale aclarar que las 3 señales importadas tienen una frecuencia de muestreo de $44100Hz$. Por otra parte, se tienen los parámetros matemáticos de la fuente sonora original (la correspondiente a las mediciones).

$$x[n] = \sum_{r=1}^R A_r \cos(2\pi F_0 n + \phi_r)$$

en donde:

F_0 : es la frecuencia fundamental de la señal.

R : es el número de parcial superior.

n : muestra discreta de tiempo.

A_r : es la amplitud de cada parcial r .

ϕ_r : es la fase de cada parcial r .

Figura 2: Parámetros de la señal original

De estos parámetros, se obtienen entonces la ecuación discretizada de la presión en función del tiempo, así como su derivada. Ambas funciones, se plasman entonces en las siguientes 2 funciones del código:

```
44 def x_distancia(r):
45     x = np.zeros(n)
46     for i in range(len(Fr)):
47         x = x + Ar[i]*np.cos(2*np.pi*F0*Fr[i])*np.exp(-1j*r*Fr[i]/343)/r
48     return x.real
49
50 def dx_distancia(r):
51     dx = np.zeros(n)
52     for i in range(len(Fr)):
53         dx = dx + 2*Ar[i]*Fr[i]*np.pi*F0*np.exp(-1j*r*Fr[i]/343)*np.sin(2*np.pi*Fr[i]*t)
54     return dx.real
```

Figura 3: Funciones de presión respecto del tiempo y su derivada

Estas 2 funciones, tienen incluidas el factor de decaimiento respecto de la distancia, que se calcula para cada armónico. Este factor aporta entonces un elemento suplementario de precisión a la hora de plasmar el modelo de divergencia de una onda esférica (para ecuación sin derivar. Por caso, con estos parámetros, el valor obtenido que caracteriza (en dB) el decaimiento al doble de la distancia es de unos $7,5dB$, cercano a los $6dB$ esperados.

La aplicación del filtro se plasma para ser realizada en dB , por lo cual las señales son tratadas previamente. En un primer paso, se elige una ventana para el paso a valores RMS correspondiente al período (de muestras) mas grande de los 5 armónicos presente en la señal de origen, resultando ser una ventana de 441 muestras. De esta manera, podemos asegurar que los valores obtenidos en el nuevo subconjunto de las mediciones reflejan fielmente un valor RMS . Esos nuevos vectores se calibran mediante un valor RMS global (extraído de la señal de calibración con la misma función RMS pero variando la ventana al total del largo de dicha señal) para luego ser pasados a $dB SPL$.

```
98 # PARÁMETROS PRINCIPALES PASO A dB
99 m1_dB = PtoDB(med1m,n0,n)
100 m2_dB = PtoDB(med2m,n0,n)
101 x1_dB = PtoDB(x_distancia(1),n0,n)
102 x2_dB = PtoDB(x_distancia(2),n0,n)
103 dx1_dB = PtoDB(dx_distancia(1),n0,n)
104 dx2_dB = PtoDB(dx_distancia(2),n0,n)
105
106 # PARÁMETROS INICIALES DEL FILTRO (dB)
107 aprox = m1_dB
108 var = varianza(aprox)
109 medicion = m2_dB
110 x_modelo = x2_dB - x1_dB
111 dx_modelo = dx2_dB - dx1_dB
```

Figura 4: Parámetros iniciales del filtro

Los parámetros iniciales del filtro se envían a nombres de entradas convenientes para la aplicación de las funciones de filtrado (Kalman). La primera función (`kalman1`) tiene como salida un estimador óptimo con el modelo matemático de divergencia de onda esférica. La segunda en cambio, usa la derivada del modelo matemático de la ecuación de la onda. Sobre esta última, se aplica la iteración del filtro (por lo cual se define el filtro de Kalman como recursivo).

```
79 def kalman1(entrada,var_entrada,medicion,modelo1,ruido=0):
80     # MODELO 1
81     apriori1 = entrada + modelo1
82     var_apriori1 = var_entrada + varianza(modelo1)
83     K1 = var_apriori1 / (var_apriori1 + varianza(medicion) + varianza(ruido))
84     aposteriori1 = apriori1 - K1*(medicion - apriori1)
85     var_aposteriori1 = (1-K1)*var_apriori1
86     return aposteriori1,apriori1,var_aposteriori1,K1
87
88 def kalman2(entrada,var_entrada,medicion,modelo2,ruido=0):
89     # MODELO 2
90     apriori2 = entrada - modelo2
91     var_apriori2 = var_entrada + varianza(modelo2)
92     K2 = var_apriori2 / (var_apriori2 + varianza(medicion) + varianza(ruido))
93     aposteriori2 = apriori2 + K2*(medicion - apriori2)
94     var_aposteriori2 = (1-K2)*var_apriori2
95     # SALIDAS
96     return aposteriori2,apriori2,var_aposteriori2,K2
```

Figura 5: Funciones de Kalman

Vale aclarar, que para ambas funciones, se simula un ruido aleatorio del cual la varianza es entrada en ambas funciones de kalman, dando así, una mejor aproximación a los estimadores.

```
120 # RUIDO ALEATORIO
121 ruido = np.random.normal(0,1,n)
122
123 ### FILTRADO ###
124 veces = 100
125 (aprox,sp0,var,K1) = kalman1(aprox,var,medicion,x_modelo,ruido)
126 for ejecucion in range(veces):
127     (aprox,sp,var,K2) = kalman2(aprox,var,medicion,dx_modelo,ruido)
```

Figura 6: Aplicación del filtrado

Finalmente, se grafican las señales, para obtener los resultados del proceso.

