

# Modelo circuital de la cóclea

Federico Miyara

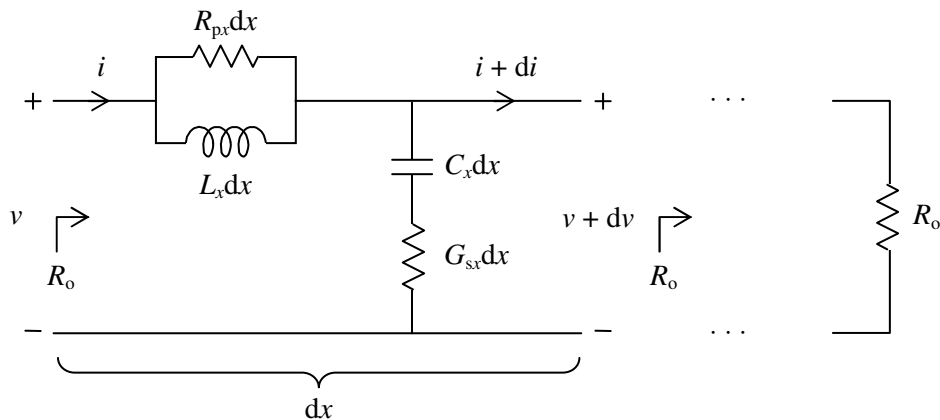
## Introducción

Analizaremos un modelo circuital de la cóclea propuesto por Deutsch y Micheli-Tzanakou (1987), citado en Deutsch et al. (1994). El mismo parte de una línea de transmisión con parámetros variables que en principio nos permitirá simular la oscilación de la membrana basilar y así apreciar las ondas viajeras y su envolvente en función de la posición. Los parámetros mecánicos tales como la masa, la resistencia viscosa y la compliancia mecánica serán reemplazados por sus analogías eléctricas inductancia, resistencia y capacidad, respectivamente, en tanto que las variables velocidad  $u$  y fuerza  $f$  serán análogas a la corriente  $i$  y la tensión  $v$ . Para simplificar la conversión se considerarán constantes unitarias, de modo que los valores correspondientes en el dominio mecánico y eléctrico sean numéricamente iguales. Así, una corriente de 1 A corresponderá a una velocidad de 1 m/s y una tensión de 1 V, a 1 N.

Consideremos entonces una línea de transmisión con una inductancia por unidad de longitud  $L_x$ , una resistencia paralelo por unidad de longitud  $R_{px}$ , una capacidad por unidad de longitud  $C_x$  y una conductancia serie por unidad de longitud  $G_{sx}$ . En la figura 1 se muestra el modelo de un segmento diferencial de longitud  $dx$  y la carga final  $R_o$ . Supondremos que todos los parámetros dependen de la posición  $x$  a lo largo de la cóclea (donde  $x = 0$  corresponde a la entrada en la ventana oval y  $x = 0,035$  m corresponde al helicotrema o ápex) y adoptaremos la hipótesis simplificadora de que la impedancia vista desde cualquier punto de la línea es siempre resistiva y de valor  $R_o$ . Para ello plantearemos la impedancia vista a la entrada del elemento diferencial y la igualaremos a  $R_o$ :

$$\frac{sL_x dx R_{px} dx}{sL_x dx + R_{px} dx} + \frac{R_o \left( \frac{1}{sC_x dx} + \frac{1}{G_x dx} \right)}{R_o + \frac{1}{sC_x dx} + \frac{1}{G_x dx}} = R_o. \quad (1)$$

Operando,



**Figura 1.** Un segmento elemental de longitud  $dx$  del modelo de la cóclea como una línea de transmisión con pérdidas.

$$\frac{sL_x R_{px}}{sL_x + R_{px}} dx + \frac{R_o}{R_o \frac{sC_x G_{sx}}{sC_x + G_{sx}} dx + 1} = R_o. \quad (2)$$

Dado que  $dx$  es arbitrariamente pequeño, podemos aproximar el segundo término del primer miembro por un polinomio de Taylor de primer grado, resultando

$$\frac{sL_x R_{px}}{sL_x + R_{px}} dx + R_o \left( 1 - R_o \frac{sC_x G_{sx}}{sC_x + G_{sx}} dx \right) \cong R_o. \quad (3)$$

Operando, se obtiene, en el límite,

$$\frac{sL_x R_{px}}{sL_x + R_{px}} = R_o^2 \frac{sC_x G_{sx}}{sC_x + G_{sx}}. \quad (4)$$

Quitando denominadores e igualando coeficientes resulta

$$\frac{L_x}{C_x} = R_o^2, \quad (5)$$

$$\frac{R_{px}}{G_{sx}} = R_o^2. \quad (6)$$

Pasando al dominio fasorial y llamando, para una frecuencia de entrada  $f$ ,  $Z_x(f)$  a la impedancia por unidad de longitud formada por  $R_{px}$  y  $L_x$  en  $\Omega/m$ ,

$$Z_x(f) = \frac{j2\pi f L_x R_{px}}{j2\pi f L_x + R_{px}}, \quad (7)$$

se encuentra (Deustch y Deutsch, 1994) que se puede lograr un buen ajuste con los resultados experimentales si<sup>1</sup>

$$Z_x(f, x) = 0,39 f^2 e^{-100\sqrt{L-x}} + j0,1325 f e^{-20\sqrt{L-x}} \quad (8)$$

donde  $f$  es la frecuencia,  $L = 0,035$  m es la longitud de la cóclea y  $x$  la posición en m a lo largo de la cóclea a partir de la ventana oval (ubicada en la espira basal). Análogamente, la admitancia por unidad de longitud

$$Y_x(f) = \frac{j2\pi f C_x G_{px}}{j2\pi f C_x + G_{px}}, \quad (9)$$

será igual, según (4), a

---

<sup>1</sup> Es fácil constatar, multiplicando y dividiendo el segundo miembro de la ecuación (7) por  $R_{px} - j2\pi f L_x$ , que la dependencia de la frecuencia de la impedancia por unidad de longitud  $Z(f)$  es de la forma  $a f^2 + j b f$ . Los coeficientes  $a$  y  $b$ , dependientes de  $x$  se eligieron para un ajuste óptimo a los picos de resonancia.

$$Y_x(f) = \frac{1}{R_o} Z_x(f). \quad (10)$$

Los valores individuales de los parámetros  $L_x$ ,  $R_{px}$ ,  $C_x$  y  $G_{sx}$ , aunque podrían despejarse de las ecuaciones (7), (8), (9) y (10), no son necesarios.

Podemos ahora plantear las siguientes ecuaciones en transformada de Laplace

$$i = -\left(\frac{dv}{R_{px} dx} + \frac{dv}{sL_x dx}\right), \quad (11)$$

$$v + dv = (i + di)R_o. \quad (12)$$

La primera puede reescribirse como

$$i = -\left(\frac{1}{R_{px}} + \frac{1}{sL_x}\right) \frac{dv}{dx}, \quad (13)$$

Con respecto a la segunda, dado que  $v = R_o i$ , la misma relación resulta entre las derivadas, por lo que

$$\frac{dv}{dx} = R_o \frac{di}{dx}. \quad (14)$$

Estamos interesados en obtener la respuesta en función de la distancia suponiendo una excitación de corriente (correspondiente a una excitación de velocidad en la ventana oval). Para ello obtengamos primero la ecuación diferencial que gobierna el sistema. De las ecuaciones (13) y (14) podemos obtener

$$i = -\left(\frac{1}{R_{px}} + \frac{1}{sL_x}\right) R_o \frac{di}{dx}, \quad (15)$$

es decir,

$$\frac{di}{dx} + \frac{1}{R_o} \frac{sL_x R_{px}}{sL_x + R_{px}} i = 0. \quad (16)$$

Si la entrada es senoidal de frecuencia  $f$ , podemos reemplazar el coeficiente de  $i$  mediante (7), resultando:

$$\frac{di}{dx} + \frac{Z_x(f, x)}{R_o} i = 0. \quad (17)$$

Esta es una ecuación diferencial a coeficientes variables que puede resolverse fácilmente. Para ello observemos que

$$\frac{d \ln(i)}{d x} = \frac{1}{i} \frac{d i}{d x}, \quad (18)$$

de donde resulta, comparando con (17),

$$\frac{d \ln(i)}{d x} = -\frac{Z_x(f, x)}{R_o}. \quad (19)$$

Integrando,

$$\ln(i) = -\int \frac{Z_x(f, x)}{R_o} d x + \ln(I_o). \quad (20)$$

Finalmente,

$$i = I_o e^{-\int \frac{Z_x(f, x)}{R_o} d x}. \quad (21)$$

Esto, juntamente con (8) nos proporciona una expresión para la corriente en función de la posición. Para ello necesitamos obtener integrales del tipo

$$\int e^{a\sqrt{L-x}} d x, \quad (22)$$

para lo cual sustituimos

$$w = \sqrt{L-x}, \quad (23)$$

de donde resulta  $dx = -2w dw$ , por lo tanto

$$\int e^{a\sqrt{L-x}} d x = -2 \int e^{aw} w d w. \quad (24)$$

Esta integral se resuelve integrando por partes<sup>2</sup>

$$\int e^{aw} w d w = \frac{1}{a} e^{aw} w - \frac{1}{a^2} e^{aw} = e^{aw} \left( \frac{w}{a} - \frac{1}{a^2} \right), \quad (25)$$

por lo tanto, volviendo a la variable original  $x$ ,

$$\int e^{a\sqrt{L-x}} d x = \frac{2}{a^2} e^{a\sqrt{L-x}} \left( 1 - a\sqrt{L-x} \right). \quad (26)$$

Combinando las ecuaciones (8), (21) y (26) se llega a que

$$i = I_o e^{-\left( 0,039 f^2 e^{-100\sqrt{L-x}} (1 + 100\sqrt{L-x}) + j 0,033125 f e^{-20\sqrt{L-x}} (1 + 20\sqrt{L-x}) \right)} + K \quad (27)$$

<sup>2</sup> Para obtener una primitiva que se anule en  $y = 0$  bastará sumar  $1/a^2$ .

donde  $K$  es una constante de integración (compleja) que se elegirá para que en  $x = 0$  se tenga una corriente igual a  $I_0$  independientemente de la frecuencia. Para hallarla simplemente igualamos el exponente a 0 (en parte real e imaginaria). Llamando

$$K = K_1 f^2 + jK_2 f, \quad (28)$$

resulta

$$\begin{aligned} K_1 &= 5.765 \times 10^{-10}, \\ K_2 &= 0.0037248. \end{aligned} \quad (29)$$

### Desplazamiento de la membrana basilar

A partir de la corriente  $i(x, f)$ , que, por la analogía adoptada, es numéricamente igual a la velocidad del líquido (endolinfa), podemos calcular la velocidad de desplazamiento de la membrana basilar, tal como se ilustra esquemáticamente en la figura 2.

