Contenido

1.1 Conversión de archivos de audio tipo .WAV a archivos de datos tipo .PRN

LISTADO DEL PROGRAMA WAV_PRN.PAS ESCRITO EN TURBO PASCAL 5.0

```
{ Para el procesamiento digital de señales de voz  lo primero que se necesita
  es una señal de voz. Con la proliferación de sistemas multimedios para com-
  putadores personales, ahora es fácil capturar señales de voz y almacenarlas
  en archivos tipo .WAV.  Como los documentos de MathCAD  que se presentan en
  este reporte requieren señales de voz en el formato .PRN,  aquí se presenta
  un programa en pascal que convierte archivos  .WAV  MONOFONICOS DE 8 BITS a
  archivos .PRN.                                                            }

uses dos,                   { Biblioteca de funciones para manejo de archivos }
     crt;                   { Biblioteca de funciones para manejo de pantalla }
var
  NombreArchivo : String;   { Nombre del archivo. Cambia la extension WAV-PRN }
  Archivoin     : text;     { Archivo de entrada tipo .WAV                    }
  archivoout    : text;     { Archivo de salida  tipo .PRN                    }
  i,j,n         : integer;  { indice auxiliar, valor de c/muestra, # muestras }
  directorio    : SearchRec;{ Estructura de pascal para leer directorios      }
  c             : char;     { Lectura de cada byte del archivo de entrada     }

begin
  clrscr;
  repeat
    write(#7,'¿Nombre del Archivo? ');          { Acepta un nombre de archivo }
    readln(NombreArchivo);
    NombreArchivo := NombreArchivo + '.wav';    { Añade la extensión .WAV     }
    FindFirst(NombreArchivo,archive,directorio);{ y verifica su existencia.   }
  until DosError = 0;                           { Repite si no es válido      }
  assign(Archivoin,NombreArchivo);
  reset(Archivoin);                  { Abre el archivo de entrada para lectura }
  i:=length(NombreArchivo);          { Cambia la extensión para conformar el   }
  NombreArchivo[i-0]:='n';           { nombre del archivo de salida            }
  NombreArchivo[i-1]:='r';
  NombreArchivo[i-2]:='p';
  assign(ArchivoOut,NombreArchivo);
  rewrite(ArchivoOut);              { Abre el archivo de salida para escritura }
  writeln('Conversión en progreso...');  { Reporta la actividad de conversión }
  for i:=1 to 40 do read(archivoin,c); { Ignora los primeros 40 bytes del     }
  read(archivoin,c);                   { archivo de entrada. Los siguientes 4 }
  n:=ord(c);                           { bytes indican el número de muestras  }
  read(archivoin,c);
  n:=n + abs(256*ord(c));
  read(archivoin,c);
  read(archivoin,c);
  for i:=1 to n do                     { Lee cada una de las muestras         }
  begin
    read(Archivoin,c);
    j:=ord(c)-128;                     { Elimina el nivel dc                  }
    writeln(ArchivoOut,j:4);
  end;
  close(ArchivoOut);                   { Cierra los dos archivos              }
  close(ArchivoIn);
end.
```

1.2 Conversión de archivos de datos tipo .PRN a archivos de audio tipo .WAV

LISTADO DEL PROGRAMA PRN_WAV.PAS ESCRITO EN TURBOPASCAL 5.0

```
uses dos, crt;              { Bibliotecas de funciones para archivos y pantalla }

var
  NombreArchivo         : String;       { Nombre de los archivos, .PRN o .WAV }
  Archivoin, archivoout : text;         { Archivos de entrada y salida        }
  ValorTexto            : string;       { Buffer para lectura de datos        }
  i,j,k,n               : integer;      { Indices Auxiliares                  }
  directorio            : SearchRec;    { Estructura de archivos en pascal    }
  x,max                 : double;       { Valor de cada muestra, valor maximo }
  c,d                   : char;         { Para escritura de c/byte en .WAV    }

begin
  clrscr;
  repeat
    write(#7,'¿Nombre del Archivo? ');  { Acepta un nombre de archivo, le     }
    readln(NombreArchivo);              { añade la extensión .PRN y verifica }
    NombreArchivo:=NombreArchivo+'.prn'; { su existencia.                    }
    FindFirst(NombreArchivo,archive,directorio);
  until DosError=0;                     { Repite si el archivo no existe      }
  assign(Archivoin,NombreArchivo);
  reset(Archivoin);                  { Abre el archivo de entrada para lectura }
  n:=0; max:=-1e38;                  { Numero de muestras, muestra máxima      }
  while not(eof(Archivoin)) do
  begin
    readln(Archivoin,ValorTexto);    { Lee cada muestra, la convierte a double }
    val(ValorTexto,x,i);             { y actualiza los valores de n y de max   }
    if abs(x)>max then max:=abs(x);
    inc(n);
  end;
  close(ArchivoIn);
  i:=length(NombreArchivo);          { Conforma el nombre del archivo de salida }
  NombreArchivo[i-0]:='v';           { añadiendo la extensión .WAV              }
  NombreArchivo[i-1]:='a';
  NombreArchivo[i-2]:='w';
  assign(ArchivoOut,NombreArchivo); { Abre el archivo de salida para escritura }
  rewrite(ArchivoOut);              { y el de entrada para lectura             }
  reset(ArchivoIn);
  writeln('Se detectaron ',n,' muestras en ',NombreArchivo,'.');
  write('¿Frecuencia de muestreo ? ');         { Introduce la tasa de muestreo }
  readln(ValorTexto);                          { y calcula su representación   }
  val(ValorTexto,j,k);                         { en bytes para escribirla en   }
  i:=j mod 256;                                { el archivo de salida          }
  c:=chr(i);                                   { byte menos significativo      }
  i:=j div 256;
  d:=chr(i);                                   { byte más significativo        }
  writeln('Conversión en progreso...');
  write(ArchivoOut,'RIFF');                      { Encabezado constante        }
  write(ArchivoOut,chr((n+36)  MOD 256));        { Longitud del archivo        }
  write(ArchivoOut,chr(((n+36) DIV 256) MOD 256));
  write(ArchivoOut,#0,#0);
  write(ArchivoOut,'WAVEfmt ');                  { Formato WAVE                }
  write(ArchivoOut,#16,#0,#0,#0,#1,#0,#1,#0);    { Datos constantes... (¿?)    }
  write(ArchivoOut,c,d,#0,#0,c,d,#0,#0);         { Frecuencia de muestreo      }
  write(ArchivoOut,#1,#0,#8,#0,'data');
  write(ArchivoOut,chr(n  MOD 256));             { Número de muestras          }
  write(ArchivoOut,chr((n DIV 256) MOD 256));
  write(ArchivoOut,#0,#0);
  for i:=1 to n do
  begin
    readln(Archivoin,ValorTexto);
    val(ValorTexto,x,k);
    j:=trunc(127.0*x/max + 128);
    c:=chr(j);
    write(ArchivoOut,c);
  end;
  close(ArchivoIn); close(ArchivoOut);
end.
```

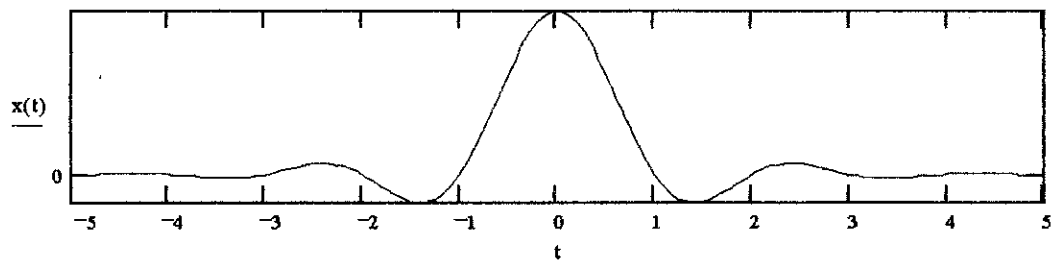
2.1 Espectro de x(t) y de su version muestreada x[n] = x(nT)

$$x(t) := \text{if}\left(t = 0, 1, \frac{\sin(\pi \cdot t)}{\pi \cdot t}\right) \cdot \frac{\cos\left(\frac{\pi \cdot t}{5}\right) + 1}{2}$$

Señal análoga a discretizar

$t := -5, -4.92 .. 5$

Rango de definición de la señal análoga



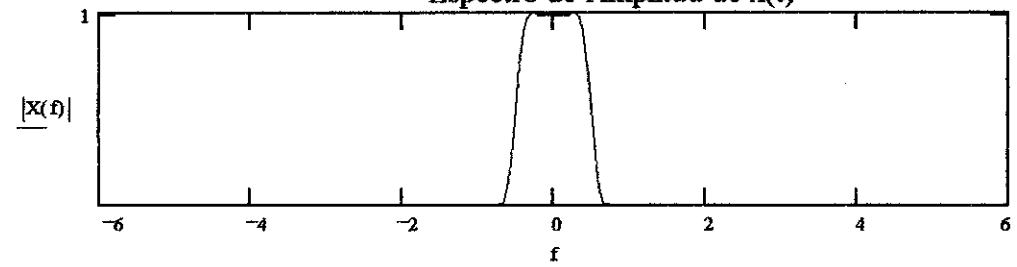
$$X(f) := \int_{-5}^{5} x(t) \cdot \exp(-2i \cdot \pi \cdot f \cdot t)\, dt$$

Composición espectral de x(t), calculada mediante la Transformada de Fourier

$f := -6, -5.95 .. 6$

Rango de frecuencias a estudiar

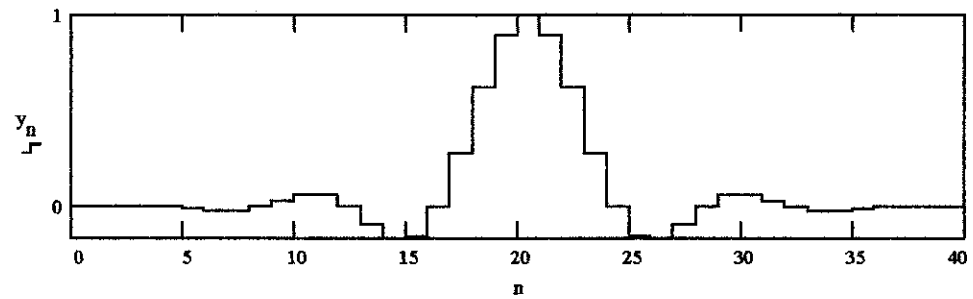
Espectro de Amplitud de x(t)



$n := 0 .. 40$

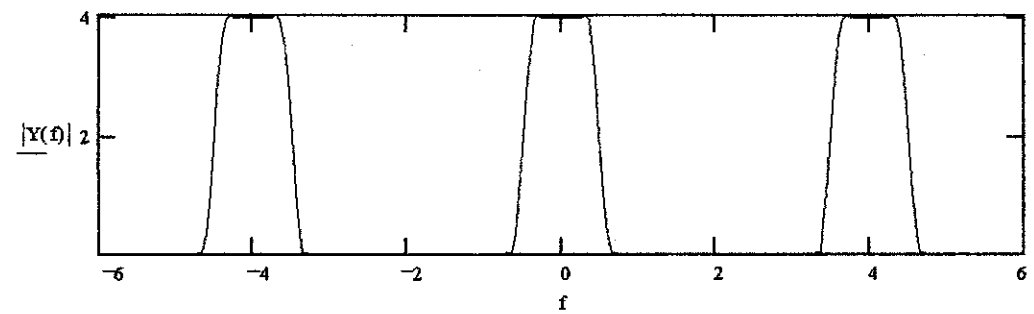
$$y_n := x\left(\frac{n}{4} - 5\right)$$

Se tomarán 41 muestras separadas 0.25 unidades de tiempo entre si para una tasa de muestreo de 4 Hz



$$Y(f) := \sum_n y_n \cdot \exp\left(\frac{-2i \cdot \pi \cdot n \cdot f}{4}\right)$$

Composición espectral de y[n], calculada mediante la Transformada de Fourier en tiempo discreto



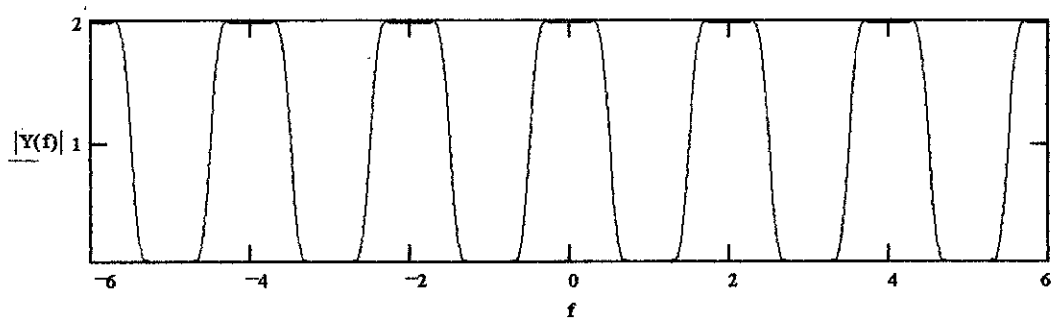
El espectro de la señal análoga se repite cada cuatro Hz en el espectro de la señal muestreada

$n := 0..20$

$y_n := x\left(\frac{n}{2} - 5\right)$

$Y(f) := \sum_n y_n \cdot \exp\left(\frac{-2i \cdot \pi \cdot n \cdot f}{2}\right)$

Al bajar la tasa de muestreo a 2 Hz las réplicas del espectro de la señal análoga se acercan más, pero aún se pueden separar.

