3. CASOS PRÁCTICOS.

En esta sección se va a abordar la implementación práctica de los esquemas estudiados en el apartado teórico. Ante los aspectos genéricos de la teoría de los codificadores se concretarán las soluciones, aproximaciones y parámetros.

Se ha trabajado con ventanas cuadradas de 30 milisegundos para el codificador LPC (en el que se ha considerado solapamiento entre tramas) y el codificador CELP (para ajustarnos al modelo genérico de 7.5 milisegundos por subtrama), para el resto de casos prácticos se han considerado ventanas de 20 milisegundos.

La asignación de bits y la tasa de codificador se describirán en cada apartado por separado.

3.1. CODIFICADOR BASADO EN LA DCT.

Recordemos del apartado teórico que este codificador usa la DCT tratando de aprovechar la propiedad de concentrar las componentes más importantes de la voz en aquellos coeficientes con mayor amplitud.

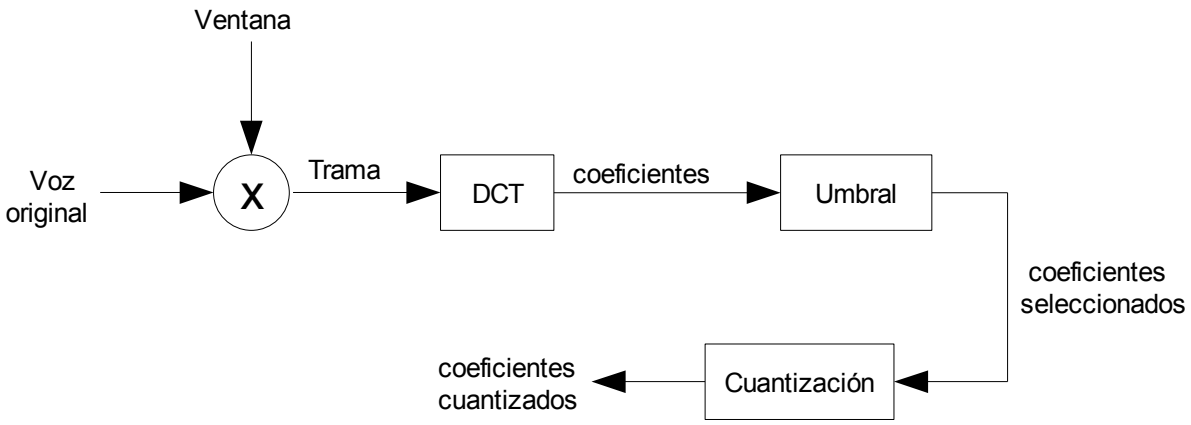


Figura 40: Codificador DCT

En el proceso de análisis para cada trama se han calculado sus coeficientes DCT, y se han seleccionado aquellos cuya magnitud en valor absoluto decae un porcentaje determinado respecto al coeficiente de mayor amplitud de entre todas las tramas. Este porcentaje es un parámetro libre en principio. Cuanto mayor sea el porcentaje, mayor número de coeficientes seleccionará el esquema, obteniendo una reconstrucción más fiel. Pero también aumentará el número de parámetros enviados al decodificador, o lo que es lo mismo, la tasa del codificador. Asimismo, incluso habiendo fijado dicho porcentaje, el número de coeficientes que lo superan es incierto a priori, llevándonos a una tasa de codificador variable.

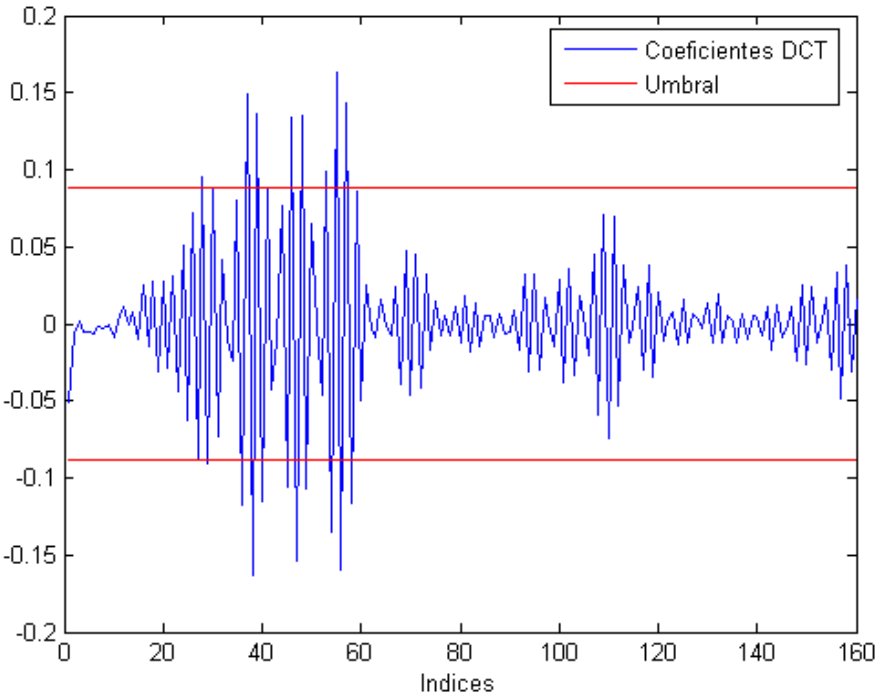


Figura 41: Coeficientes de la DCT y el umbral propuesto

El proceso de síntesis consiste simplemente en antitransformar los coeficientes seleccionados en el codificador y concatenar las tramas resultantes, obteniendo la señal sintetizada.