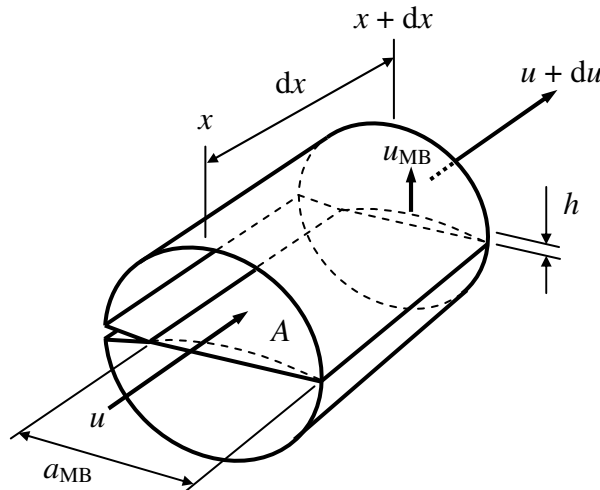
Si  $u$  es la velocidad de la endolinfa que ingresa en un elemento diferencial y  $A$  es el área transversal de la ramba vestibular, entonces el caudal  $q$  que ingresa en un elemento diferencial de largo  $dx$  es

$$q = Au. \quad (30)$$

Si consideramos que el líquido es esencialmente incompresible, ya que despreciamos el fenómeno de propagación de ondas en el líquido (que es irrelevante para el funcionamiento de la cóclea), el incremento de caudal  $dq$ , dado por

$$dq = Adu, \quad (31)$$

se deberá al desplazamiento de la membrana basilar perpendicular al eje de la cóclea. Llamando  $u_{MB}$  a la velocidad media de la cóclea resulta, entonces



**Figura 2.** Desplazamiento vertical de la membrana basilar asociado al incremento de caudal entre la entrada y la salida de un elemento diferencial de longitud  $dx$ . La cuña que se ve hacia la izquierda es la *lámina espiral*, por donde pasa el nervio auditivo. Nota: por simplicidad del dibujo no se muestra la membrana de Reissner.

$$dq = u_{MB} a_{MB} dx, \quad (32)$$

donde  $a_{MB}$  es el ancho de la membrana basilar. Igualando (31) y (32) resulta

$$u_{MB} = \frac{A}{a_{MB}} \frac{du}{dx}. \quad (33)$$

El desplazamiento  $y_{MB}$  de la membrana basilar en dirección normal, promediado sobre la línea transversal ubicada en la posición  $x$ , puede obtenerse integrando la velocidad con respecto al tiempo, lo cual para señales senoidales equivale a dividir por  $j\omega$ :

$$y_{MB} = \frac{A}{j2\pi f a_{MB}} \frac{du}{dx}. \quad (34)$$

Ahora bien, dado que en un rango bastante amplio de frecuencias el desplazamiento  $y_{VO}$  de la ventana oval es aproximadamente proporcional a la presión sonora en el tímpano, resulta de mayor interés la función de transferencia entre  $y_{VO}$  e  $y_{MB}$ , la cual depende también de  $x$ ,

$$H(f, x) = \frac{y_{MB}}{y_{VO}}. \quad (35)$$

Podemos obtener el desplazamiento  $y_{VO}$  integrando respecto al tiempo la velocidad en  $x = 0$  que, nuevamente, para señales senoidales equivale a dividir por  $j\omega$ :

$$y_{VO} = \frac{U_o}{j2\pi f}, \quad (36)$$

donde  $U_o$  es la velocidad de desplazamiento de la membrana oval ( $x = 0$ ), numéricamente igual a  $I_o$ . Entonces, de (34) y (36) resulta

$$H(f, x) = \frac{A}{a_{MB} U_o} \frac{du}{dx}. \quad (37)$$

De (17), y teniendo en cuenta las analogías,

$$\frac{du}{dx} = -\frac{Z_x(f, x)}{R_o} u. \quad (38)$$

Sustituyendo en (37)

$$H(f, x) = -\frac{A}{a_{MB}} \frac{Z_x(f, x)}{R_o} \frac{u}{U_o}, \quad (39)$$

donde  $u$  puede obtenerse de (27) sustituyendo  $i$  por  $u$  e  $I_o$  por  $U_o$ , debido a la analogía y a la elección de constante unitaria entre variables análogas. Resulta

$$u = U_0 e^{-\left(0,039 f^2 e^{-100\sqrt{L-x}} (1 + 100\sqrt{L-x}) + j 0,033125 f e^{-20\sqrt{L-x}} (1 + 20\sqrt{L-x})\right)} + K \quad (40)$$

Para cada frecuencia, el módulo de la función de transferencia de la ecuación (39) presenta un máximo en una posición específica dentro de la cóclea.

En la figura 3 se muestran las envolventes de la oscilación de la membrana basilar en función de  $x$  para cuatro valores de frecuencia, pudiéndose apreciar los picos de máxima oscilación. Las envolventes se grafican representando  $|H(f_i, x)|$  y normalizando cada envolvente de modo que su máximo sea 1.

Dado que  $H(f, x)$  tiene módulo y fase, si fijamos la frecuencia en un valor  $f_0$ , tendremos una fase diferente para cada posición  $x$  lo cual significa que el valor instantáneo para un instante  $t$  dado variará con la posición. Si suponemos que la señal de desplazamiento aplicada a la ventana oval es

$$y_{VO}(t) = Y_{VO} \text{sen}(2\pi f_0 t), \quad (41)$$

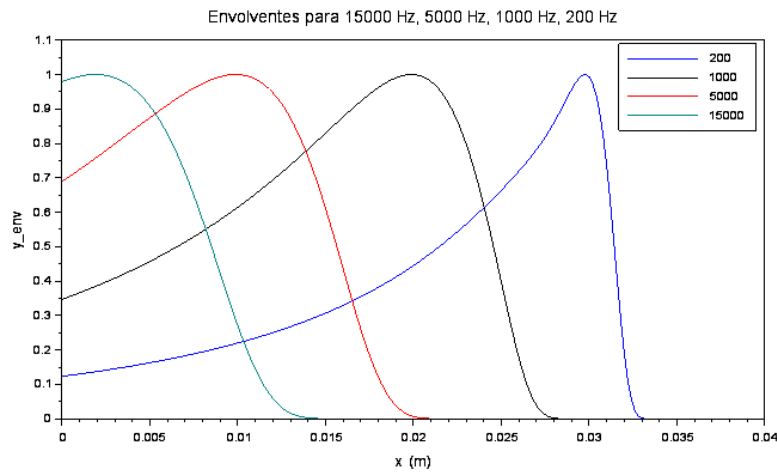
entonces la oscilación de la membrana basilar en una posición  $x$  será

$$y_{MB}(t) = |H(f_0, x)| Y_{VO} \text{sen}(2\pi f_0 t + \Phi_{H(f_0, x)}), \quad (42)$$

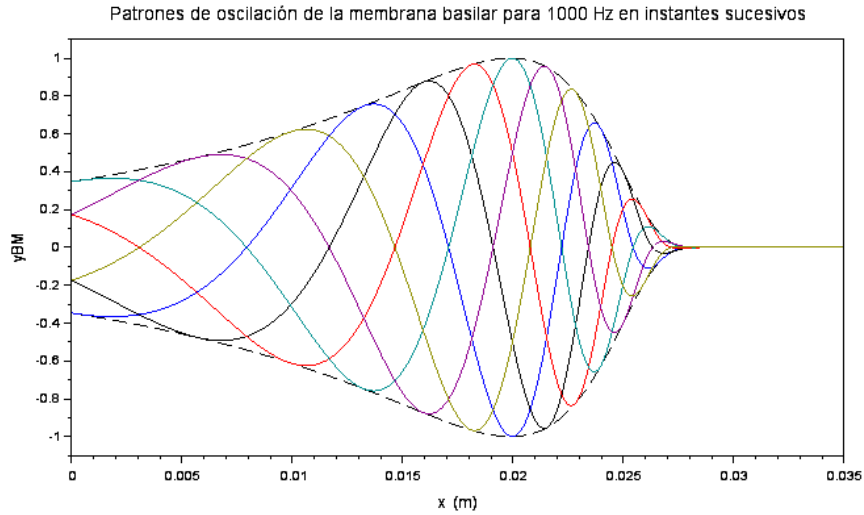
donde

$$\Phi_{H(f_0, x)} = \arg(H(f_0, x)). \quad (43)$$

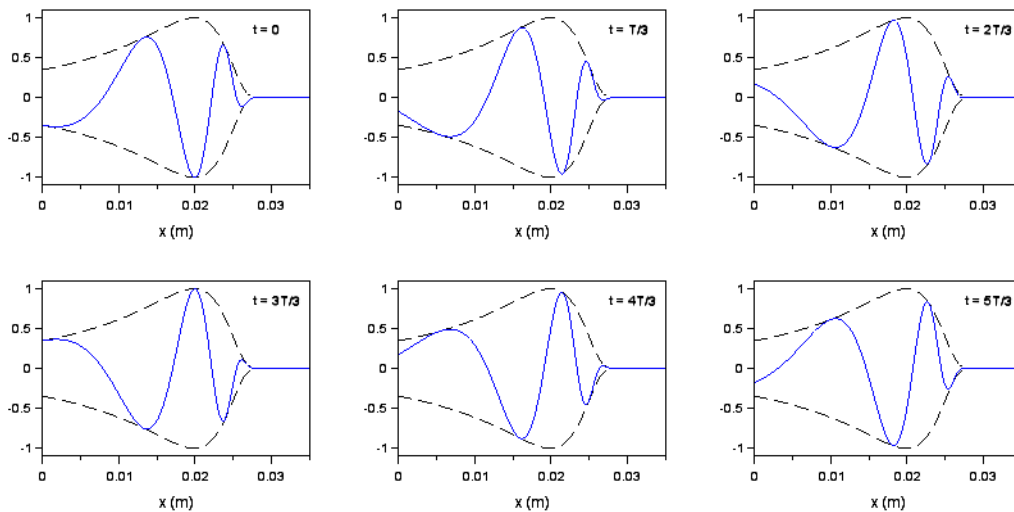
Esto permite obtener el desplazamiento de toda la cóclea para cada instante de tiempo, o la oscilación de un punto de la cóclea a lo largo del tiempo. En la figura 4 se muestra la onda viajera en la membrana basilar en varios instantes de tiempo. En la figura 5 se han separado las ondas correspondientes a instantes diferentes en una secuencia de gráficos.