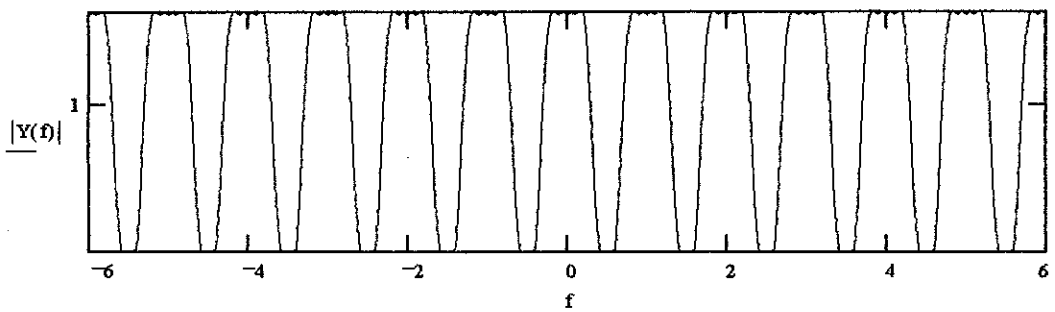


$n := 0..16$

$y_n := x\left(\frac{n}{1.6} - 5\right)$

$Y(f) := \sum_n y_n \cdot \exp(-2i \cdot \pi \cdot n \cdot f)$

La tasa de muestreo crítica es de 1.6 Hz pues el ancho de banda de la señal análoga original es de cerca de 0.8 Hz. Aunque aún se pueden distinguir las distintas réplicas ya es muy difícil tratar de separarlas.

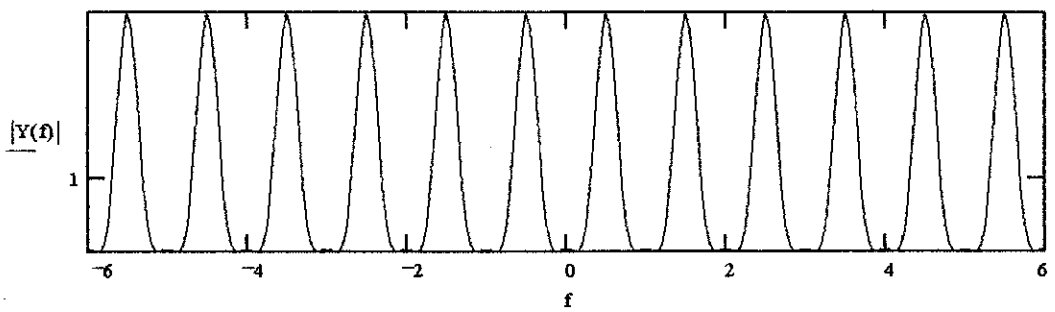


$n := 0..8$

$y_n := x\left(\frac{n}{0.8} - 5\right)$

$Y(f) := \sum_n y_n \cdot \exp(-2i \cdot \pi \cdot n \cdot f)$

A tasas de muestreo menores que la frecuencia Nyquist el fenómeno del aliasing impide una adecuada representación de la señal original.

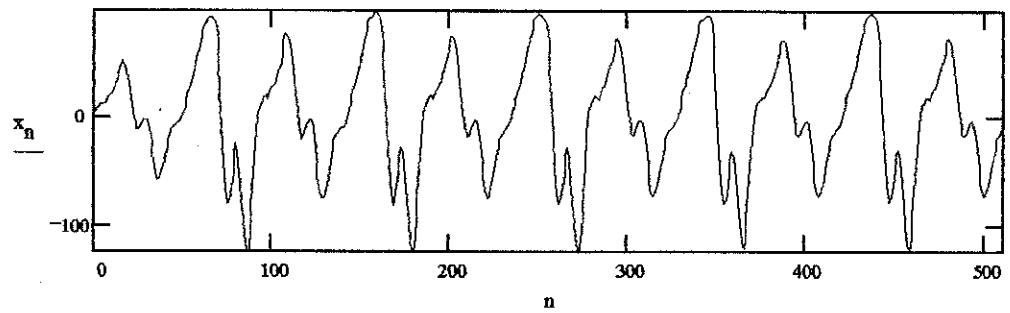


2.2 Transformada rápida de Fourier, FFT

$x := READPRN(a)$ Señal a analizar

$N := 512$ Número de muestras

$n := 0..N-1$ Indice de tiempo

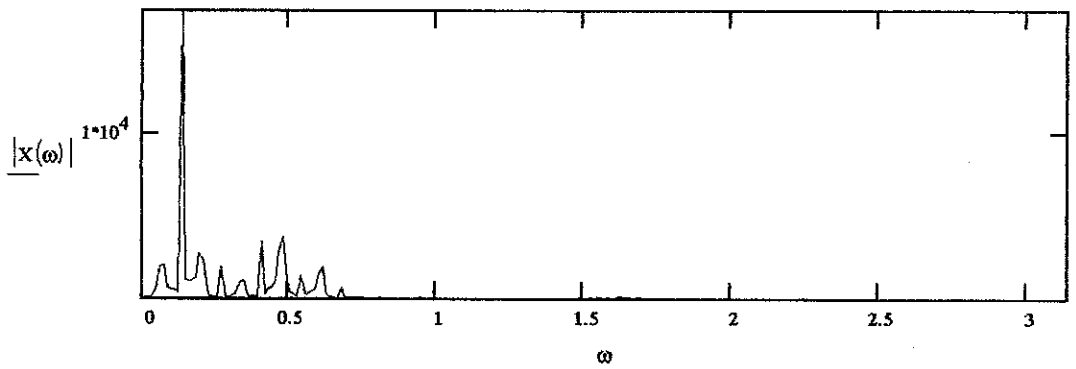


$$X(\omega) := \sum_n x_n \cdot \exp(-i \cdot \omega \cdot n)$$

Transformada de Fourier en tiempo discreto, DTFT, definida en el dominio continuo de la frecuencia normalizada, $\omega = 2\pi f / fs$, donde fs es la frecuencia de muestreo.

$$\omega := 0, \frac{2\cdot\pi}{N} .. \pi$$

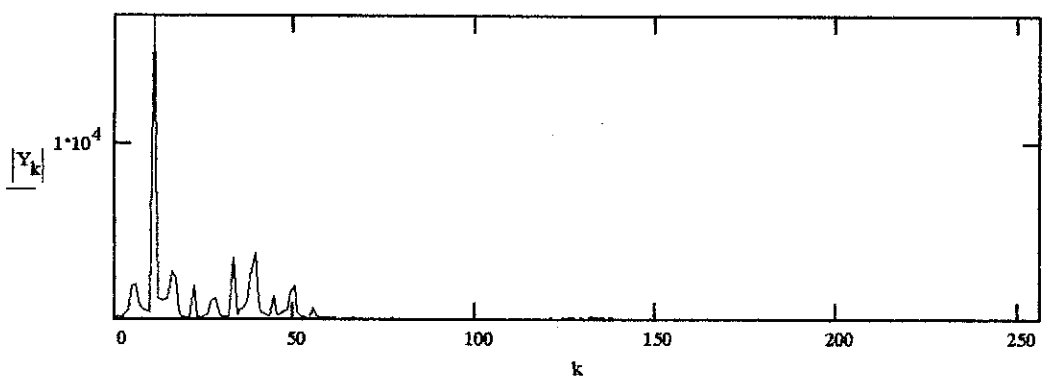
Muestreo en frecuencia de la DTFT para f de 0 a fs/2



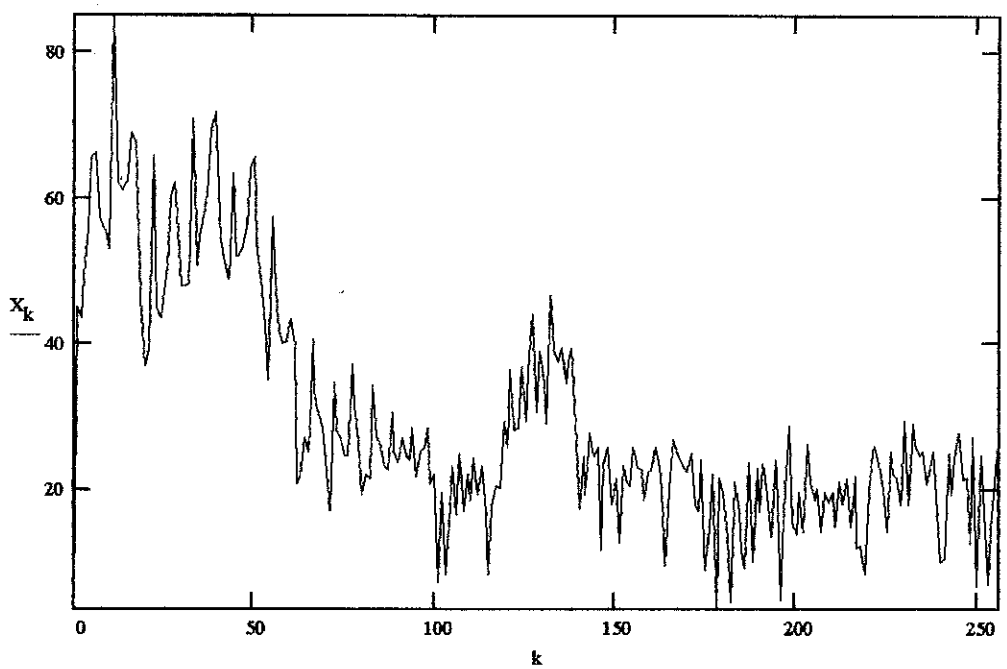
$Y := fft(x) \cdot \sqrt{N}$

$k := 0.. \frac{N}{2}$

La transformada rápida de Fourier, FFT, calcula las muestras de la DTFT directamente con un algoritmo muy eficiente. La FFT particular que utiliza MathCAD requiere que N sea potencia de dos y retorna 1+N/2 muestras en frecuencia, correspondientes a una frecuencia entre 0 y fs/2, donde fs es la frecuencia de muestreo. Por alguna razón muy particular de MathCAD, la magnitud de la FFT viene normalizada respecto a la raíz del número de muestras en tiempo.



$X_k := 20 \cdot \log(|Y_k|)$ Una forma más común de representar el espectro de amplitud es en unidades relativas o decibeles:



Nótese que se trata de una serie de picos separados cada 6.5 muestras en frecuencia, aproximadamente, y multiplicados por una envolvente con máximos a las 13, 40 y 132 muestras. Conociendo que fs=11025 Hz, podemos determinar las frecuencias correspondientes:

$\frac{6.5 \cdot 11025}{N} = 139.966$ Los armónicos se suceden cada 140 Hz, que corresponde al tono de la señal de voz

$\frac{13 \cdot 11025}{N} = 279.932$ La primera frecuencia de resonancia de la cavidad oral es de 280 Hz

$\frac{40 \cdot 11025}{N} = 861.328$ La segunda frecuencia de resonancia de la cavidad oral es de 861.3 Hz

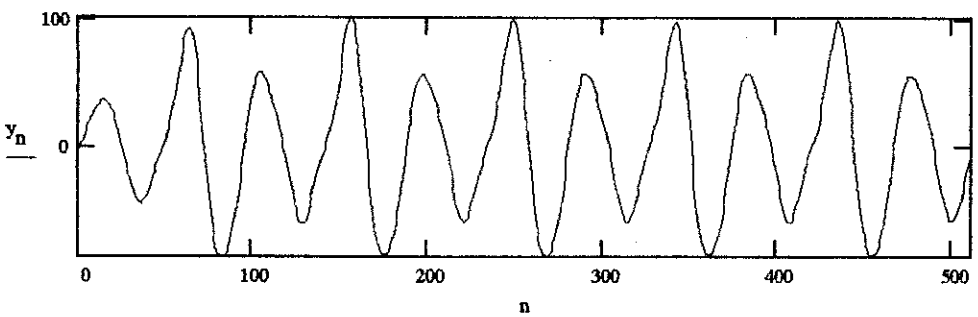
$\frac{132 \cdot 11025}{N} = 2.842 \cdot 10^3$ La tercera frecuencia de resonancia de la cavidad oral es de 2.84 kHz

---

$k := 30 .. \frac{N}{2}$ $Y_k := 0$

$y := \frac{\text{ifft}(Y)}{\sqrt{N}}$

En el lado de la transformada es fácil hacer operaciones como el filtraje mediante la eliminación de las frecuencias indeseadas. En este caso, un filtro pasabajos. Nótese nuevamente la normalización inversa en la IFFT.



2.3 Diseño de filtros FIR mediante ventanas en el tiempo

$\text{sinc}(\alpha) := \text{if}\left(\alpha = 0, 1, \frac{\sin(\pi\cdot\alpha)}{\pi\cdot\alpha}\right)$ Definición de la función SENO-C

$N := 20$ Orden de los filtros

$k := 0..N-1$ Indice de tiempo

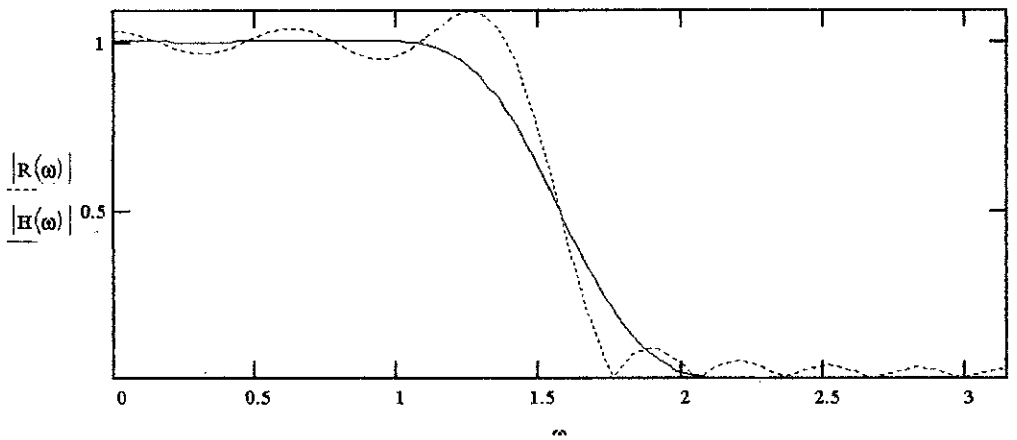
$r_k := \frac{\text{sinc}\left(\frac{k-0.5\cdot N}{2}\right)}{2}$ Filtro pasabajos de media banda mediante ventana rectangular

$h_k := \left(0.54 - 0.46\cdot\cos\left(\frac{2\cdot\pi\cdot k}{N}\right)\right)\cdot r_k$ Filtro pasabajos de media banda mediante ventana Hamming

$R(\omega) := \sum_k r_k\cdot\exp(-i\cdot k\cdot\omega)$

$H(\omega) := \sum_k h_k\cdot\exp(-i\cdot k\cdot\omega)$ Respuesta en frecuencia de cada filtro

$\omega := 0, 0.01\cdot\pi..\pi$



Aunque el filtro mediante ventana rectangular permite una transición más rápida, sus bandas pasante y no-pasante presentan oscilasciones indeseables. El filtro mediante ventana Hamming tiene una transición más lenta pero, en cambio, sus bandas pasante y no-pasante son más planas.