4. Resultados y conclusiones

La aplicación del filtro de Kalman en lenguaje de programación resulta exitosa. Se obtiene un filtro estable y convergente, con un criterio de aplicación que resulta funcional dados los parámetros

conocidos y mediciones que se usan de punto de partida.

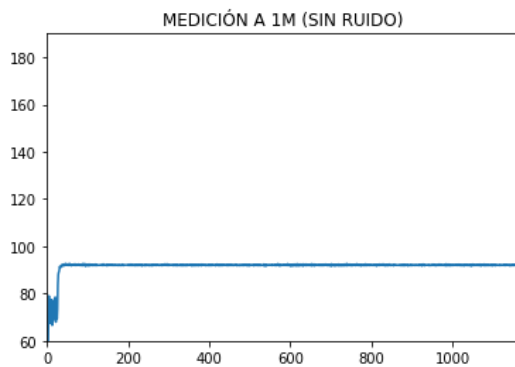


Figura 7: Nivel de presión sonora de la medición a 1 metro sin ruido

Algunos aspectos preocupan sin embargo, en cuánto a la validez de los resultados y las dificultades para corroborar resultados obtenidos.

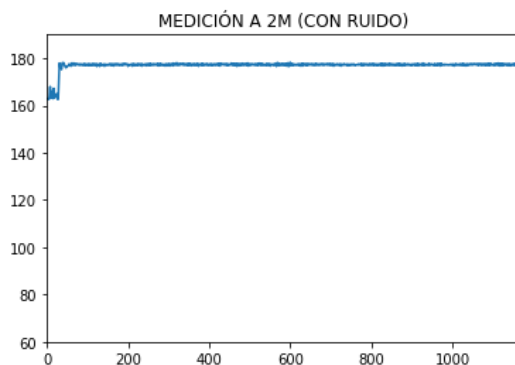


Figura 8: Nivel de presión sonora de la medición a 2 metros con ruido

Esta claro que la aplicación de diferentes modelos predictivos ayuda a obtener un estimador óptimo del nivel $dB SPL$ a 2 metros con menor cantidad de iteraciones del filtro.

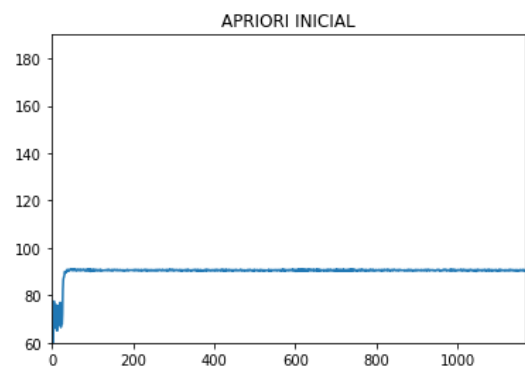


Figura 9: Nivel de presión sonora del primer estimador apriori a 2 metros

Basandose en un primer estimador apriori, el filtro eventualmente converge hacia el nivel de presión sonora esperado, dando como válido varios aspectos tratados en el trabajo.

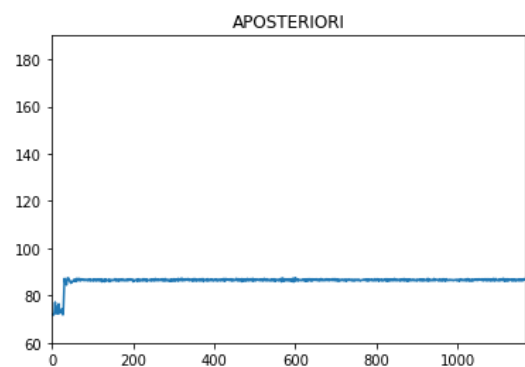


Figura 10: Nivel de presión sonora del estimador óptimo a 2 metros con 100 iteraciones

Concluyendo, se da por válido gran parte de lo hecho en este trabajo. Sin embargo algunos aspectos, tanto relativos a la teoría del aspecto iterativo del filtro de Kalman como de su aplicación, merecen más tiempo de atención para afirmar que el código presentado adquiere relevancia para una real aproximación en otras condiciones.

Referencias

- [1] Roger R. Labbe Jr. *Kalman and Bayesian Filters in Python*. 2018.
- [2] Florent Masson. *Acústica y Psicoacústica 1, UNTreF*. 2017.
- [3] Peter S. Maybeck. *Stochastic models, estimation, and control*. 1979.