**Figura 3.** Envolventes de la oscilación de la membrana basilar para las frecuencias 200 Hz, 1000 Hz, 5000 Hz y 10 000 Hz según el modelo, normalizadas para que los picos tengan amplitud 1.



**Figura 4.** Ondas viajeras correspondientes a instantes  $kT/3$  ( $k = 0, \dots, 5$ ) para una senoide de 1000 Hz, donde  $T$  es el periodo.

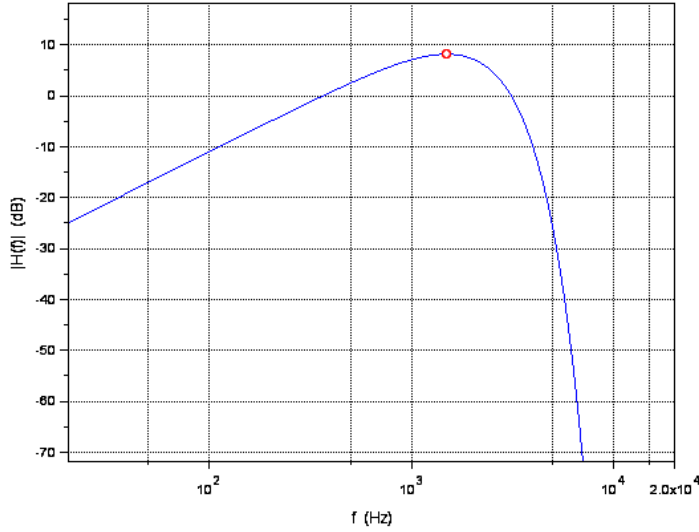


**Figura 5.** Ondas viajeras correspondientes a instantes  $kT/3$  ( $k = 0, \dots, 5$ ) para una senoide de 1000 Hz separadas en diferentes gráficas.

En la figura 6 se muestra la curva de resonancia en la posición de la máxima amplitud de oscilación para 1000 Hz ( $x_0 = 0,019883$  m). Se puede apreciar que la frecuencia de resonancia en  $x_0$  no coincide con 1000 Hz, es decir, la resonancia frecuencial difiere de la resonancia espacial. También se observa que para bajas frecuencias la pendiente crece a razón de 20 dB/déc, pero una vez alcanzada la resonancia la caída es abrupta.

Nos interesa ahora obtener la posición del pico de la envolvente para cada frecuencia, para lo cual deberemos obtener el valor de  $x$  que hace máximo  $|H|$ . Para simplificar el cálculo tengamos en cuenta que el máximo se puede calcular respecto a la variable  $w$  introducida en (23), ya que después bastará despejar  $x$  de (23). Reescribamos, entonces, (8) y (40) del siguiente modo:





**Figura 6.** Curva de resonancia en la posición en la que el pico de oscilación corresponde a 1000 Hz (0,0198830 m). Se observa que la resonancia ocurre en 1483 Hz.

$$Z_x(f, x) = A_1 f^2 e^{-a_1 w} + j A_2 f e^{-a_2 w} \quad (44)$$

$$u = U_0 e^{-\left(B_1 f^2 e^{-a_1 w} (1 + a_1 w) + j B_2 f e^{-a_2 w} (1 + a_2 w)\right)} + K_1 f^2 + j K_2 f \quad (45)$$

Podemos simplificar más aún teniendo en cuenta que el máximo de  $|H|$  se da en la misma ubicación que el de  $|H|^2$  (que tiene una forma analítica más simple), y que las constantes multiplicativas tampoco alteran la posición del máximo. Además, la parte imaginaria del exponente de (45) no afecta. Por lo tanto el problema queda planteado con la ecuación

$$\frac{d}{d w} \left( \left( A_1^2 f^4 e^{-2a_1 w} + A_2^2 f^2 e^{-2a_2 w} \right) e^{-2B_1 f^2 e^{-a_1 w} (1 + a_1 w)} \right) = 0 \quad (46)$$

Luego de algunos cálculos resulta

$$P(w) f^4 + Q(w) f^2 + R(w) = 0 \quad (47)$$

donde

$$\begin{aligned} P(w) &= a_1^2 A_1^2 B_1 w e^{2(a_1 - a_2) w} \\ Q(w) &= a_1 A_1^2 e^{(a_1 - 2a_2) w} + a_1^2 A_2^2 B_1 w \\ R(w) &= a_2 A_2^2 e^{-a_1 w} \end{aligned} \quad (48)$$

Lamentablemente es imposible obtener una solución explícita de  $w$  en función de  $f$  por ser la ecuación trascendente en  $w$ , pero es sencillo obtener  $f$  en función de  $w$ , de donde podría obtenerse  $w$  por inversión numérica. La solución de  $f$  se obtiene mediante

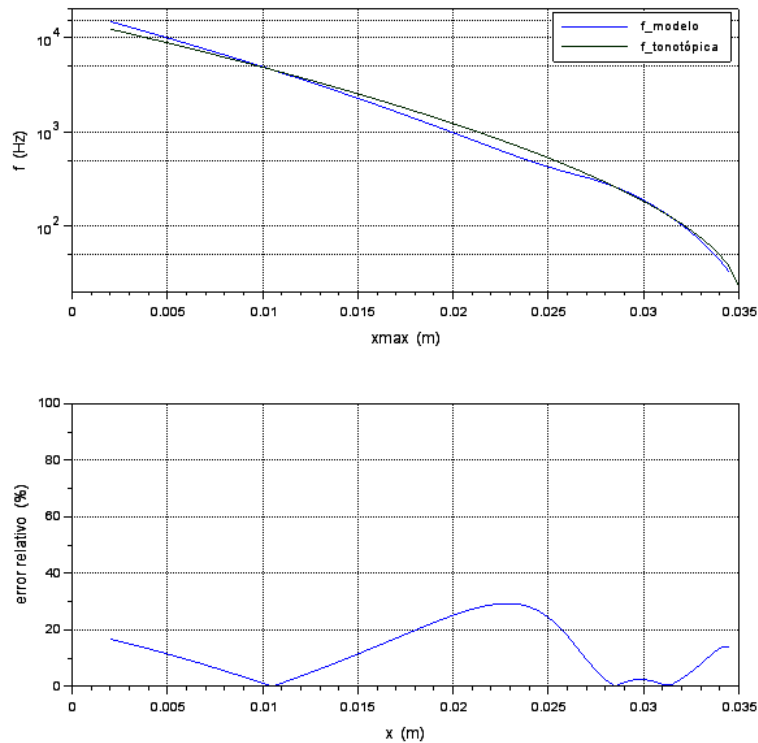
$$f(w) = \sqrt{\frac{-Q(w) + \sqrt{Q^2(w) - 4P(w)R(w)}}{2P(w)}} \quad (49)$$

En la figura 7 se muestra la curva tonotópica (frecuencia vs. posición de máxima oscilación) según Deutsch et al. (1993) juntamente con la correspondiente a esta ecuación. El error máximo es de un 29 % y se produce a 583 Hz.

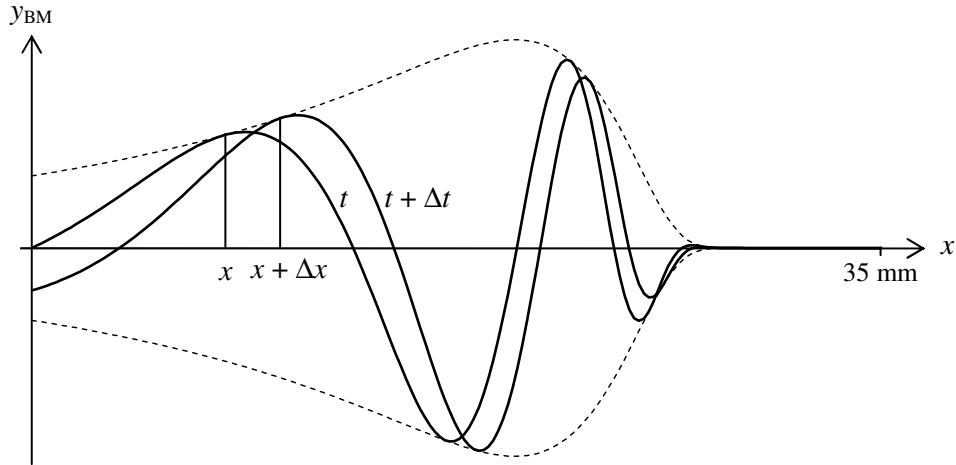
### Velocidad de propagación de la onda viajera

Resulta interesante analizar la velocidad de propagación de la onda viajera a lo largo de la cóclea. Para ello tengamos en cuenta que lo que caracteriza la propagación de la onda es que tomamos un rasgo característico de la onda, por ejemplo un cruce por cero o un pico, y observamos su desplazamiento en función del tiempo (ver figuras 4 y 5). Lo que caracteriza a esos rasgos es que su fase es constante, por ejemplo un cero con pendiente creciente corresponde a fase 0, un pico positivo,  $\pi/2$ , etc. Por esta razón se denomina *velocidad de fase* a esta velocidad. Con referencia a la figura 8, buscamos obtener el desplazamiento  $\Delta x$  correspondiente a un incremento de tiempo  $\Delta t$ . De acuerdo con la ecuación (42), para que la fase de la onda senoidal sea la misma en ambos instantes debe ser

$$2\pi f_o(t + \Delta t) + \Phi_H(f_o, x + \Delta x) = 2\pi f_o t + \Phi_H(f_o, x), \quad (50)$$



**Figura 7.** Comparación entre la curva tonotópica real y la obtenida con el modelo estudiado. En la gráfica de abajo se muestra el error relativo porcentual.