$N := 128$ Orden de los filtros

$k := 0..N-1$ Indice de tiempo de las respuestas al impulso

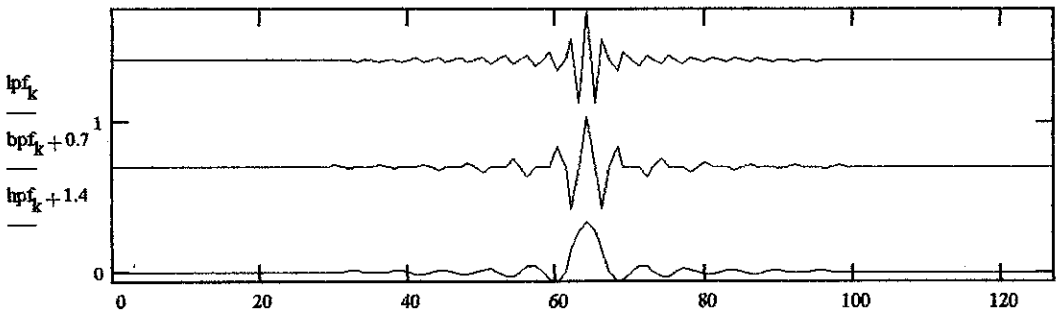
Forma general de la k-ésima muestra de un filtro FIR pasabanda con frecuencias de corte inferior y superior iguales a $\theta 1$ y $\theta 2$, respectivamente.

$$g(k,\theta 1,\theta 2) := \left[\frac{\theta 2}{\pi}\cdot\text{sinc}\left[\frac{(k-0.5\cdot N)\cdot\theta 2}{\pi}\right] - \frac{\theta 1}{\pi}\cdot\text{sinc}\left[\frac{(k-0.5\cdot N)\cdot\theta 1}{\pi}\right]\right]\cdot\left(0.54 - 0.46\cdot\cos\left(\frac{2\cdot\pi\cdot k}{N}\right)\right)$$

$lpf_k := g\left(k, 0, \frac{\pi}{3}\right)$ Coeficientes de un filtro pasabajos a fs/6

$bpf_k := g\left(k, \frac{\pi}{3}, \frac{2\cdot\pi}{3}\right)$ Coeficientes de un filtro pasabanda entre fs/6 y fs/3

$hpf_k := g\left(k, \frac{2\cdot\pi}{3}, \pi\right)$ Coeficientes de un filtro pasaalto desde fs/3 (hasta fs/2, claro, porque la señal muestreada no debe tener componentes espectrales mayores a fs/2).

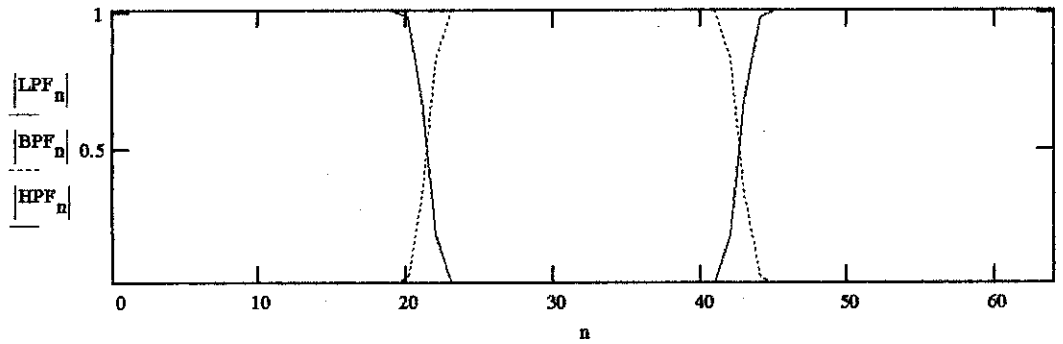


$LPF := fft(lpf)\cdot\sqrt{N}$ Respuesta en frecuencia del filtro pasabajos

$BPF := fft(bpf)\cdot\sqrt{N}$ Respuesta en frecuencia del filtro pasabanda

$HPF := fft(hpf)\cdot\sqrt{N}$ Respuesta en frecuencia del filtro pasaaltos

$n := 0..\ 0.5\cdot N$



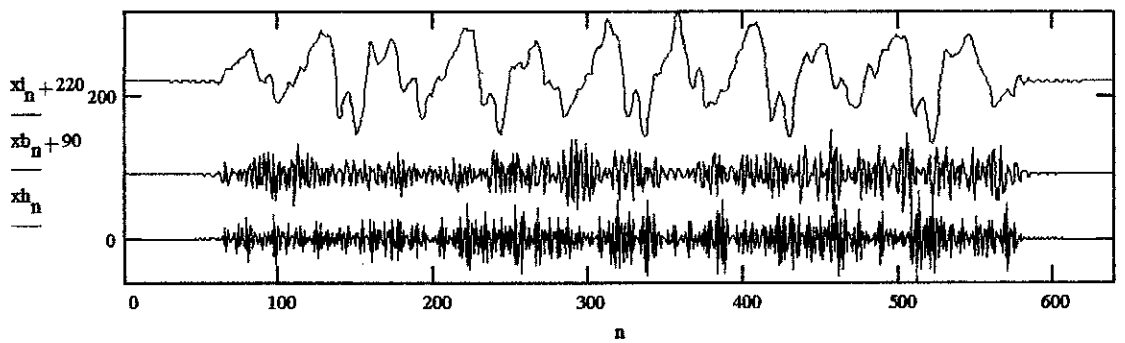
$x := READPRN(sa)$ Señal de voz a filtrar

$n := 0..\ size(x) + N$ Número de muestras en las señales filtradas

$$xl_n := \sum_k lpf_k \cdot if\left(n \geq k, if\left(n - k \leq size(x), x_{n-k}, 0\right), 0\right)$$ Salida del filtro pasabajos

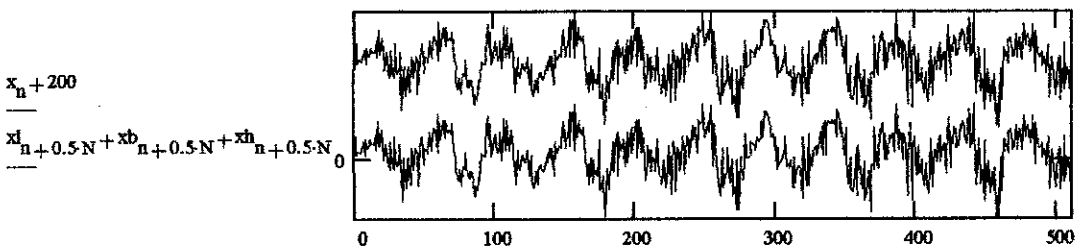
$$xb_n := \sum_k bpf_k \cdot if\left(n \geq k, if\left(n - k \leq size(x), x_{n-k}, 0\right), 0\right)$$ Salida del filtro pasabanda

$$xh_n := \sum_k hpf_k \cdot if\left(n \geq k, if\left(n - k \leq size(x), x_{n-k}, 0\right), 0\right)$$ Salida del filtro pasaaltos



$n := 0..\ size(x)$

La señal original es aproximadamente igual a la suma de las tres bandas



2.4 Convolución mediante FFT

$N := 512$ Orden del filtro

$k := 0..N-1$ Indice de tiempo para la respuesta al impulso

$$h_k := \text{if}\left[k = 0.5\cdot N,\ 0.2,\ \frac{\sin\left[\frac{(k-0.5\cdot N)\cdot\pi}{5}\right]}{(k-0.5\cdot N)\cdot\pi}\right]\cdot\left(0.54-0.46\cdot\cos\left(\frac{2\cdot\pi\cdot k}{N}\right)\right)$$

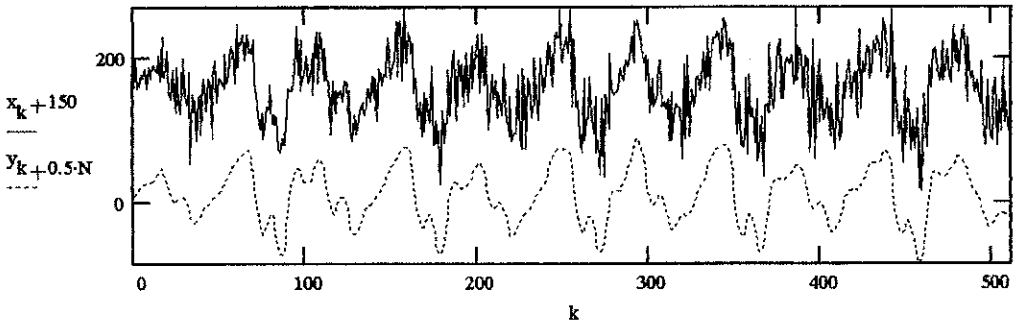
Respuesta al impulso del Filtro pasabajos

$x := \text{READPRN}(sa)$ Señal a filtrar

$n := 0..2\cdot N-1$ Indice de tiempo de la señal filtrada

$$y_n := \sum_k h_k\cdot\text{if}\left(n\geq k,\text{if}\left(n-k<N,x_{n-k},0\right),0\right)$$

Convolución lineal



¿Se debe esperar obtener el mismo resultado con la IFFT del producto de las FFT?

$X := \text{fft}(x)\cdot\sqrt{N}$ Espectro de la señal a filtrar

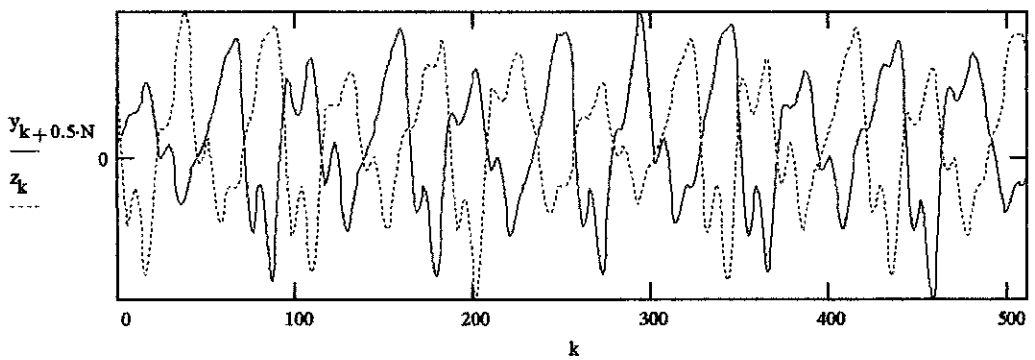
$H := \text{fft}(h)\cdot\sqrt{N}$ Espectro de la respuesta al impulso del filtro

$n := 0..0.5\cdot N$ Indice de frecuencia

$Z_n := X_n\cdot H_n$ Producto de las transformadas

$$z := \frac{\text{ifft}(Z)}{\sqrt{N}}$$

Transformada inversa del producto de las transformadas: ¡No es la convolución lineal!



Se obtuvo la "Convolución Circular" debido a que el espectro discreto de señales muestreadas equivale a la serie de Fourier de señales periódicas muestreadas. Para calcular la convolución lineal mediante la FFT, se debe hacer un "relleno" de la señal y la respuesta al impulso del filtro para evitar la superposición de los supuestos períodos:

$k := N .. 2 \cdot N - 1$ Muestras adicionales

$x_k := 0$ Relleno de la señal

$h_k := 0$ Relleno de la respuesta al impulso

$X := fft(x) \cdot \sqrt{2 \cdot N}$ Nuevo espectro de amplitud de la señal

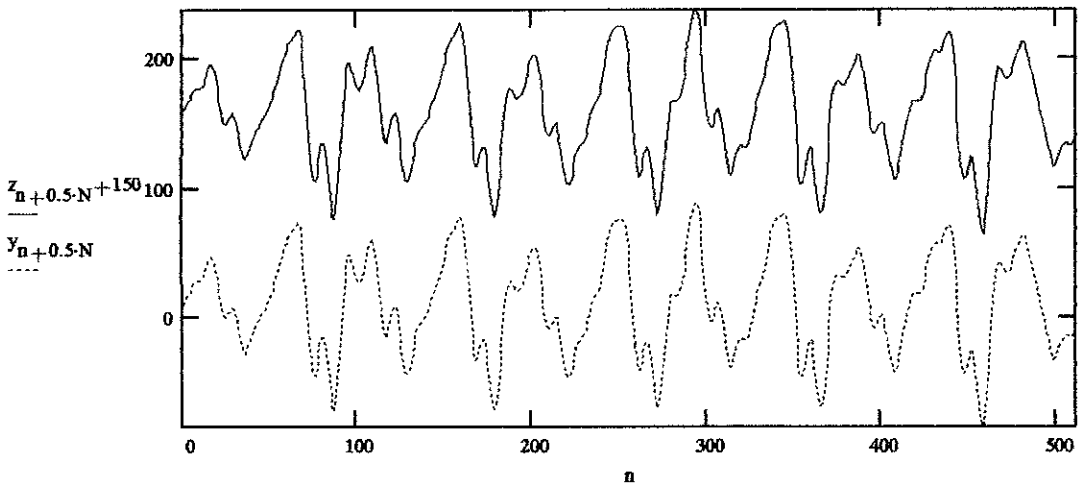
$H := fft(h) \cdot \sqrt{2 \cdot N}$ Nuevo espectro de la respuesta al impulso

$n := 0 .. N$ Indice de frecuencias

$Z_n := X_n \cdot H_n$ Espectro de la convolución lineal

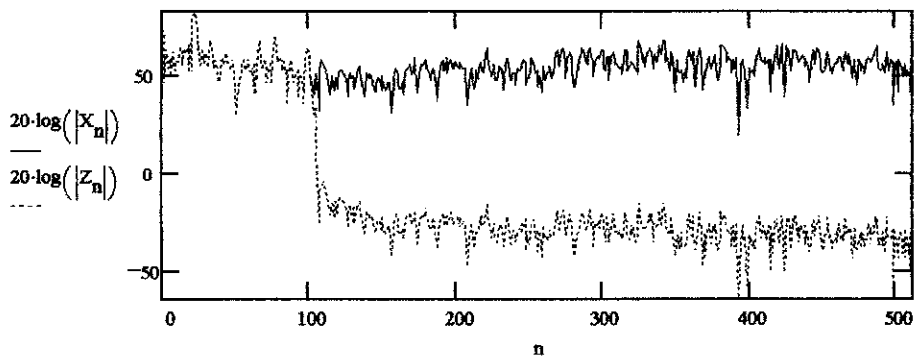
$z := \frac{ifft(Z)}{\sqrt{2 \cdot N}}$ Salida del filtro

$n := 0 .. N - 1$ Indice de tiempo



Ahora si se obtuvo la verdadera salida del filtro. El cálculo mediante la FFT se hizo muchas veces más rápido que mediante la evaluación de la suma de convolución.