**Figura 8.** Disposición para el cálculo de la velocidad de fase de la onda viajera. Se muestra la forma espacial de la membrana basilar para dos instantes  $t$  y  $t + \Delta t$  y dos puntos correspondientes a la misma fase  $\pi/2$ . Nota: por estar la onda viajera afectada por una envolvente la fase  $\pi/2$  no se da en el máximo sino en el punto de tangencia.

de donde

$$\Phi_H(f_0, x + \Delta x) - \Phi_H(f_0, x) = -2\pi f_0 \Delta t. \quad (51)$$

El incremento de la fase puede aproximarse utilizando su derivada

$$\frac{d\Phi_H(f_0, x)}{dx} \Delta x \cong -2\pi f_0 \Delta t, \quad (52)$$

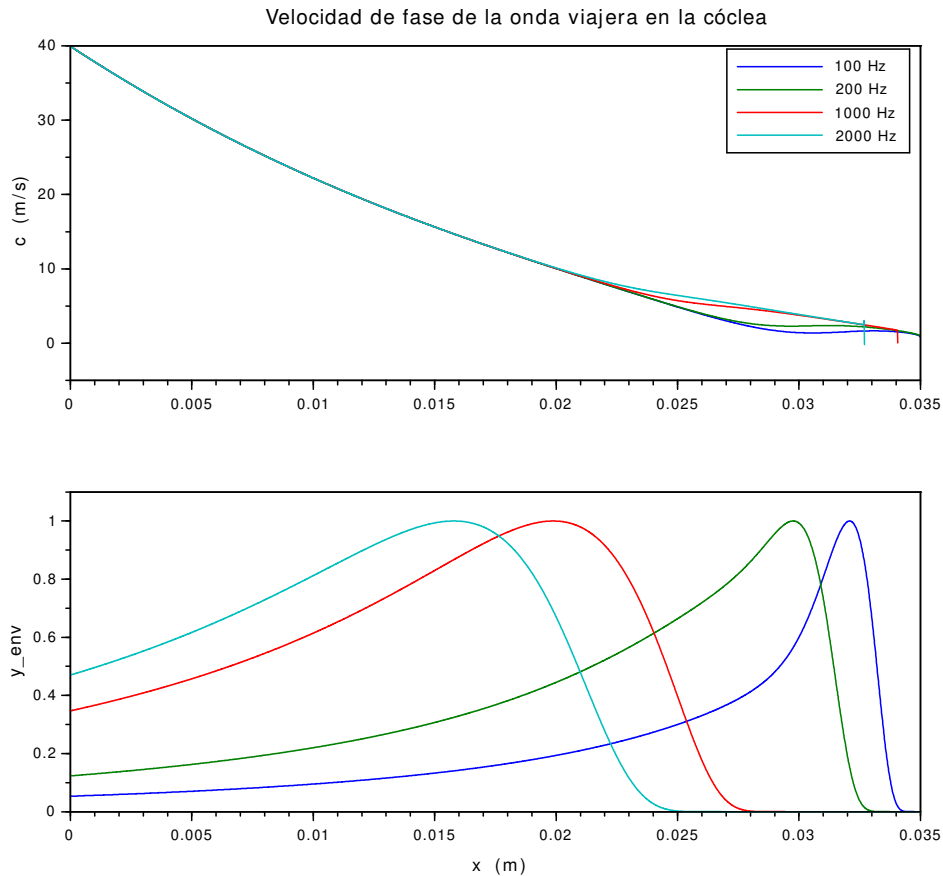
de donde

$$\frac{\Delta x}{\Delta t} \cong -\frac{2\pi f_0}{\frac{d\Phi_H(f_0, x)}{dx}}. \quad (53)$$

Pasando al límite obtenemos la velocidad  $c$  de propagación de los puntos de fase constante de la onda progresiva

$$c = -\frac{2\pi f_0}{\frac{d\Phi_H(f_0, x)}{dx}}. \quad (54)$$

En la figura 9 se muestran las velocidades de fase en función de la distancia a la ventana oval para cuatro frecuencias. Se observa que entre  $x = 0$  hasta  $x = 0,02$  m la velocidad es constante, por lo que la membrana basilar se comporta como un medio no dispersivo. A partir de  $x = 0,02$  m se presenta una ligera anomalía, más marcada para las



**Figura 9.** Velocidad de fase en función de la distancia. Se observa que hasta aproximadamente 0,02 m de la ventana oval la velocidad es independiente de la frecuencia, mientras que para distancias mayores se produce una ligera dependencia de la frecuencia. Para comparación se muestra, en la figura de abajo, las envolventes para cada frecuencia

altas frecuencias. Ello sucede, no obstante, en una zona de la cóclea donde la respectiva respuesta es prácticamente nula, como se aprecia en la gráfica de las envolventes.

### Tonos múltiples

En los ejemplos de las figuras 4, 5 y 7 había un único tono. Es posible obtener las ondas viajeras sumando las ondas viajeras. Cuando intervienen tonos múltiples las envolventes se superponen de una forma poco clara, y si bien hay una envolvente general, la misma tiene poca relación con las envolventes correspondientes a las frecuencias presentes. Ello se debe a que el modelo eléctrico es completamente pasivo, por lo que la discriminación frecuencial resulta demasiado gruesa. En la cóclea real se produce una realimentación local a través del efecto contráctil de las células ciliadas externas ante estímulos neuroeléctricos eferentes (que vienen desde el cerebro). Esta realimentación está sintonizada por lo que se refuerza la resonancia.

Ello puede simularse agregando al modelo una no linealidad expansiva a las envolventes, es decir que mantenga los niveles altos cercanos al pico y en cambio reduzca mucho los niveles menores que se tienen hacia ambos lados del pico. Una función muy simple que logra este objetivo es una potencia de exponente alto, por ejemplo 8 o 10, de modo que la ecuación (42) se reemplaza por

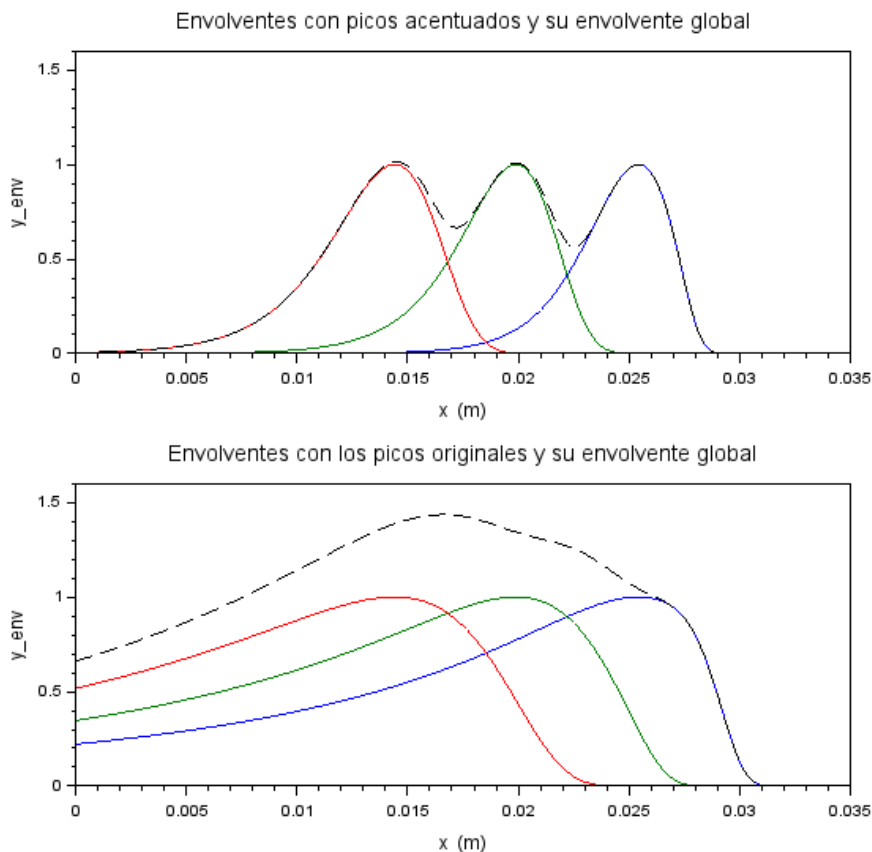
$$y_{MB}(t) = |H(f_o, x)|^m Y_{VO} \text{sen}(2\pi f_o t + \Phi_{H(f_o, x)}), \quad (55)$$

donde se agregó el exponente  $m$ . Si además se utiliza el criterio de superposición energética podemos obtener una envolvente global mediante