Comparación espectral a la entrada y la salida del filtro:



3. Modelo de producción de voz

Coeficientes del filtro inverso para el fonema /A/. Sonoro, con período de 85 muestras

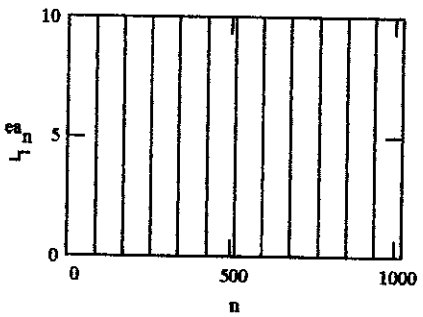
$a := READPRN(c\_lpc\_a)$

$$a = \begin{bmatrix} 1 \\ -2.326 \\ 1.652 \\ -0.3113 \\ -0.08789 \\ 0.6424 \\ -0.5074 \\ -0.5582 \\ 0.3833 \\ 0.3652 \\ -0.2332 \end{bmatrix}$$

$n := 0..1023$

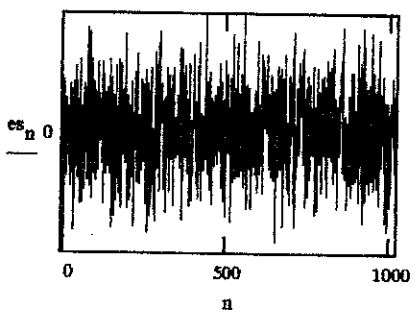
Fuente de excitación sonora

$$ea_n := if\left(n = 85 \cdot floor\left(\frac{n}{85}\right), 10, 0\right)$$



Fuente de excitación sorda

$$es_n := \sqrt{-40 \cdot \ln(rnd(1))} \cdot \cos(2 \cdot \pi \cdot rnd(1))$$



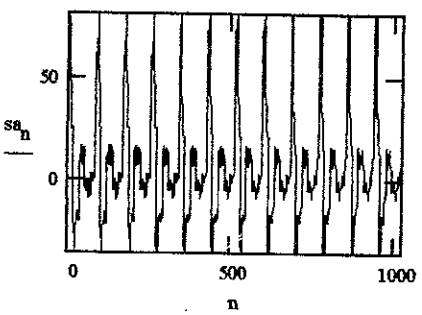
Síntesis del fonema /A/

$sa_0 := ea_0$

$n := 1..1023 \qquad k := 1..10$

$$sa_n := ea_n - \sum_k a_k \cdot if(n - k < 0, 0, sa_{n-k})$$

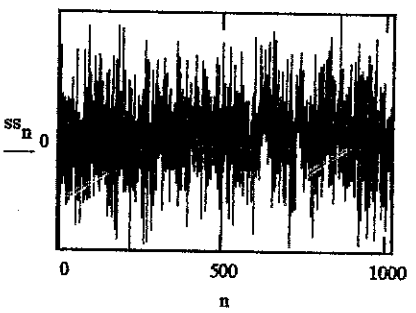
$n := 0..1023$



Síntesis del fonema /S/

$ss_0 := es_0$

$$ss_n := es_n - \sum_k s_k \cdot if(n - k < 0, 0, ss_{n-k})$$



Síntesis de la palabra /ASA/

$x_n := sa_n \cdot \left\langle 0.55 - 0.45 \cdot \cos\left\langle \frac{\pi \cdot n}{512} \right\rangle \right\rangle$ Fonema /A/

$n := 800 .. 1023$

$x_n := x_n + \frac{n - 800}{224} \cdot ss_{n-800}$ Transición de /A/ a /S/

$n := 224 .. 1023$

$x_{n+800} := ss_n$ Fonema /S/

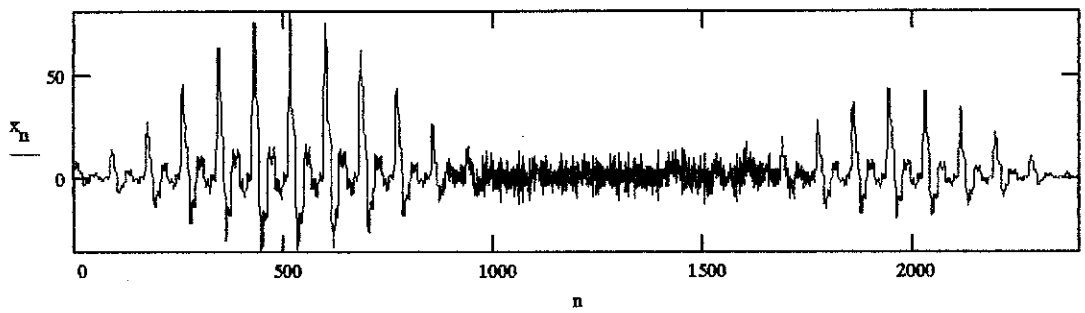
$n := 1600 .. 1823$

$x_n := x_n \cdot \left\langle 1 - \frac{n - 1600}{224} \right\rangle + x_{n-1600}$ Transición de /S/ a /A/

$n := 224 .. 800$

$x_{n+1600} := x_n \cdot \frac{800 - n}{676}$ Fonema /A/ final

$n := 0 .. 2400$

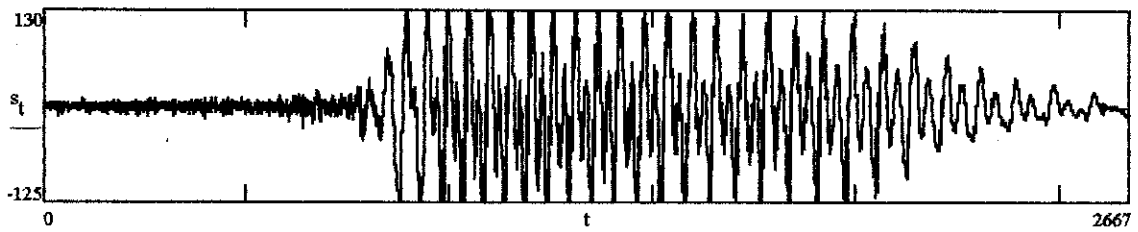


WRITEPRN(as) := x Se almacena en un archivo .PRN para convertir a .WAV y escuchar la palabra sintetizada (ver sección 1.2)

4.1 Energía promedio y Tasa promedio de cruces por cero

$s := READPRN(si)$ Señal de voz a analizar: /SI/

$t := 0..size(s)$ Indice de tiempo



Se escogen 51 tramas de 100 muestras cada una, separadas entre si por 50 muestras:

$n := 0..99$ Indice dentro de cada Trama

$k := 0..50$ Número de tramas

$signo(x) := if(x<0,-1,1)$ Función 'Signo de x'

$$E_k := \frac{\sum (s_{n+50\cdot k})^2}{100}$$

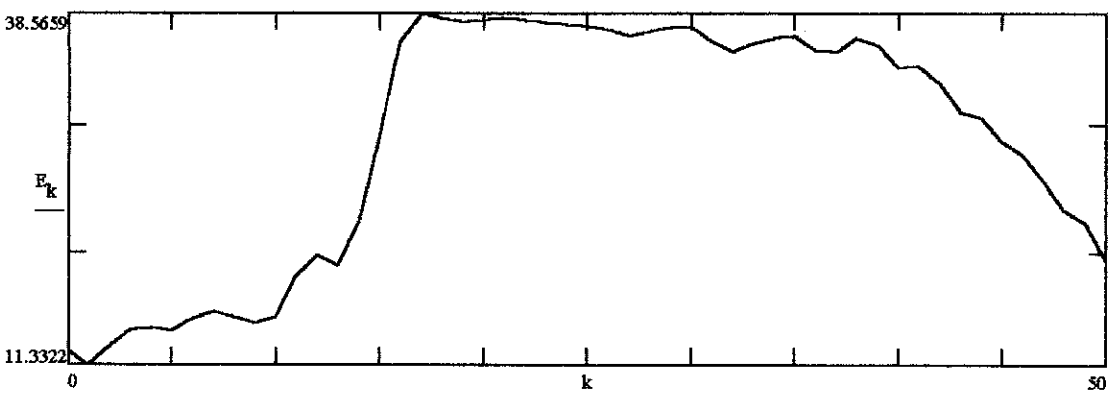
Función de Energía en Tiempo Corto

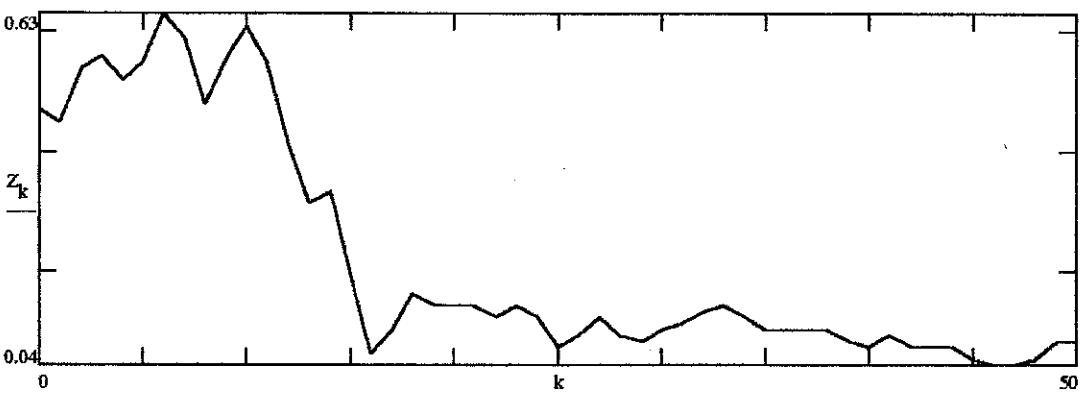
$$Z_k := \frac{\sum |signo(s_{n+50\cdot k}) - signo(s_{n+50\cdot k+1})|}{200}$$

Función de Tasa Promedio de Cruces por Cero en Tiempo Corto

Debido al rango dinámico de la Energía promedio en tiempo corto, es conveniente visualizarla en una escala logarítmica:

$E_k := 10\cdot\log(E_k)$ Energía en decibeles





Obsérvese cómo, durante el sonido sordo, la energía es baja y la tasa de cruces por cero es alta mientras que, durante el sonido sonoro, la energía es alta y la tasa promedio de cruces por cero es baja. Esto sugiere un algoritmo para determinar la condición sonoro/sordo:

Primero se normalizan las funciones de energía y cruces por cero:

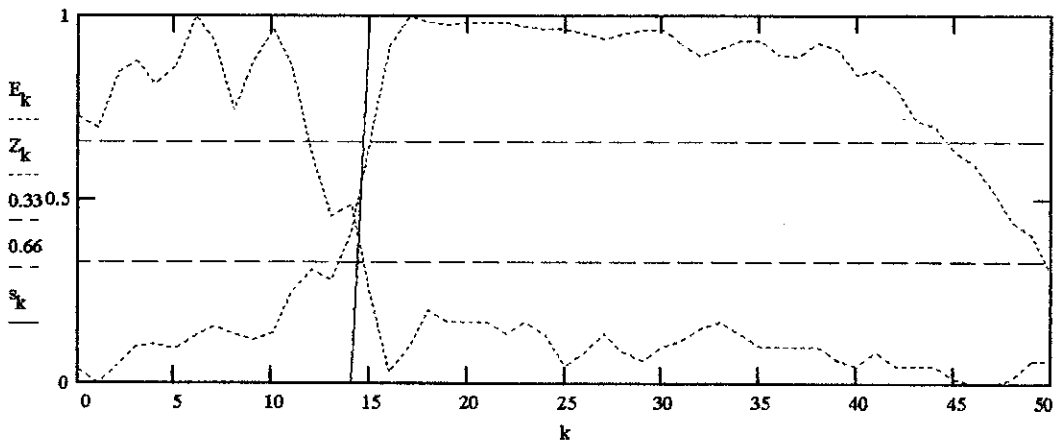
$em := \min(E)$ $\qquad$ $zm := \min(Z)$

$ex := \max(E)$ $\qquad$ $zx := \max(Z)$

$$E_k := \frac{E_k - em}{ex - em} \qquad Z_k := \frac{Z_k - zm}{zx - zm}$$

Y ahora se comparan sus valores respectos a dos niveles:

$s_k := -1$	La clasificación de cada ventana se deja incierta
$s_k := \text{if}(E_k > 0.66, \text{if}(Z_k < 0.66, 1, s_k), s_k)$	Ventana sonora
$s_k := \text{if}(Z_k > 0.66, \text{if}(E_k < 0.66, 0, s_k), s_k)$	Ventana sorda
$s_k := \text{if}(E_k > 0.33, \text{if}(Z_k < 0.33, 1, s_k), s_k)$	Ventana sonora
$s_k := \text{if}(Z_k > 0.33, \text{if}(E_k < 0.33, 0, s_k), s_k)$	Ventana sorda
$s_k := \text{if}(s_k < 0, s_{k-1}, s_k)$	Se deja un ciclo de histéresis entre distintas clasificaciones

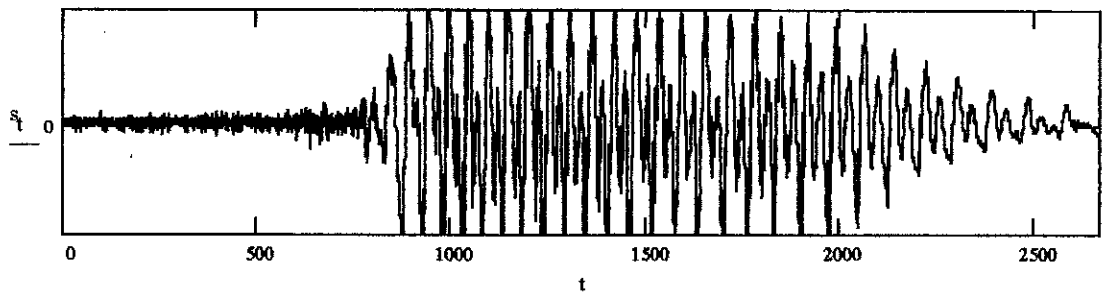


Hasta la ventana 14 la señal se clasifica como sorda y, desde la ventana 15, como sonora.