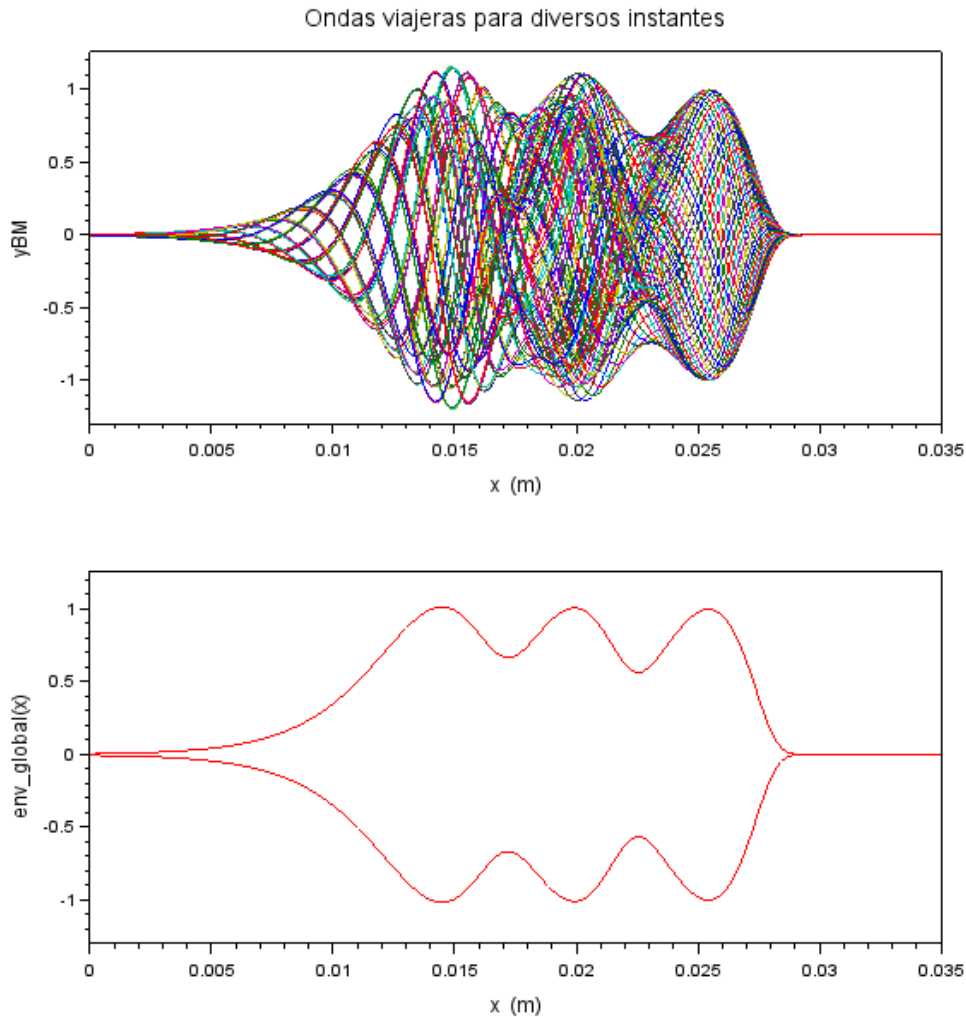
$$\text{env}(x) = \sqrt{\sum_{k=1}^n \left( |H(f_k, x)|^m Y_{VO,k} \right)^2} \quad (56)$$

En la figura 10 se muestran las envolventes con picos acentuados y la envolvente global resultante de su superposición energética para el caso de tres senoides, y se comparan con la envolvente con los picos originales. En el primer caso se observa con mayor claridad la discriminación de las componentes tonales

En la figura 11 se muestran superpuestas las sucesivas posiciones de las ondas viajeras obtenidas superponiendo tres ondas senoidales de igual amplitud para múltiples instantes de tiempo, y la comparación con la envolvente global. Se puede apreciar cómo la tonotopía actúa separando en buena medida las componentes individuales que constituyen el sonido compuesto.



**Figura 10.** Arriba, envolventes correspondientes a tonos puros de 400 Hz, 1000 Hz y 2500 Hz e igual amplitud luego de aplicar un exponente 8 para acentuar la región selectiva, junto con la envolvente global por superposición energética correspondiente en línea de trazos. Abajo, envolventes originales junto con la envolvente resultante en línea de trazos.



**Figura 11.** Arriba, onda viajera compuesta por tres tonos puros de 400 Hz, 1000 Hz y 2500 Hz de igual amplitud. En cada pico predomina una frecuencia. Abajo, envolventes superior e inferior obtenidas por superposición energética de las envolventes correspondientes a las frecuencias individuales.

En los apéndices 1 y 2 se incluyen las funciones de Scilab que permiten implementar la función de transferencia del modelo de la cóclea y el cálculo de las ondas viajeras en la cóclea, incluyendo la posibilidad del exponente de expansión para refinar los picos tonotópicos. En el apéndice 3 se incluye un script de Scilab que permite calcular los ejemplos desarrollados. En los apéndices 4 y 5 se incluye una función y un script para mostrar una versión animada de la onda viajera.

## Referencias

- Deutsch, Sid; Micheli-Tzanakou, Evangelia (1987) *Neuroelectric Systems*. New York University Press. New York
- Deutsch, Sid; Deutsch, Alice (1994) *Understanding the nervous system. An engineering perspective*. IEEE Press. New York



## Apéndice 1

### Función cochlear\_model.sci

```
function [H, f]=cochlea_model(x, f)
// Transfer function of a circuit model of the cochlea
//
// Usage:
//     H = cochlea_model(x, f)
// where
//     x: Scalar or vector of distances in m
//         from the oval window
//     f: Scalar or vector of frequencies in Hz
//     H: Transfer function matrix. There is
//         one row per frequency and one column
//         per distance from the oval window
//
// The model consists in a transmission line with
// variable parameters along the basilar membrane. The
// transmission line can be viewed as a mechanical one
// or an electrical one. The model has been proposed
// by Deutsch & M-Tzanakou (see the Reference section).
// The model allows to compute the transfer function
// between the displacement of the oval window at
// the base of the cochlea (which is roughly
// proportional to the sound pressure at the eardrum)
// and a point at a distance x from the oval window
// and a frequency f. Both x and f can be vectors.
//
// NOTES: If f is not provided, the frequencies of
//         the centre of the one third octave bands
//         are used. If x is not provided x is a
//         71-column vector of distances separated
//         by 0.5 mm
//
// References
// Deutsch, Sid; Micheli-Tzanakou, Evangelina (1987)
//     "Neuroelectric Systems". New York University
//     Press. New York
// Deutsch, Sid; Deutsch, Alice (1993) "Understanding
//     the nervous system. An engineering perspective".
//     IEEE Press. New York
// Miyara, Federico (2024) "Análisis de un modelo
//     circuital de la cóclea"
//
// See also: cochlea_travelling_wave()
//
// -----
// Federico Miyara
// 2024-09-26
//
// Missing arguments
rhs = argn(2);
if rhs<2
    f = 1000;
end
if rhs<1
    x = 0:0.0005:0.035;
end
```



```

// Length of basilar membrane from the oval window
// to the helicotrema, in m
L = 0.035;
// Characteristic resistance of the cochlea in ohm
// or N*s/m
Ro = 0.02;
// Mean width of basilar membrane in m
aBM = 0.0003;
// Mean cross sectional area of vestibular scala in m2
A = 1.5e-6;
if argn(2)<2
    // Vector of distances from the oval window in m
    x = 0:0.0005:L;
else
    // Ensure x is a row vector
    x = x(:)';
end
if argn(2)<1
    // Frequency vector in Hz
    f = 20*10.^(0:0.1:3)';
else
    // Ensure f is a column vector
    f = f(:);
end

// Model current or perilymph velocity amplitude
// in A o m/s
Io = 1

// Auxiliary variable (nonlinear distance
// transformation)
w = sqrt(L - x);

// Impedance per unit length (Miyara, ec. 12,
// Deutsch et al., ec 9.9)
a1 = -100;
a2 = -20;
A1 = 0.39;
A2 = 0.1325;
Zx = A1*f.^2*exp(a1*w) + %i* A2*f*exp(a2*w);

// Integrals of exp(a1*w) and exp(a2*w) respect to x
int_1 = 2/a1.^2 * exp(a1*w).*(1 - a1*w);
int_2 = 2/a2.^2 * exp(a2*w).*(1 - a2*w);

// Integral of impedance per unit length respect to x
int_Zx = -( A1*f.^2 * int_1 + %i* A2*f * int_2);

// Complex integration constant to adjust the value
// of current to Io at x = 0 (Miyara, ecs. 26 a 28)
K1 = A1^2/a1^2/Ro * exp(a1*sqrt(L))*(1 - a1*sqrt(L))
K2 = A2^2/a2^2/Ro * exp(a2*sqrt(L))*(1 - a2*sqrt(L))
K = K1*f.^2 + %i*K2*f;

// Current (or velocity) as a function of x
I = Io*exp(int_Zx/Ro + K*ones(x));

// Transfer function as a function of x
H = -A/aBM * Zx/Ro .* I/Io;

```

endfunction

## Apéndice 2

### cochlea\_travelling\_wave.sce

```
function [y, env]=cochlea_travelling_wave(x, t, f, alpha, e)
// Travelling waves in the cochlea
//
// Computes the shapes of the cochlea at distances
// x from the oval window, at instants t when
// excited by sine waves of frequencies f and
// phases alpha
//
// Usage
// [y, env] = cochlea_travelling_wave(x, t, f, alpha, e)
// where
// x:      Vector of positions along the
//         cochlea in m
// t:      Vector of time instants in s
// f:      Vector of frequencies in Hz
// alpha:  Vector of phases, in radians. Must
//         have the same length as f
// e:      Resonance enhancement exponent
// y:      Array of dimensions length(x),
//         length(t), length(f) containing
//         the displacement of the cochles
//         from the equilibrium at the
//         corresponding values of x and t
//         for the corresponding f, in
//         arbitrary units.
// env:    Array of dimensions length(f),
//         length(x) that contains the
//         envelopes for each frequency
//
// The algorithm uses a transmission line electric
// circuit model given in the references. In
// addition, resonant envelopes may be enhanced
// (i. ., improve selectivity) by applying an
// exponent to the envelopes normalized by their
// peak value, so that the peak ei kept but the
// neighbouring values decrease much faster than
// their original slopes.
//
// References
// Deutsch, Sid; Micheli-Tzanakou, Evangelina (1987)
// "Neuroelectric Systems". New York University
// Press. New York
// Deutsch, Sid; Deutsch, Alice (1993) "Understanding
// the nervous system. An engineering perspective".
// IEEE Press. New York
// Miyara, Federico (2024) "Análisis de un modelo
// circuital de la cóclea"
//
// See also: cochlea_model()
//
// -----
// Author: Federico Miyara
// Date: 2024-10-94
```

```

// Missing arguments
rhs = argn(2);
if rhs<2
    t = 0;
end
if rhs<3
    f = 1000;
end
if rhs<4
    alfa = zeros(f);
end
if rhs<5
    e = 1;
end

if 1==2 // testing values --DELETE--
    // Distances to the oval window
    x = 0:0.00001:0.035;
    // Frequencies
    f = [400 1000 2500];
    // Phases
    alpha = [0 0 0];
    // Times
    t = 0:0.00005:0.005;
    // Resonance enhancement exponent
    e = 1;
end

// Transfer function of the cochlea
H = cochlea_model(x,f);
// Normalized envelopes (respect to the
// maximum of each envelope)
env = abs(H) ./ (max(abs(H), "c") * ones(1, size(H,2)));
// Transfer function phase in radians
phi = phasemag(H, "c") * %pi/180;

// Initialize y
y = zeros(length(t), length(x), length(f));
for i=1:length(f)
    // Phases due to the transfer function and
    // the position on the cochlea, replicated
    // once per time instant
    phis = repmat(phi(i,:), length(t), 1);
    // Phases due to the oval window input sine
    // waves (includes initial phase and time
    // dependence for current frequency)
    psi = 2 * %pi * f(i) * t(:) + alpha(i);
    // Phases replicated once per position
    psis = repmat(psi, 1, length(phi(i,:)));
    // Envelope for current frequency replicated
    // once per time instant and, if required,
    // enhanced through the application of
    // a resonance enhancement exponent e
    envs = repmat(env(i,:) .^ e, length(t), 1);
    // Traveling waves for current frequency
    // Create a matrix with as many rows as
    // time instants and as many columns as
    // positions along the cochlea
    y(:, :, i) = y(:, :, i) + envs .* sin(phis + psis);
end
endfunction

```

## Apéndice 3

### Script cochlea\_model\_prueba.sce

```
// Análisis de un modelo circuital de la cóclea
//
// El modelo consiste en una línea de transmisión con
// parámetros variables con la distancia a la ventana
// oval
//
// Referencia
// Deutsch, Sid; Micheli-Tzanakou, Evangelina (1987)
// "Neuroelectric Systems". New York University
// Press. New York
// Deutsch, Sid; Deutsch, Alice (1993) "Understanding
// the nervous system. An engineering perspective".
// IEEE Press. New York
// Miyara, Federico (2024) Análisis de un modelo circuital
// de la cóclea
//
// -----
// Federico Miyara
// 2024-09-26
// 2024-09-30

clear
close(winsid());

// Carga de la función cochlea_model()
exec cochlea_model.sci;

// Carga de la función cochlea_travelling_wave()
exec cochlea_travelling_wave.sci;

// -----
// 1 ENVOLVENTES
// -----

// Vector de distancias a la ventana oval
x = 0:0.0002:0.035;
caso = 2
select caso
case 1
    // Vector de frecuencias utilizadas en Deutsch
    // et al. (1994)
    f = [170 1160 4700 15000]';
case 2
    // Otro vector de frecuencias
    f = [200 1000 5000 15000]';
end

// Obtener las respuestas espaciales para
// cada frecuencia
[H, f] = cochlea_model(x, f);

// Obtener los máximos de cada envolvente
// y los índices donde éstos se dan
[Hmax, imax] = max(abs(H), "c")
```

```

// Posiciones tonotópicas de los máximos
xmax = x(imax)

// Normalización de las envolventes respecto
// a sus respectivos máximos
Hnor = H ./ repmat(Hmax,1,length(x));

// Gráfica de las envolventes
scf(1);
clf(1);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
gcf().figure_size = [800,550];
// Título contextualizado
tit1 = "Envolventes para ";
for i=length(f):-1:2
    tit1 = tit1 + string(f(i)) + " Hz, ";
end
tit1 = tit1 + string(f(1)) + " Hz";
title(tit1)
plot(x, abs(Hnor))
legend(string(f'));
xlabel("x (m)")
ylabel("y_env")
gca().tight_limits = ["on","on","off"];
gca().data_bounds = [0, 0; 0.040, 1.1];

// -----
// 2 ONDA VIAJERA
// -----

// Vector de distancias a la ventana oval
x = 0:0.00001:0.035;

// Frecuencia
f = 1000;

// Fase
alpha = 0;

// Vector de tiempos
T = 1/f;
t = 0:T/6:5/6*T;

// Obtención de la onda viajera en varios
// instantes sucesivos y la envolvente
[y, env] = cochlea_travelling_wave(x, t, f, alpha);

// Gráfica de la deformación de la membrana
// basilar a intervalos de 1/6 de ciclo
scf(2);
clf(2);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
gcf().figure_size = [800,550];
plot(x, env, "k--", x, -env, "k--", x, y)
tit2 = "Patrones de oscilación de la membrana basilar para ";
tit2 = tit2 + string(f) + " Hz en instantes sucesivos";
title(tit2)
xlabel("x (m)")

```

```

ylabel("yBM")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, -1.1; 0.035, 1.1];

// Serie de gráficas conteniendo una curva por
// cada gráfica
scf(3);
clf(3);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15 ];
gcf().figure_size = [850,550]
for i=1:6
    subplot(2,3,i)
    plot(x, env, "k--", x, -env, "k--", x, y(i,:))
    if i==1
        lbl = "t = 0";
    elseif i==2
        lbl = "t = T/3";
    else
        lbl = "t = " + string(i-1) + "T/3";
    end
    xstring(0.027, 0.7, lbl, 1);
    xlabel("x (m)")
    gca().data_bounds = [0, -1.1; 0.035, 1.1];
    gca().tight_limits = ["on", "on", "on"];
end

// -----
// 3 VELOCIDAD DE LA ONDA VIAJERA
// -----

// Vector de distancias a la ventana oval
deltax = 0.0001;
x = 0:deltax:0.035;

// Frecuencias
f = [100, 200, 1000, 2000];

// Fases
alfa = ones(f)*%pi/4;

// Dos tiempos que difieren en un pequeño incremento
deltat = 0.0001;
t = [0, deltat]

// Obtención de la onda viajera en dos instantes
// próximos para dos frecuencias diferentes
[y, env] = cochlea_travelling_wave(x, t, f, alfa);

// Figura a exportar para ilustrar el concepto
// de velocidad de fase
scf(4);
clf(4);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15 ];
gcf().figure_size = [800,550];
plot(x, env(3,:), "k--", x, -env(3,:), "k--", x, y(:, :, 3))
tit2 = "Patrones de oscilación de la membrana basilar para ";
tit2 = tit2 + string(f(3)) + " Hz en instantes sucesivos";
title(tit2)
xlabel("x (m)")

```

```

ylabel("yBM")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, -1.1; 0.035, 1.1];

// Se toman ahora gran cantidad de puntos
// en x para tener buena precisión en la
// derivada
deltax = 0.00001;
x = 0:deltax:0.035;

// Función de transferencia de la cóclea
// para las frecuencias requeridas
H = cochlea_model(x, f);

// Envolventes
env = abs(H) ./ (max(abs(H), "c") * ones(1, size(H, 2)));

// Fase de la función de transferencia en
// radianes
phi = phase(H, "c") * pi / 180;

// Cálculo de la velocidad de fase de la onda
// viajera. Aplicaremos la fórmula
//  $c = -2 * \pi * f / (dfase/dx)$ 
c = -2 * pi * repmat(f(:), 1, length(x)-1) ./ (diff(phi, 1, "c") / deltax);
xc = x(1:length(x)-1);

scf(5);
clf(5);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
gcf().figure_size = [650, 550];
subplot(2, 1, 1)
plot(xc, c)
title("Velocidad de fase de la onda viajera en la cóclea")
legend(string(f) + " Hz");
//xlabel("x (m)")
ylabel("c (m/s)")
subplot(2, 1, 2)
plot(x, env)
xlabel("x (m)")
ylabel("y_env")
gca().tight_limits = ["on", "on", "on"];
gca().data_bounds = [0, 0; 0.035, 1.1];

// -----
// 4 MÁXIMOS
// -----

// Longitud de la membrana basilar
L = 0.035
// Resistencia característica de la cóclea
Ro = 0.02
// Coeficiente de los exponentes de w
a1 = -100
a2 = -20
// Coeficientes de las partes real e imaginaria
// de la impedancia por unidad de longitud
A1 = 0.39
A2 = 0.1325
// Coeficientes de las partes real e imaginaria

```

```

// del exponente de la corriente del modelo
B1 = A1/Ro*2/a1^2
B2 = A2/Ro*2/a2^2

// Posición
xx = 0.002:0.0005:0.035;

// Transformación no lineal de la posición
w = sqrt(L - xx);

// Coeficientes de la ecuación cuadrática de
// la frecuencia al cuadrado en función de la
// transformación no lineal de la posición
P = A1^2*2*B1*a1^2*exp(2*(a1-a2)*w).*exp(a1*w).*w;
Q = 2*a1*A1^2*exp(2*(a1-a2)*w) + A2^2*2*B1*a1^2*exp(a1*w).*w;
R = 2*a2*A2^2;

// Resolución de la ecuación cuadrática
f = sqrt((-Q + sqrt(Q.^2 - 4*P*R))/2./P);

// Curva tonotópica obtenida por mí a partir
// de la curva de Deutsch et al. (1993)
ff = 10.^(1.3012919 + 0.3863057*(35.034435 - xx*1000).^0.5652075);

// Error
er = abs(f - ff);
// Eliminación del error infinito para x = L
er_rel = er(1:$-1)./f(1:$-1);
// Error relativo máximo
[er_rel_max, i] = max(er_rel);
f_er_rel_max = f(i);
disp(f_er_rel_max, er_rel_max)

// Gráfica en función de la frecuencia
scf(6);
clf(6);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
gcf().figure_size = [650,550];
subplot(2,1,1)
plot("nl", xx, f, xx, ff)
title("Frecuencia para la cual el máximo se da en x")
legend("f_tonotópica", "f_modelo")
xlabel("xmax (m)");
ylabel("f (Hz)");
gca().grid = [1 1];
gca().data_bounds = [0, 20; 0.035, 20000];
gca().tight_limits = ["on", "on", "on"];
subplot(2,1,2)
plot(xx(1:$-1), 100*er_rel)
title("Error entre el modelo y la curva tonotópica real")
gca().grid = [1 1];
xlabel("x (m)");
ylabel("error relativo (%)");
gca().data_bounds = [0, 0; 0.035, 100];
gca().tight_limits = ["on", "on", "on"];