4.2 Autocorrelación en tiempo corto

$s := \text{READPRN}(si)$ Señal de voz a analizar

$t := 0..\text{size}(s)$ Indice de tiempo



Ahora se calculará la función de autocorrelación en un segmento sordo y en un segmento sonoro de la señal de voz:

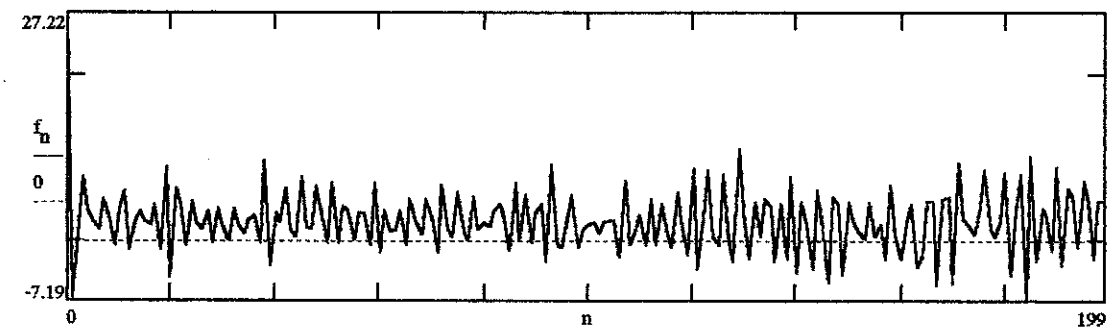
$k := 0..99$ $n := 0..199$

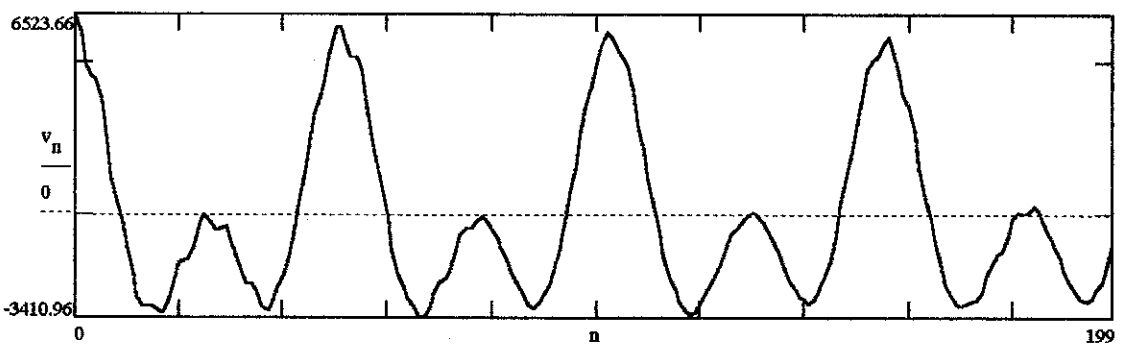
$$f_n := \sum_k \frac{s_{200+k} \cdot s_{200+k+n}}{100}$$

$$v_n := \sum_k \frac{s_{1000+k} \cdot s_{1000+k+n}}{100}$$

Autocorrelación del segmento sordo (fricativo) entre las muestras 200 y 400

Autocorrelación del segmento sonoro (vocalizado) entre las muestras 1000 y 1200

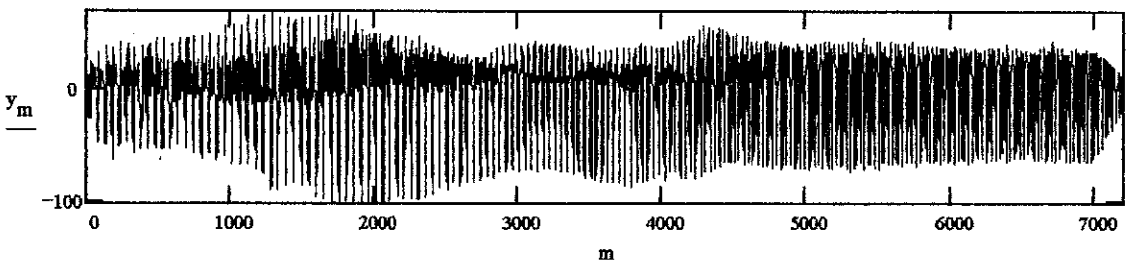




En el primer caso no hay evidencia de periodicidad, mientras en el segundo hay un pico claro cada período de la señal de voz. Esta es una indicación de sonoridad pero, principalmente, sugiere un método para detección de tono, como se explora en la sección 4.3

4.3 Estimación de tono

$y := READPRN(do\_mayor)$ Señal de voz a analizar
$m := 0..size(y)$ Indice de tiempo



Primero se pasa la señal por un filtro pasabajos a 375 Hz. Por tratarse de una frecuencia de muestreo de 6 kHz, el filtro es de 1/8 de banda:

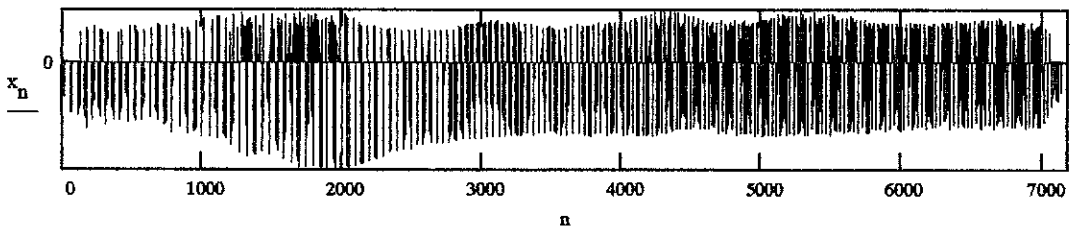
$sinc(\alpha) := if\left(\alpha = 0, 1, \frac{\sin(\pi \cdot \alpha)}{\pi \cdot \alpha}\right)$ Definición de la función SENO-C

$k := 0..19$ Indice de tiempo

$h_k := (0.54 - 0.46 \cdot \cos(0.1 \cdot \pi \cdot k)) \cdot sinc((k - 10) \cdot 0.125) \cdot 0.125$ Filtro pasabajos de 1/4 banda

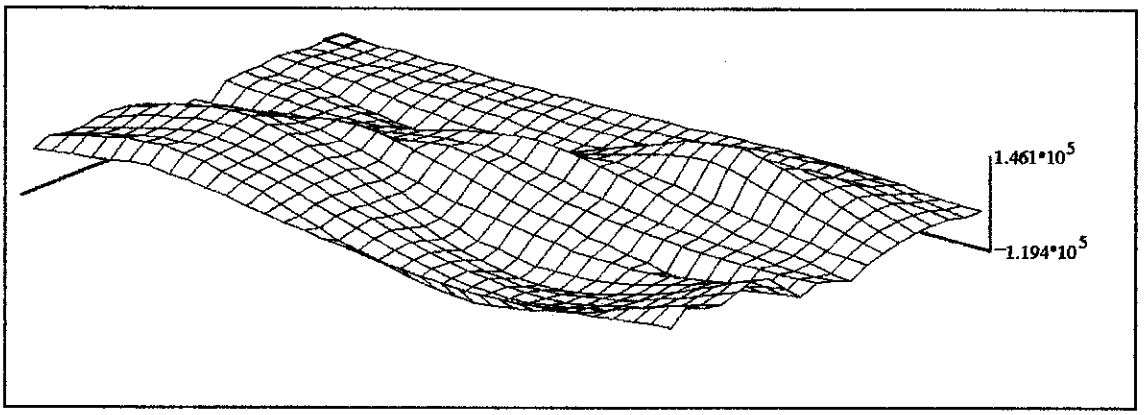
$n := 0..size(y) - 20$ $x_n := \sum_k h_k \cdot y_{n+20-k}$ Se filtra la señal

$x_n := if(|x_n| < 20, 0, x_n)$ Y se recorta para "aplanamiento" espectral



$k := 0..19$ Se tomarán 20 ventanas
$i := 0..159$ de 160 muestras cada una
$n := 26..59$ y se considerarán posibles tonos entre 100 y 230 Hz (6000/59..6000/26)

$T_{k,n-26} := \sum_i x_{365 \cdot k + i} \cdot x_{365 \cdot k + i + n}$ Se espera que la función de autocorrelación presente máximos cada período.



T

$n := 0..33$

El instante en que se presente el máximo de cada ventana indica su período. El tono está dado por su inverso.

$i := 0 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$

$i := 1 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 1

$i := 2 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 2

$i := 3 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 3

$i := 4 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 4

$i := 5 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 5

$i := 6 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 6

$i := 7 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 7

$i := 8 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 8

$i := 9 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 9

$i := 10 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 10

$i := 11 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 11

$i := 12 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 12

$i := 13 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 13

$i := 14 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 14

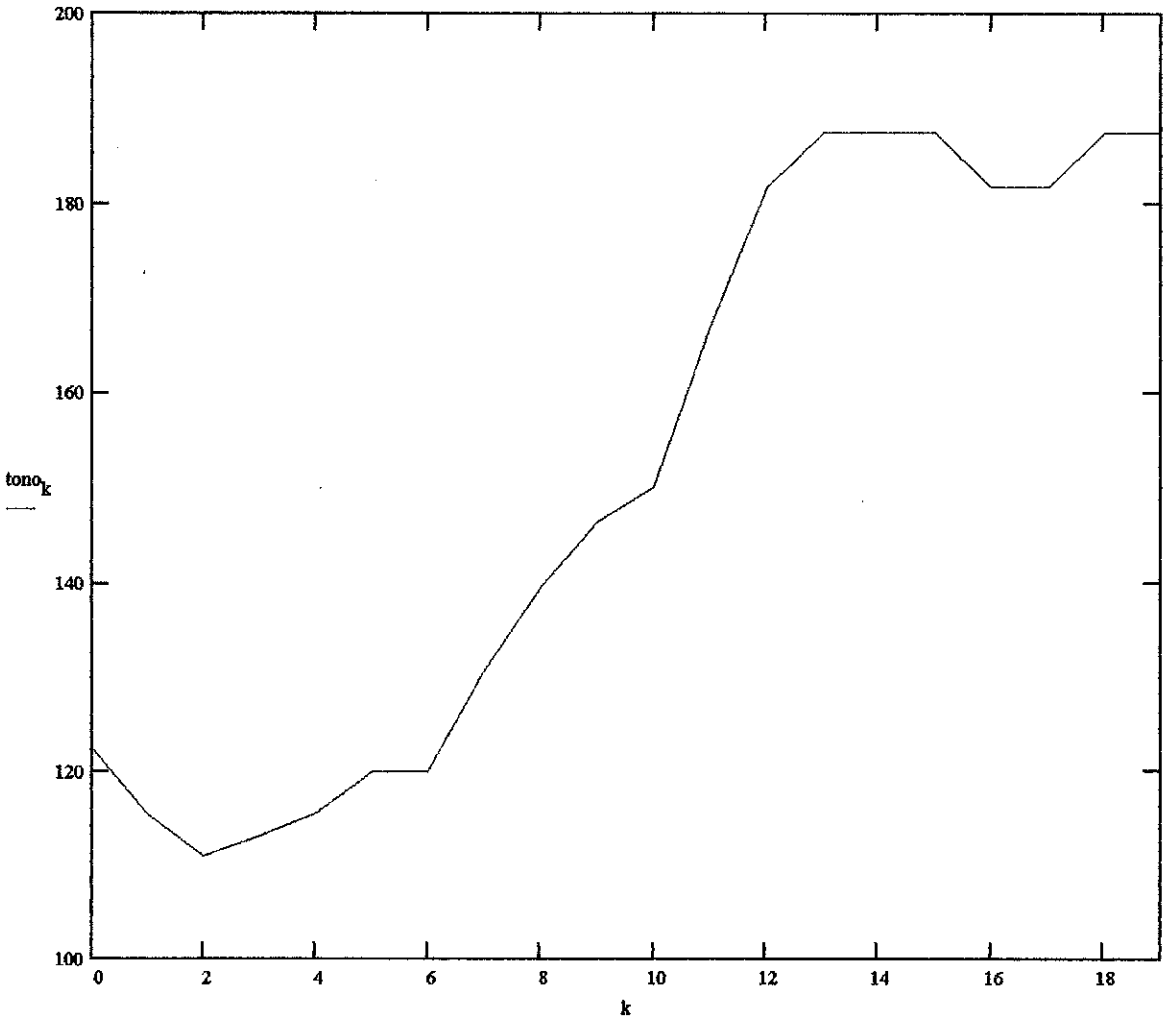
$i := 15 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 15

$i := 16 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 16

$i := 17 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 17

$i := 18 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 18

$i := 19 \quad t_n := T_{i,n} \quad m := \max(t) \quad tono_i := \frac{6000}{26 + \sum^{n} if(m = t_n, n, 0)}$ Ventana 19

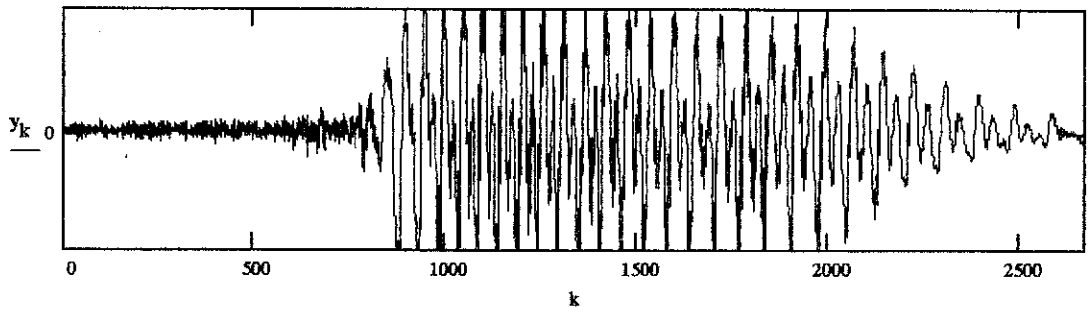


Se trató de cantar la secuencia de la tríada correspondiente a Do Mayor (do - mi - sol). Así, la función de tono debió ser una escalera con escalones en 131, 165 y 196 Hz. El programa muestra que sólo se alcanzó a cantar la secuencia de Si Mayor ligeramente desafinada...