```



```

// -----
// 5 RESONANCIA EN UNA POSICIÓN ESPECÍFICA
// -----

// Aproximación de la posición del máximo para
// fo = 1000
fo = 1000;
[delta, i] = min(abs(f - fo));
f(i-1:i+1)
xx(i-1:i+1)
xo = interp1(f(i-1:i+1), xx(i-1:i+1), 1000, "spline")

fff = 20*10.^(0:0.01:3);
H = cochlea_model(xo, fff);
HdB = 20*log10(abs(H));
[Hmax, i] = max(HdB)
disp(fff(i), Hmax)

scf(7);
clf(7);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
plot("ln", fff, HdB, fff(i), Hmax, "or")
title("Resonancia en la posición de máxima oscilación para " + ..
      string(fo) + " Hz")
xlabel("f (Hz)");
ylabel("|H(f)| (dB)");
gca().grid = [1 1];
gca().data_bounds = [20, max(HdB)-80; 20000, max(HdB)+10];
gca().tight_limits = ["on", "on", "on"];

// -----
// 6 MÚLTIPLES TONOS
// -----

// Aquí combinaremos múltiples tonos. Supondremos
// que las resonancias respecto a la posición se
// encuentran reforzadas a través de un exponente e
e = 8;

// Vector de distancias a la ventana oval
x = 0:0.00001:0.035;

// Frecuencias
f = [400 1000 2500];

// Fases
alpha = zeros(f);

// Obtención de la respuesta espacial compleja
HH = cochlea_model(x, f);

// Envolventes normalizadas
env = abs(HH) ./ (max(abs(HH), "c") * ones(1, size(HH, 2)));

// Envolvente global original
env_global = sqrt(sum(env.^2, "r"));

```

```

// Aplicación de una transformación no lineal
// que emula una acentuación del pico de
// resonancia
env1 = env.^8;

// Envolvente global transformada
env_global1 = sqrt(sum(env1.^2, "r"));

scf(8);
clf(8);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15 ];
subplot(2,1,1)
plot(x, abs(env1), x, env_global1, "k--")
title("Envolventes con picos acentuados y su envolvente global")
xlabel("x (m)")
ylabel("y_env")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, 0; 0.035, 1.6];
subplot(2,1,2)
plot(x, abs(env), x, env_global, "k--")
title("Envolventes con los picos originales y su envolvente global")
xlabel("x (m)")
ylabel("y_env")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, 0; 0.035, 1.6];

// Fase en radianes
phi = phasemag(HH, "c") * %pi / 180;

// Vector de tiempos
t = 0:0.00005:0.005;

// Obtener las ondas viajeras para todos los
// instantes y todas las frecuencias
y = cochlea_travelling_wave(x, t, f, alpha, e);

// Sumar las ondas de las diversas frecuencias
yy = sum(y, 3);

// Graficar las ondas viajeras para todos los
// instantes de tiempo en que se las calculó
mini = min(yy) - 0.1;
maxi = max(yy) + 0.1;
scf(9);
clf(9);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
subplot(2,1,1)
plot(x, yy)
title("Ondas viajeras para diversos instantes")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, mini; 0.035, maxi];
xlabel("x (m)")
ylabel("yBM")
subplot(2,1,2)
plot(x, env_global1, "r", x, -env_global1, "r")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, mini; 0.035, maxi];
xlabel("x (m)")
ylabel("env_global(x)")

```

```
scf(10);
clf(10);
fig = gcf().figure_id;
gcf().figure_position = [180 + fig*20, 160 - fig*15];
plot(x, yy(1,:))
plot(x, env_global1, "r--", x, -env_global1, "r--")
gca().tight_limits = ["on", "on", "off"];
gca().data_bounds = [0, -3; 0.035, 3];
```

## Apéndice 4

### Función `animate.sci`

```
// Scilab ( https://www.scilab.org/ ) - This file is part of Scilab
// Copyright (C) INRIA
// Copyright (C) 2012 - 2024 - Scilab Enterprises
//
// This file is hereby licensed under the terms of the GNU GPL v2.0,
// pursuant to article 5.3.4 of the CeCILL v.2.1.
// This file was originally licensed under the terms of the CeCILL
// v2.1,
// and continues to be available under such terms.
// For more information, see the COPYING file which you should have
// received along with this program.
//
// Improvements: Copyright Federico Miyara, 2024-10-06

function y=animate(f, x, theta, flag, rect, fps, del)
// Animated plot of x-->f(x,t) for t=theta(1),theta(2),etc
//
// Usage:
//     animate(f,x,theta,flag,rect,fps,del)
//     y = animate(f,x,theta,flag,rect,fps,del)
// where
//     f:      function y=f(x,t). f is a Scilab function or
//            a dynamically linked routine (referred to as
//            a string). It may also be a previously computed
//            NxM matrix whose elements are f(x(i),t(k)),
//            i=i, ...,N, k=1, ..., M
//     x:      N-vector of x-values
//     theta:  M-vector of parameters
//     flag:   Optional parameter. If flag = 'yes', screen
//            is cleared between consecutive plots). If
//            flag = 'no' then screen not cleared between
//            consecutive plots).
//     rect:   [xmin, xmax, ymin, ymax] (1 x 4 real vector),
//            containing a priori lower and upper bounds
//            for x and f(t,x).
//     fps:   Frames per second. May be a single number,
//            a two-component vector, indicating the
//            initial and ending values of frames per
//            second, hence allowing to accelerate or
//            decelerate (slow motion <---> fast motion)
//            or a vector with more than two components
//            to control the frame rate more accurately
//            during playback
//     del:   Optional argument indicating whether the
//            previous content of the figure will ("yes")
//            or will not ("no") be erased
//     y:     Optional output argument providing the whole
//            set of y curves as a matrix, each column of
//            which contains the y values corresponding to
//            x values for one value of the parameter. This
//            may be used as the first argument in subsequent
//            playback repetitions, which is useful when
//            the calculation of y values takes too long.
//
// This function is based on the original paramfplot2d()
```

```

// function, with the following Improvements:
//
// 1) The name animate() is much more self-evident than the
//    original one.
// 2) The whole set of values of the function for all x and
//    all theta is computed in advance and saved into a
//    matrix. Even if this requires more memory (in the
//    original realization only a vector was created at a
//    single time) this allows time-consuming computations
//    to be kept outside of the playback loop, preventing
//    artifacts. The memory requirement should not be an
//    issue even for quite long animations.
// 3) Along the same line, the function argument can now be
//    replaced by the whole data set, where each column
//    contains the curve for one value of the parameter.
//    This is important since in certain cases in which
//    the data computation could take quite a long time,
//    the original method would require a long preparation
//    time each time the animation is run.
// 4) It is possible to select, through a new argument, the
//    number of frames per second (fps).
// 5) It is even possible to provide a transition from a
//    starting fps and an ending fps, allowing thus to
//    accelerate or decelerate playback.
// 6) The new version is fully commented to facilitate
//    interested users to understand how it works.
// 7) An optional output argument has been included, which
//    offers the whole set of y values corresponding to
//    all x and all t, in case the user may find them
//    useful for further processing.
// 8) Since the order and meaning of the first arguments
//    have not changed, it is a fully-compatible replacement
//    for the original paramfplot2d().
// 9) A new argument makes it optional to delete the content
//    of the previous figure, allowing complementary material
//    to be included.
// 10) The style has been polished removing unnecessary lines.
//
// Example:
//
//    function y = f(x,t)
//        y = abs(cos(1.5*x+4*t)).*sin(x+10*t),
//    endfunction
//    x = linspace(0, 20*pi, 500);
//    theta = 0:0.05:5;
//    animate(x, theta, f, "no", [0, -1, 500, 1], [10 30])

rhs = argn(2);

// If no argument is given, run a demo
if ~rhs
    deff("y=f(x,t)", "y=t*sin(x)")
    x = linspace(0,2*pi,50);
    theta = 0:0.05:1;
    clf;
    animate(f,x,theta);
    return;
end
// Error message
msg = "%s: Wrong number of input argument(s): At least %d
expected.\n"

```

```

if rhs<3
    error(msprintf(gettext(msg), "paramfplot2d", 3));
end

// Ensure x is a column vector
x = x(:);
// Ensure theta is a row vector
theta = theta(:).';

msg = "Error: First argument must be a function or a " + ..
      "length(theta) x length(x) matrix";
// Compute the whole set of curves
if type(f)==13
    // If f is a Scilab function, compute the whole
    // data set as an N x M vector
    y = feval(x, theta, f);
elseif type(f)==1
    // If f is a data matrix check whether it has
    // the right size
    if size(f)==[length(x), length(theta)]
        // f has correct size
        y = f;
    elseif size(f)==[length(theta), length(x)]
        // f has reversed size, transpose it
        y = f.';
    else
        // f has definitely an unrev;covably wrong size
        error(msprintf(msg))
        return
    end
else
    // f type is incompatible
    error(msprintf(msg))
    return
end

// Set default del
if rhs<7
    del = "yes";
end
// Set default fps
if rhs<6
    fps = 10;
end
// Set default rectangle
if rhs<5
    // Compute the data bounds
    xmin = min(x);
    xmax = max(x);
    ymin = min(yy);
    ymax = max(yy);
    rect = [xmin, xmax, ymin, ymax];
end
// Set default flag
if rhs<4
    flag = "no";
end

// If fps is a vector, intepolate linearly to get
// the fps corresponding to each value of the
// parameter

```

```

if length(fps)>1
    m = length(fps);
    n = length(theta);
    fps = interp1(1 + (n-1)/(m-1)*(0:m-1),fps, 1:n);
else
    fps = ones(theta)*fps;
end

// Compute delta times between frames
dt = 1./fps;
// Set realtime unit in s
realtimeinit(min(dt));

// Clear existing figure or create blank one
if del=="yes"
    clf();
end

// Get axes handle
a = gca();
// Convert rectangle format to data bounds
a.data_bounds = matrix(rect,2,2);
// Make axes visible
a.axes_visible = "on";
// Draw polygon
xpoly(x, y(:,1));
// Get the polyline handle
p = gce();

// Initialize timer
realtime(0);
if flag=="no"
    for k=1:size(theta, "*")
        // Update after dt(k) seconds
        realtime(sum(dt(1:k)/min(dt)));
        // Update the ordinate data to the
        // k-th plot
        p.data(:, 2) = y(:,k);
    end
else
    for k=1:size(theta, "*")
        realtime(sum(dt(1:k)/min(dt)));
        // Prepare to plot a new polyline within
        // the same axes
        plot2d(x, y(:,k))
    end
end
endfunction