4.4 Transformada de Fourier en tiempo corto

$y := READPRN(si)$ Señal a analizar

$k := 0..size(y)$ Indice de tiempo



Se construirán 36 ventanas de 256 muestras cada una, separadas 64 muestras entre si.

$j := 0..35$ Número de ventana

$i := 0..90$ Indice de frecuencia (de 0 a 3900 Hz)

$n := 0..255$ Indice de tiempo en cada ventana

Construye cada ventana, calcula su FFT y almacena la amplitud en un arreglo bidimensional

$x_n := y_{64\cdot0+n}$ $X := fft(x)$ $S_{i,0} := |X_i|$

$x_n := y_{64\cdot1+n}$ $X := fft(x)$ $S_{i,1} := |X_i|$

$x_n := y_{64\cdot2+n}$ $X := fft(x)$ $S_{i,2} := |X_i|$

$x_n := y_{64\cdot3+n}$ $X := fft(x)$ $S_{i,3} := |X_i|$

$x_n := y_{64\cdot4+n}$ $X := fft(x)$ $S_{i,4} := |X_i|$

$x_n := y_{64\cdot5+n}$ $X := fft(x)$ $S_{i,5} := |X_i|$

$x_n := y_{64\cdot6+n}$ $X := fft(x)$ $S_{i,6} := |X_i|$

$x_n := y_{64\cdot7+n}$ $X := fft(x)$ $S_{i,7} := |X_i|$

$x_n := y_{64\cdot8+n}$ $X := fft(x)$ $S_{i,8} := |X_i|$

$x_n := y_{64\cdot9+n}$ $X := fft(x)$ $S_{i,9} := |X_i|$

$x_n := y_{64\cdot10+n}$ $X := fft(x)$ $S_{i,10} := |X_i|$

$x_n := y_{64\cdot11+n}$ $X := fft(x)$ $S_{i,11} := |X_i|$

$x_n := y_{64\cdot12+n}$ $X := fft(x)$ $S_{i,12} := |X_i|$

$x_n := y_{64\cdot13+n}$ $X := fft(x)$ $S_{i,13} := |X_i|$

$x_n := y_{64\cdot14+n}$ $X := fft(x)$ $S_{i,14} := |X_i|$

$x_n := y_{64\cdot15+n}$ $X := fft(x)$ $S_{i,15} := |X_i|$

$x_n := y_{64\cdot16+n}$ $X := fft(x)$ $S_{i,16} := |X_i|$

$x_n := y_{64\cdot17+n}$ $X := fft(x)$ $S_{i,17} := |X_i|$

$x_n := y_{64\cdot18+n}$ $X := fft(x)$ $S_{i,18} := |X_i|$

$x_n := y_{64\cdot19+n}$ $X := fft(x)$ $S_{i,19} := |X_i|$

$x_n := y_{64\cdot20+n}$ $X := fft(x)$ $S_{i,20} := |X_i|$

$x_n := y_{64\cdot21+n}$ $X := fft(x)$ $S_{i,21} := |X_i|$

$x_n := y_{64\cdot22+n}$ $X := fft(x)$ $S_{i,22} := |X_i|$

$x_n := y_{64\cdot23+n}$ $X := fft(x)$ $S_{i,23} := |X_i|$

$x_n := y_{64\cdot24+n}$ $X := fft(x)$ $S_{i,24} := |X_i|$

$x_n := y_{64\cdot25+n}$ $X := fft(x)$ $S_{i,25} := |X_i|$

$x_n := y_{64\cdot26+n}$ $X := fft(x)$ $S_{i,26} := |X_i|$

$x_n := y_{64\cdot27+n}$ $X := fft(x)$ $S_{i,27} := |X_i|$

$x_n := y_{64\cdot28+n}$ $X := fft(x)$ $S_{i,28} := |X_i|$

$x_n := y_{64\cdot29+n}$ $X := fft(x)$ $S_{i,29} := |X_i|$

$x_n := y_{64\cdot30+n}$ $X := fft(x)$ $S_{i,30} := |X_i|$

$x_n := y_{64\cdot31+n}$ $X := fft(x)$ $S_{i,31} := |X_i|$

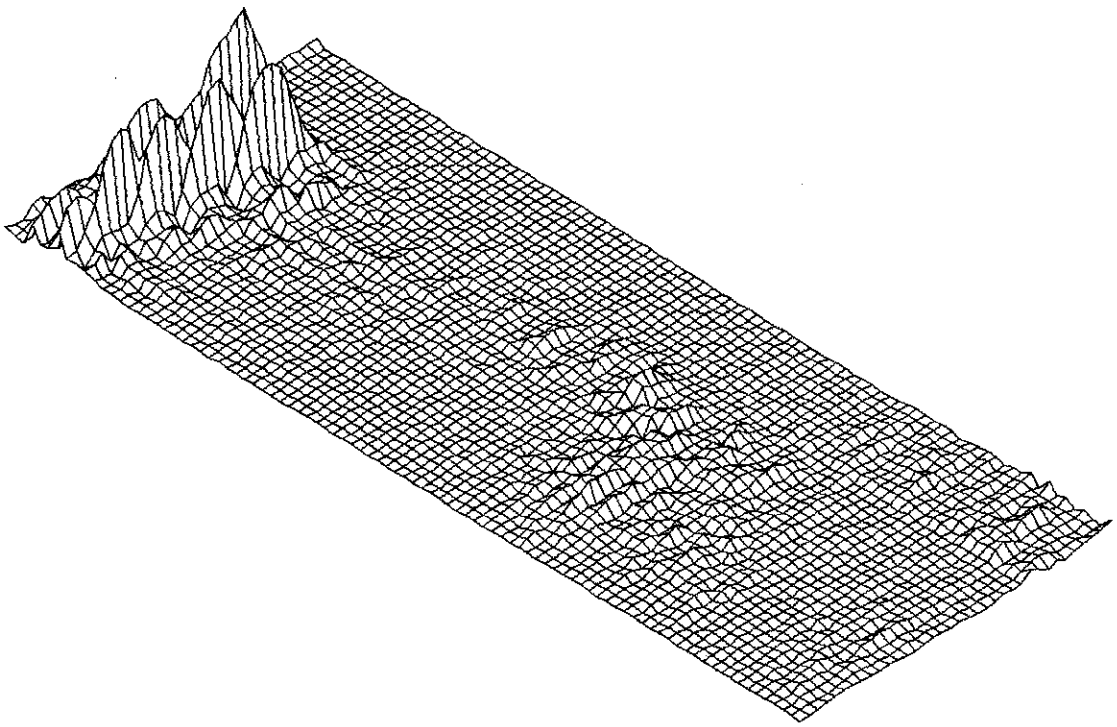
$x_n := y_{64\cdot32+n}$ $X := fft(x)$ $S_{i,32} := |X_i|$

$x_n := y_{64\cdot33+n}$ $X := fft(x)$ $S_{i,33} := |X_i|$

$x_n := y_{64\cdot34+n}$ $X := fft(x)$ $S_{i,34} := |X_i|$

$x_n := y_{64\cdot35+n}$ $X := fft(x)$ $S_{i,35} := |X_i|$

La gráfica del arreglo S se conoce como espectrograma y en él se nota cómo, durante el fonema /S/, prevalecen las altas frecuencias mientras que, durante el fonema /I/, prevalecen las bajas frecuencias con mucho mayor energía.



s

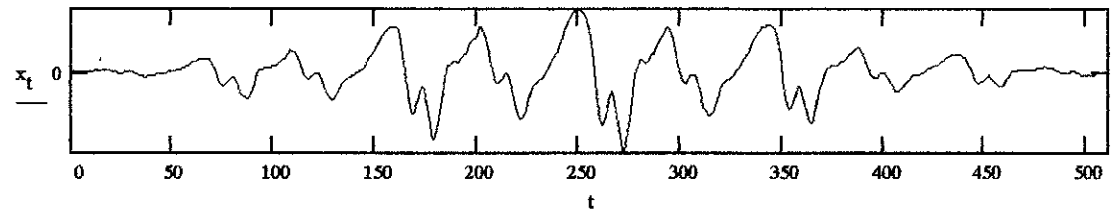
5. Análisis predictivo lineal, LPC

$x := READPRN(a)$ Muestras de la señal de voz

$N := (size(x) + 1)$ Número de muestras

$t := 0..N - 1$ Indice de tiempo

$x_t := (x_t) \cdot \left[0.54 + 0.46 \cdot \cos\left[2 \cdot \pi \cdot \left(\frac{t}{N} - 0.5 \right) \right] \right]$ Ventana Hamming



$p := 12$ Orden del filtro predictor

$\tau := 0..p$ Indice de la función de autocorrelación

$r_\tau := \sum_t x_t \cdot if(t + \tau < N, x_{t+\tau}, 0)$ Función de autocorrelación

$i := 0..p - 1$ $j := 0..p - 1$ Indices de la matriz de autocorrelación

$R_{i,j} := r_{|i-j|}$ Matriz de autocorrelación

$\rho_i := r_{i+1}$ Vector de autocorrelación

$R := R^{-1}$ Inversa de la matriz de autocorrelación

$a := R \cdot \rho$

$\alpha_{i+1} := -a_i$ $\alpha_0 := 1$ Vector de Coeficientes predictores

$k := p + 1..511$

$\alpha_k := 0$ Rellena con ceros para calcular la FFT

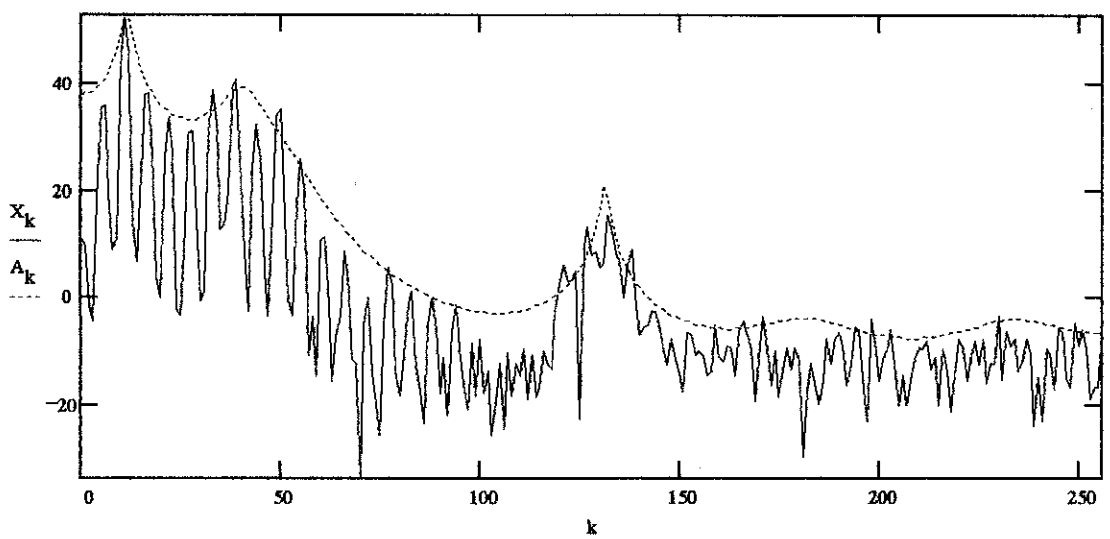
$A := fft(\alpha)$ Respuesta en frecuencia del filtro inverso

$k := 0..256$ Indice en la frecuencia

$A_k := 20 \cdot \log\left(\left| \frac{1}{A_k} \right| \right) - 20$ Respuesta en frecuencia del filtro directo

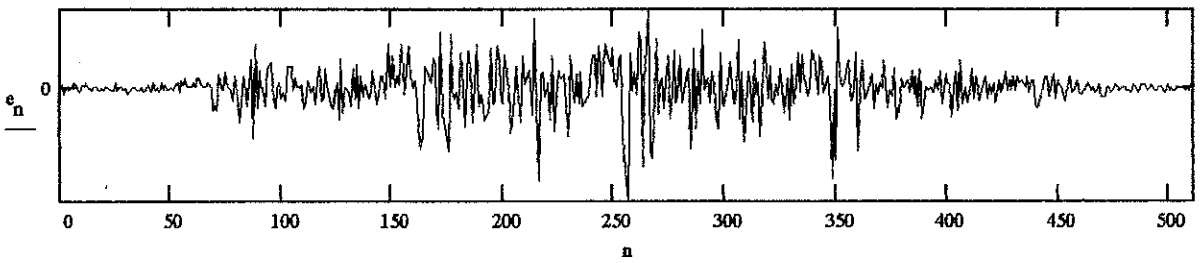
$X := fft(x)$

$X_k := 20 \cdot \log(|X_k|)$ Composición espectral de la señal original



Los formantos (frecuencias de resonancia de la cavidad oral) se encuentran en k=11, 40 y 131, correspondiendo a 11025*11/512 = 237 Hz, 11025*40/512 = 861 Hz y 11025*131/512= 2821 Hz.