```

## Apéndice 4

### Script animate\_cochlea.sce

```
// Animación de las ondas viajeras en la cóclea

// Carga de las funciones necesarias
exec cochlea_travelling_wave.sci;
exec animate.sci;

clf();

// Distancias a lo largo de la cóclea en m
x = 0:0.0002:0.035;
// Instantes de tiempo
theta = 0:0.0001:0.01;
// Frecuencias
fo = [200 1000 2500];
// Fases
alpha = [0 0 0];
// Exponente de acentuación del pico tonotópico
// (incremento de la selectividad
e = 8;

// Cálculo de las ondas para cada frecuencia
// Columnas: tiempos
// Filas: distancias
// Páginas; frecuencias
[yy, env] = cochlea_travelling_wave(x, theta, fo, alpha, e);

// Superposición de las ondas de las tres frecuencias
y = sum(yy,3);

// Envolvente global original
env_global = sqrt(sum((env.^e).^2, "r"));

// Grafica de las envolventes
plot(x, env.^e, "c-.", x, env_global, "r", x, -env_global, "r")

// Límites del rectángulo en el que se mostrará la
// animación
rect = [min(x) max(x), min(y) max(y)];

// Bandera que indica si se borran las ondas o
// se mantienen
flag = "no";

// Cuadros por segundo
fps = 10;

// No borrar las gráficas previas de las envolventes
del = "no";

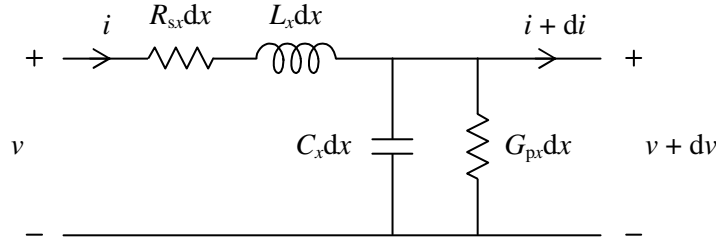
// Animación de la onda viajera compuesta por varias
// frecuencias
animate(y, x, theta, flag, rect, fps, del);
```



## Apéndice 6

### Línea de transmisión eléctrica

Consideremos una línea de transmisión con una inductancia por unidad de longitud  $L_x$ , una resistencia serie por unidad de longitud  $R_{sx}$ , una capacidad por unidad de longitud  $C_x$  y una conductancia paralelo por unidad de longitud  $G_{px}$ . En la figura A4.1 se muestra el modelo de un segmento diferencial de longitud  $dx$ .



**Figura A6.1.** Una sección elemental de una línea de transmisión con pérdidas de longitud  $dx$ .

Podemos plantear las siguientes ecuaciones

$$dv = -\left(R_{sx} dx i + L_x dx \frac{\partial i}{\partial t}\right), \quad (\text{A6.1})$$

$$di = -\left(G_{px} dx v + C_x dx \frac{\partial v}{\partial t}\right). \quad (\text{A6.2})$$

Dividiendo ambos miembros de estas ecuaciones por  $dx$  y pasando al límite cuando  $dx \rightarrow 0$  se llega a

$$\frac{\partial v}{\partial x} = -\left(R_{sx} i + L_x \frac{\partial i}{\partial t}\right), \quad (\text{A6.3})$$

$$\frac{\partial i}{\partial x} = -\left(G_{px} v + C_x \frac{\partial v}{\partial t}\right). \quad (\text{A6.4})$$

Si bien se podría eliminar  $i$  o  $v$  mediante álgebra de derivadas parciales, es más fácil eliminar la dependencia temporal mediante fasores o transformada de Laplace. Por ejemplo, tomando

$$v(x,t) = \dot{V}(x)e^{j\omega t}, \quad (\text{A6.5})$$

resulta

$$\frac{\partial v}{\partial t}(x,t) = \dot{V}(x)j\omega e^{j\omega t}, \quad (\text{A6.6})$$

luego

$$\frac{d\dot{V}}{dx} = -(R_{sx} + j\omega L_x) \dot{I}, \quad (\text{A6.7})$$

$$\frac{d\dot{I}}{dx} = -(G_{px} + j\omega C_x) \dot{V}. \quad (\text{A6.8})$$

de donde

$$\frac{d^2\dot{V}}{dx^2} = (R_{sx} + j\omega L_x)(G_{px} + j\omega C_x) \dot{V} \quad (\text{A6.9})$$

Llamando

$$\Gamma^2 = (R_{sx} + j\omega L_x)(G_{px} + j\omega C_x) \quad (\text{A6.10})$$

Resulta

$$\frac{d^2\dot{V}}{dx^2} = \Gamma^2 \dot{V} \quad (\text{A6.11})$$

Las soluciones son de la forma

$$\dot{V}(x) = A e^{-\Gamma x} + B e^{\Gamma x} \quad (\text{A6.12})$$

Si no hay pérdidas, es decir  $R_{sx} = 0$  y  $G_{px} = 0$ , resulta

$$\Gamma = j\omega \sqrt{L_x C_x} \quad (\text{A6.13})$$

Es fácil ver que

$$c = \frac{1}{\sqrt{L_x C_x}} \quad (\text{A6.14})$$

es la velocidad de propagación de la onda a lo largo de la línea de transmisión ya que

$$V(x,t) = \dot{V}(x) e^{j\omega t} = A e^{j\omega(t-x/c)} + B e^{j\omega(t+x/c)} \quad (\text{A6.14})$$

El primer término es la onda directa, que se mueve en la dirección de las  $x$  positivas y el segundo es la onda reflejada que se mueve en dirección contraria. La existencia o no de la onda reflejada dependerá de la carga aplicada en la terminación de la línea.

Si hay pérdidas resulta

$$\Gamma = \alpha + j\beta \quad (\text{A6.15})$$

$\alpha$  es el coeficiente de atenuación y  $\beta$  es el número de onda.

A partir de  $\dot{V}$  podemos obtener  $\dot{I}$  para lo cual basta integrar respecto a  $x$  la expresión de  $d\dot{I}/dx$ :

$$\begin{aligned}\dot{I} &= \int \frac{d\dot{I}}{dx} dx = -(G_{px} + j\omega C_x) \int \dot{V} dx \\ &= \frac{G_{px} + j\omega C_x}{\Gamma} (A e^{-\Gamma x} - B e^{\Gamma x})\end{aligned}\quad (\text{A6.16})$$

Llamamos *impedancia característica* de la línea a

$$Z_o = \frac{\Gamma}{G_{px} + j\omega C_x} = \sqrt{\frac{R_{sx} + j\omega L_x}{G_{px} + j\omega C_x}}. \quad (\text{A6.17})$$

Si la línea no tiene pérdidas, entonces la impedancia característica es independiente de la frecuencia y vale

$$Z_o = \sqrt{\frac{L_x}{C_x}}. \quad (\text{A6.18})$$

Supondremos ahora la línea de longitud finita  $L$  y cargada en su extremo por una impedancia  $Z_L$ , como se muestra en la figura A6.2. Tendremos las siguientes condiciones de contorno:

$$\dot{V}(0) = \dot{V}_1 \quad (\text{A6.19})$$

$$\dot{V}(L) = \dot{I}(L) Z_L$$

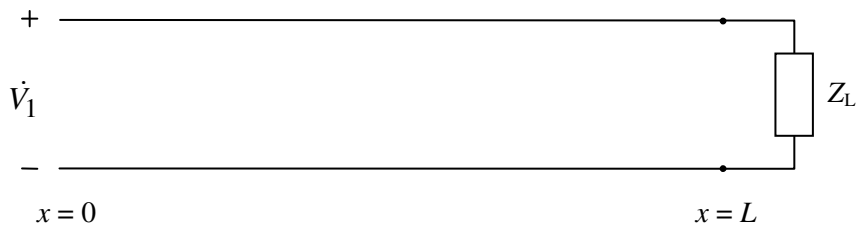
que se traducen en

$$A + B = \dot{V}_1 \quad (\text{A6.20})$$

$$A e^{-\Gamma L} + B e^{\Gamma L} = \left( \frac{A}{Z_o} e^{-\Gamma L} - \frac{B}{Z_o} e^{\Gamma L} \right) Z_L$$

Es un sistema de dos ecuaciones con dos incógnitas cuya solución es

$$\begin{aligned}A &= \frac{\dot{V}_1}{1 + \frac{Z_L - Z_o}{Z_L + Z_o} e^{-2\Gamma L}} \\ B &= \frac{\dot{V}_1}{1 + \frac{Z_L - Z_o}{Z_L + Z_o} e^{-2\Gamma L}} \frac{Z_L - Z_o}{Z_L + Z_o} e^{-2\Gamma L}\end{aligned}\quad (\text{A6.21})$$



**Figura A6.2.** Una línea de transmisión cargada con una impedancia igual a su impedancia característica.

Puede expresarse de un modo más simétrico como

$$\begin{aligned}
 A &= \dot{V}_1 \frac{(Z_L + Z_o) e^{\Gamma L}}{(Z_L + Z_o) e^{\Gamma L} + (Z_L - Z_o) e^{-\Gamma L}} \\
 B &= \dot{V}_1 \frac{(Z_L - Z_c) e^{\Gamma L}}{(Z_L + Z_o) e^{\Gamma L} + (Z_L - Z_o) e^{-\Gamma L}}
 \end{aligned}
 \tag{A6.22}$$

Si  $Z_L = Z_o$  resulta  $A = V_1$  y  $B = 0$ , es decir si la impedancia de carga es igual a la impedancia característica de la línea (que coincide con la impedancia que tendría una línea semiinfinitamente larga con iguales parámetros por unidad de longitud), no existe onda reflejada.