$n := 0..511 \qquad i := 0..p \qquad e_n := \sum_i \alpha_i \cdot if(n - i \geq 0, x_{n-i}, 0)$ Error de predicción



$k := 0..40$

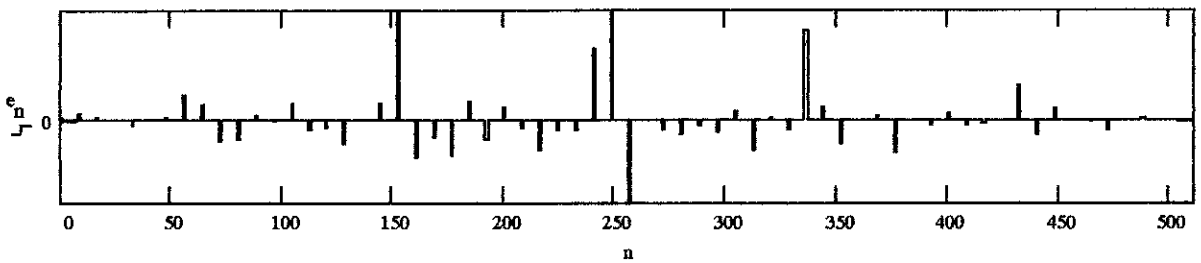
$h_k := if\left[k = 20, 0.125, 0.125 \cdot \frac{\sin(0.125 \cdot \pi \cdot (k-20))}{0.125 \cdot \pi \cdot (k-20)}\right] \cdot \left(0.54 - 0.46 \cdot \cos\left(k \cdot \frac{\pi}{20}\right)\right)$ Filtro pasabajos de 1/8 banda

$n := 0..551$

$y_n := \sum_k h_k \cdot if(n - k < 0, 0, if(n - k > 511, 0, e_{n-k}))$ Error filtrado

$n := 0..511$

$e_n := if\left(n = 8 \cdot floor\left(\frac{n}{8}\right), y_{n+20} \cdot 8, 0\right)$ Error submuestreado

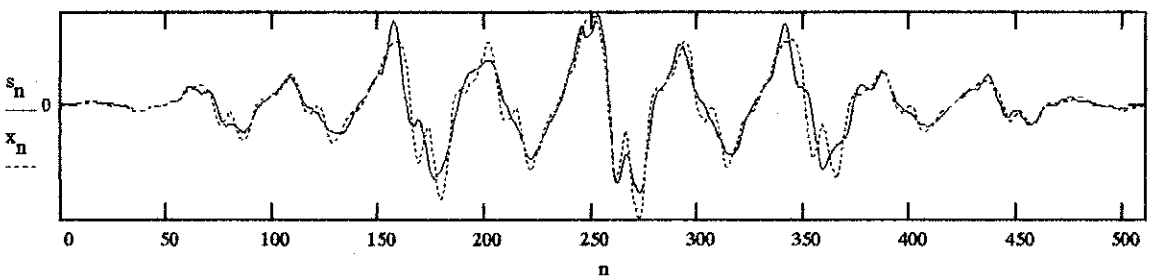


$i := 1..p \qquad n := 1..511$

$s_0 := e_0 \qquad s_n := e_n - \left(\sum_i \alpha_i \cdot if(n - i \geq 0, s_{n-i}, 0)\right)$

$n := 0..511$

Sintesis de la señal con el error filtrado y submuestreado por 8



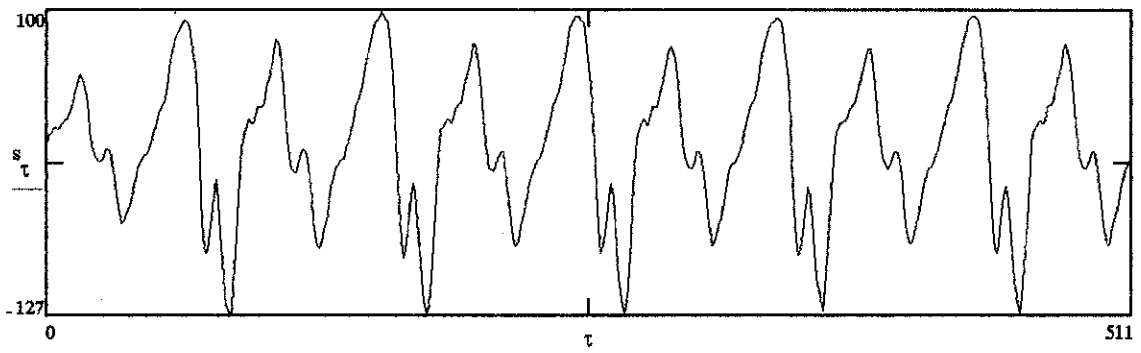
Obsérvese cómo, de las 512 muestras originales, se puede reproducir el segmento con suficiente exactitud a partir de 64 + 12 = 76 muestras

6. Análisis cepstral - Cepstro real

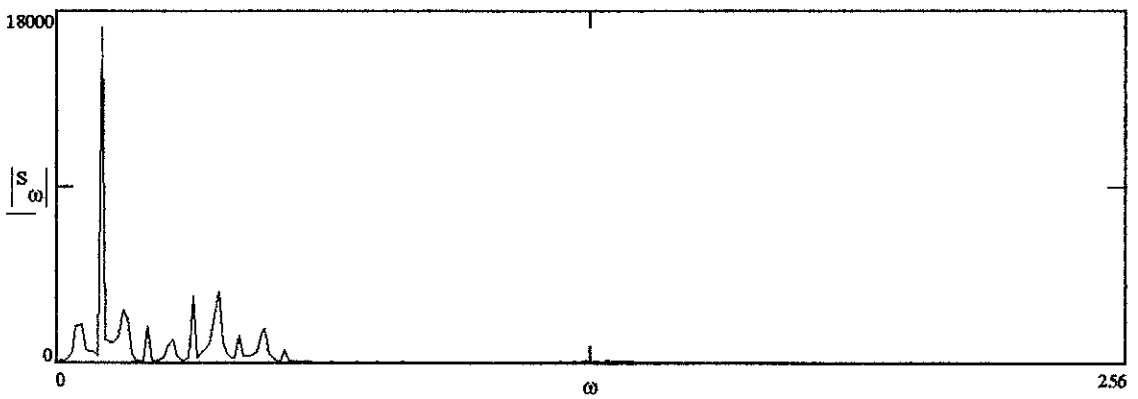
$s := READPRN(a)$ **Muestras de la señal de voz**

$L := size(s) + 1$ **Número de muestras**

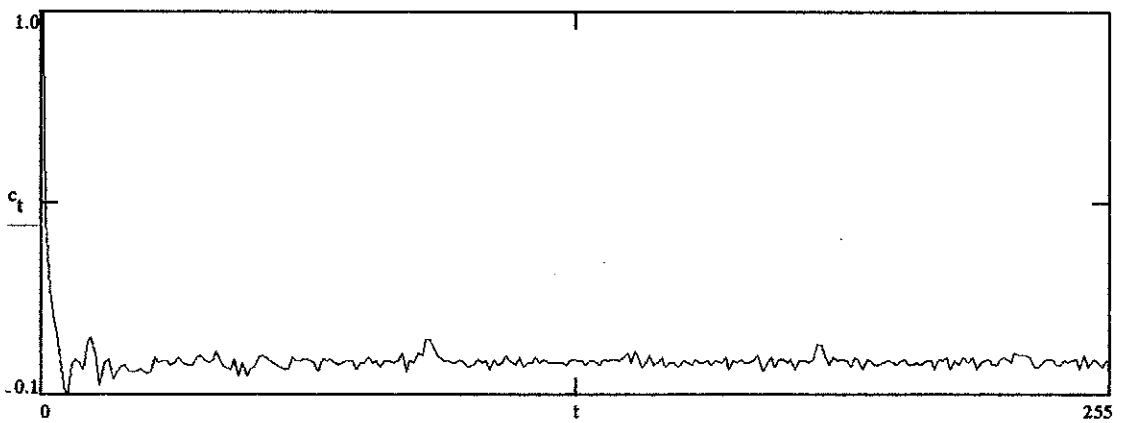
$\tau := 0 .. L - 1$ **Indice de tiempo**



$S := fft(s) \cdot \sqrt{L}$ $\omega := 0 .. \frac{L}{2}$ **Espectro de la señal original**

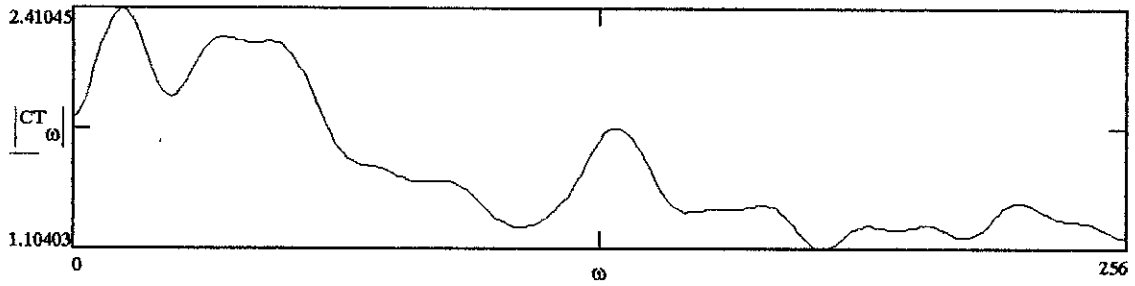


$S_\omega := \log(|S_\omega|)$ $c := \frac{ifft(S)}{\sqrt{L}}$ **Cepstro real de la señal**



$ct := c \quad i := 30..511 \quad ct_i := 0$ "Liftraje" para separar la envolvente espectral

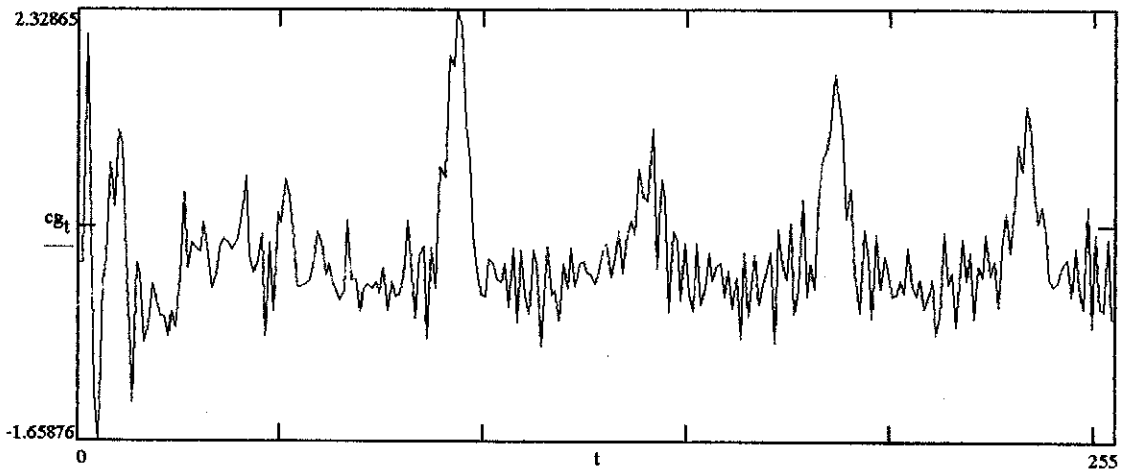
$CT := fft(ct) \cdot \sqrt{L}$ Respuesta frecuencia del tracto oral. Compárese con la obtenida mediante LPC en la sección 5.



$cg := c \quad i := 0..72 \quad cg_i := 0$ "Liftraje" para separar la excitación glotal

$CG := fft(cg) \cdot \sqrt{L}$

$CG_\omega := \exp(|CG_\omega|) \quad cg := ifft(CG)$ Estimación de la excitación glotal



El pulso de periodicidad aparece cada 93 muestras, indicando una frecuencia fundamental de 11025/93 = 118.5 Hz

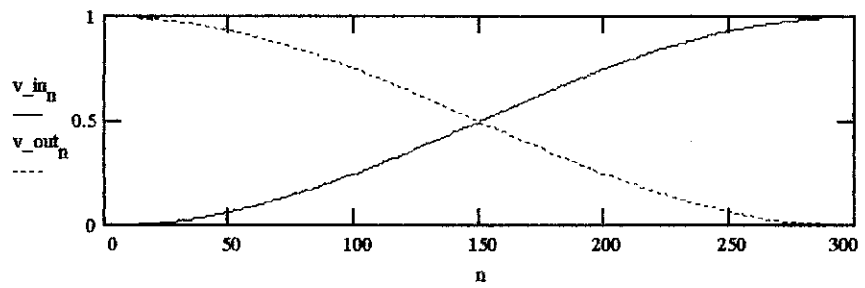
8. Síntesis de voz mediante concatenación de fonemas

$n := 0..300$

$v\_out_n := 0.5 + 0.5 \cdot \cos\left(\frac{\pi \cdot n}{300}\right)$

$v\_in_n := 1 - v\_out_n$

La transición entre fonemas se hará mediante la superposición de los extremos de los fonemas, ponderados por estas ventanas.



$y := READPRN(ss)$ Lee el fonema /S/

$x := y$ Lo "concatena" con el anterior

$y := READPRN(ii)$ Lee el siguiente fonema

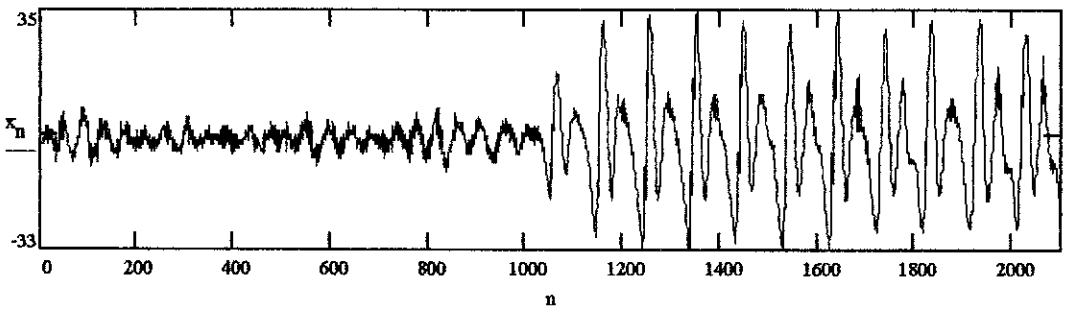
$n := 900..1200$ Región de transición entre fonemas

$x_n := x_n \cdot v\_out_{n-900} + y_{n-900} \cdot v\_in_{n-900}$ Paso de un fonema a otro

$n := 1200..2100$ Resto del fonema "entrante"

$x_n := y_{n-900}$

$n := 0..2100$



Repite el proceso para los demás fonemas:

$y := READPRN(nn)$ $n := 1800..2100$ $x_n := x_n \cdot v\_out_{n-1800} + y_{n-1800} \cdot v\_in_{n-1800}$

$n := 2100..3000$ $x_n := y_{n-2100}$

$y := READPRN(tt)$ $n := 2700..3000$ $x_n := x_n \cdot v\_out_{n-2700} + y_{n-2700} \cdot v\_in_{n-2700}$

$n := 3000..3900$ $x_n := y_{n-3000}$

$y := READPRN(ee)$ $n := 3600..3900$ $x_n := x_n \cdot v\_out_{n-3600} + y_{n-3600} \cdot v\_in_{n-3600}$

$n := 3900..4800$ $x_n := y_{n-3900}$

$y := READPRN(ss)$ $n := 4500..4800$ $x_n := x_n \cdot v\_out_{n-4500} + y_{n-4500} \cdot v\_in_{n-4500}$

$n := 4800..5700$ $x_n := y_{n-4800}$

$y := READPRN(ii)$ $n := 5400 .. 5700$ $x_n := x_n \cdot v\_out_{n-5400} + y_{n-5400} \cdot v\_in_{n-5400}$

$n := 5700 .. 6600$ $x_n := y_{n-5700}$

$y := READPRN(ss)$ $n := 6300 .. 6600$ $x_n := x_n \cdot v\_out_{n-6300} + y_{n-6300} \cdot v\_in_{n-6300}$

$n := 6300 .. 7500$ $x_n := y_{n-6300}$

Ahora hace la "concatenación" con supuestos silencios al comienzo y al final de la palabra

$n := 0 .. 300$ $x_n := x_n \cdot v\_in_n$

$n := 7200 .. 7500$ $x_n := x_n \cdot v\_out_{n-7200}$

Y, por último, pone el acento en la primera sílaba

$n := 0 .. 1000$

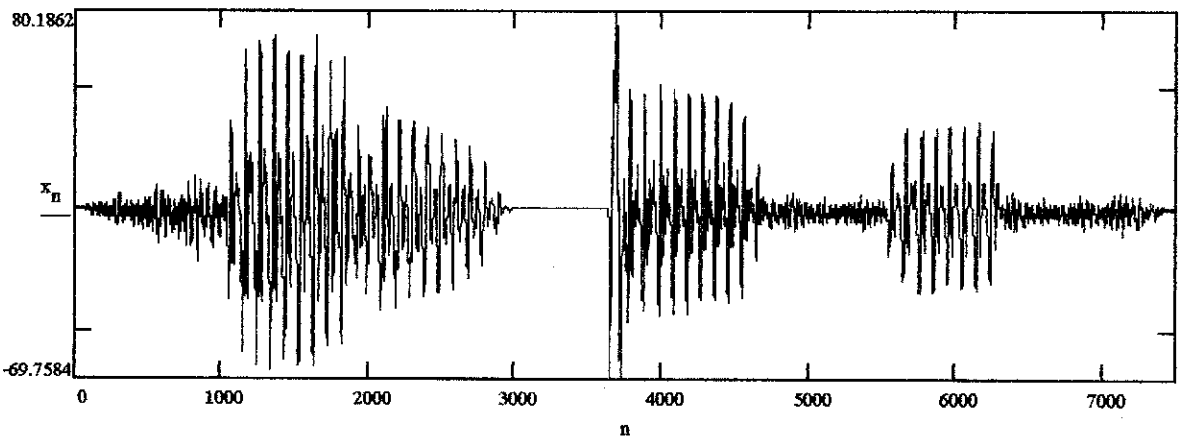
$$x_n := x_n \cdot \left(1.5 - 0.5 \cdot \cos\left(\frac{\pi \cdot n}{1000}\right)\right)$$

$$x_{n+1000} := 2 \cdot x_{n+1000}$$

$$x_{n+2000} := x_{n+2000} \cdot \left(1.5 + 0.5 \cdot \cos\left(\frac{\pi \cdot n}{1000}\right)\right)$$

$n := 0 .. 7500$

Forma de onda de la palabra /SINTESIS/ sintetizada



$WRITEPRN(sintesis) := x$ Se almacena en un archivo .PRN para convertirlo a .WAV y escucharlo mediante una tarjeta de audio

9. Alineación Dinámica de Tiempo -DTW-

$a := READPRN(cero1)$ Señal "patrón"

$b := READPRN(cero2)$ Señal de prueba

$I := floor\left(\frac{size(a)+1}{128} - 1\right)$ Número de tramas en la señal patrón $I = 49$

$J := floor\left(\frac{size(b)+1}{128} - 1\right)$ Número de tramas en la señal de prueba $J = 36$

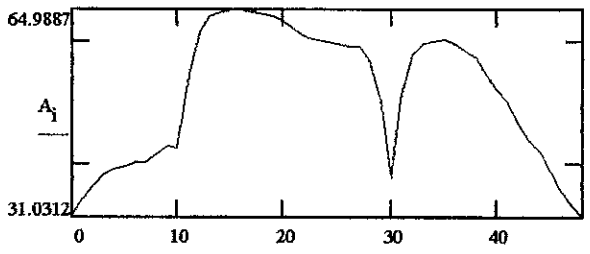
$i := 0..I-1$ Indice de la Función de Energia en Tiempo Corto (patrón)

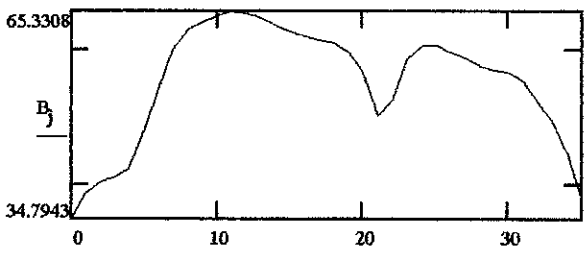
$j := 0..J-1$ Indice de la Función de Energia en Tiempo Corto (prueba)

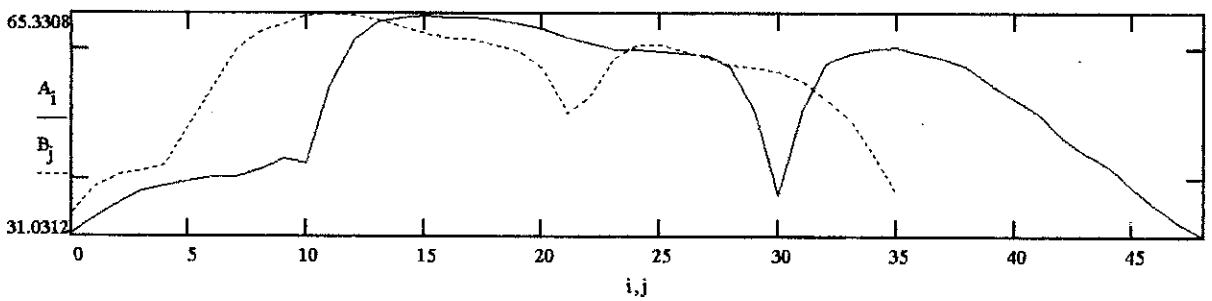
$n := 0..255$ Indice de las muestras de cada trama

$A_i := \sum_n \left(a_{128\cdot i+n}\right)^2$ $A_i := 10\cdot\log(A_i)$ Función de Energía en Tiempo Corto en dB (patrón)

$B_j := \sum_n \left(b_{128\cdot j+n}\right)^2$ $B_j := 10\cdot\log(B_j)$ Función de Energía en Tiempo Corto en dB (prueba)



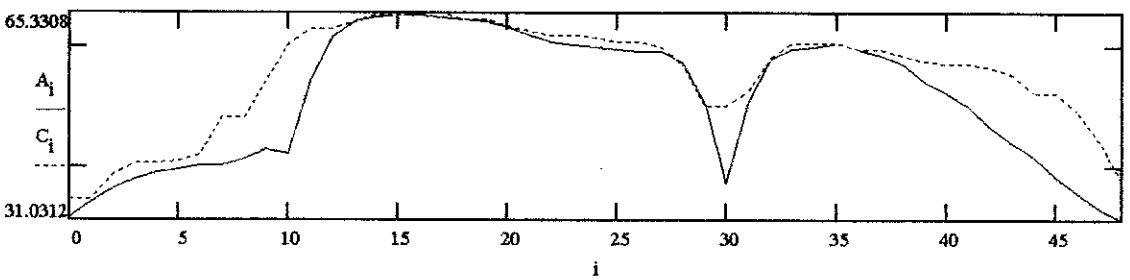




Aunque ambas corresponden a la función de energía de la palabra /cero/, sería absurdo calcular el error entre ellas, muestra a muestra, sin hacer antes alguna alineación:

$x_i := floor\left(i\cdot\frac{J-1}{I-1}\right)$ Primero se intenta una alineación lineal y se observa como ahora los patrones se hacen más "comparables". Sin embargo, en algunos puntos, como i=10 e i=45, el error sigue siendo innecesariamente grande.

$C_i := B_{(x_i)}$



$E2 := \frac{1}{I}\cdot\sum_i \left(A_i - C_i\right)^2$ $E2 = 36.34302$

El método de alineación escogido es, entonces, aquel que optimiza el error promedio entre el patrón y la prueba. La mejor técnica de optimización es la Programación Dinámica:

$d_{i,j} := (A_i - B_j)^2$ Se asigna una "distancia" (la diferencia de Energías) a cada punto (i,j) y del estudio de esta matriz se encuentra la siguiente "ruta óptima":

$\min2(a,b) := \text{if}(a<b,a,b)$

$\min4(a,b,c,d) := \min2(\min2(a,b),\min2(c,d))$

$\min5(a,b,c,d,e) := \min2(\min4(a,b,c,d),e)$

$f(a,b,c,d,e) := \text{if}(\min5(a,b,c,d,e)=a,0,\text{if}(\min5(a,b,c,d,e)=b,0,\text{if}(\min5(a,b,c,d,e)=d,1,\text{if}(\min5(a,b,c,d,e)=c,1,2))))$

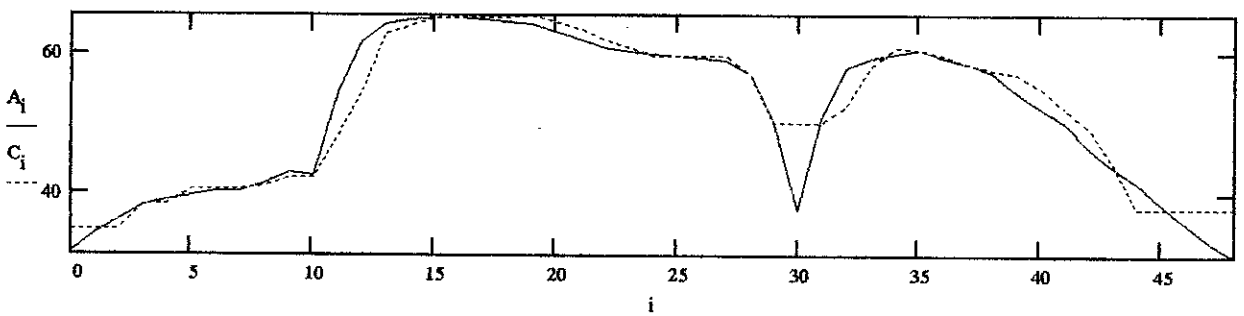
$x_0 := 0 \qquad n := 1..I-6$

$x_n := x_{n-1} + f\left(d_{n,x_{n-1}}, d_{n+1,x_{n-1}}, d_{n-1,x_{n-1}+1}, d_{n,x_{n-1}+1}, d_{n-1,x_{n-1}+2}\right)$

$j := I-5..I-1$

$x_j := J-1$

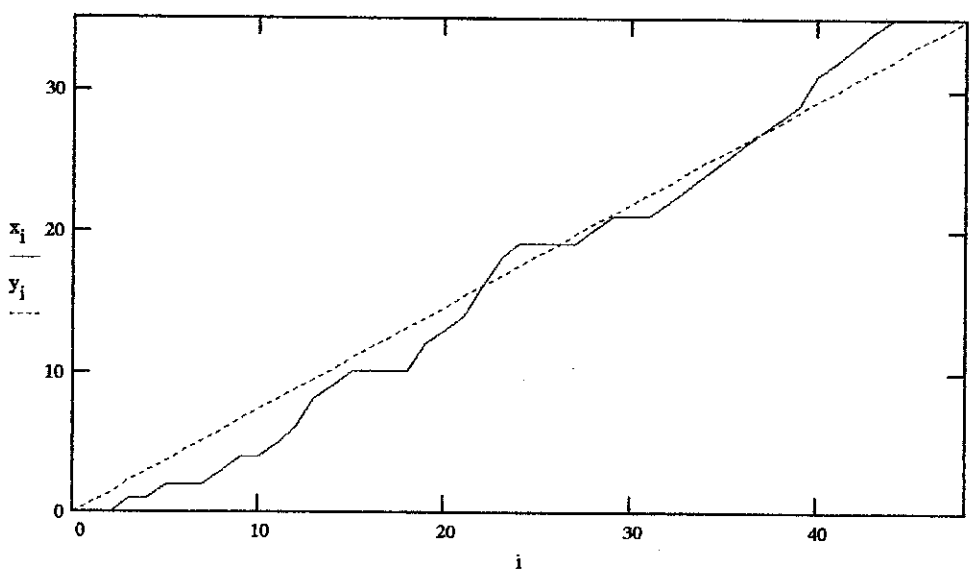
$C_i := B_{(x_j)}$



$E2 := \frac{1}{I}\cdot\sum_i (A_i - C_i)^2$ $E2 = 8.60524$ Como se esperaba, la técnica DTW permite un error promedio mucho menor que la compresión lineal

---

$y_i := i\cdot\frac{J-1}{I-1}$



Función de alineación óptima para estas dos funciones de energía de la palabra /CERO/