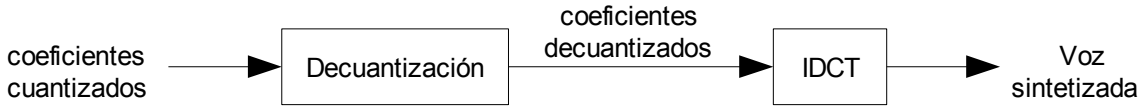


Figura 42: Decodificador DCT

En los experimentos se ha seleccionado un umbral del 3%. Para codificar los coeficientes DCT se han asignado 12 bits, para los índices 6 bits, lo que nos lleva a la siguiente tasa de salida:

$$Tasa = (12+6)\frac{bits}{punto} n \frac{puntos}{trama} 1 \frac{trama}{160\,muestras} 8000 \frac{muestras}{s} = 900\mathrm{n}\frac{bits}{s}$$

siendo *n* el número medio de puntos seleccionados por trama.

3.2. CODIFICADOR ARMÓNICO.

El codificador armónico está basado en la aplicación directa de la DFT y la posterior selección de un conjunto de tripletas {Amplitud, Frecuencia, Fase} para cada trama.

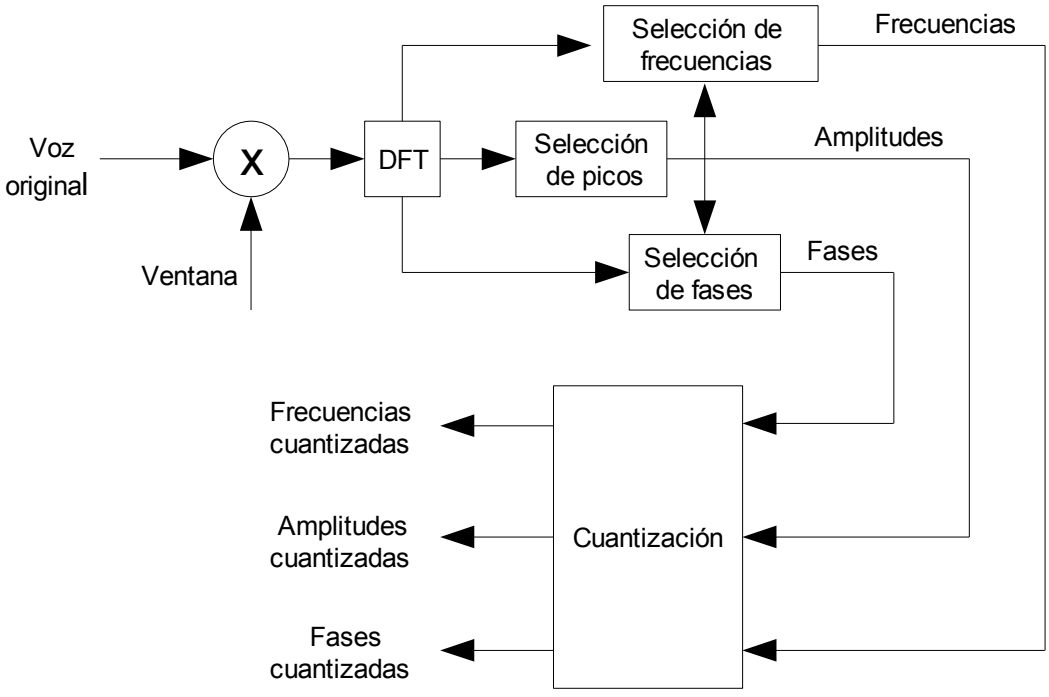


Figura 43: Codificador Armónico

En el proceso de análisis se fijará qué criterio tomar para escoger dichas tripletas:

1. Selección de picos de mayor magnitud. Sobre la magnitud espectro se escogen un número limitado de amplitudes. Es inmediato obtener sus frecuencias y, a partir de éstas, sus correspondientes fases. A pesar de ser el método más simple, es el que proporciona la mejor reconstrucción y, por tanto, la mejor medida señal-ruido. Pero como veremos más adelante, esta medida no es definitiva y este criterio no es el mejor.

2. Selección de "formantes". Este método trata de captar los formantes, las frecuencias de resonancia del tracto vocal. Para ello se ha dividido el espectro en tramos de cuatro muestras. Entre tramos correlativos se ha calculado la diferencia de energía. Por cada segmento entre aquellos en los que la diferencia de energía es mayor, se ha seleccionado una amplitud y, a partir de ella, su frecuencia y su fase. De esta manera no nos quedamos solamente con las mayores amplitudes, que pueden estar concentradas en torno a un máximo, sino que "captamos" la forma del espectro. Este método tiene en cuenta las características de la fuente (la voz humana) y, a pesar de que la medida de calidad no es tan buena como el método anterior, perceptualmente los resultados son mejores. Se ha optado por esta elección.

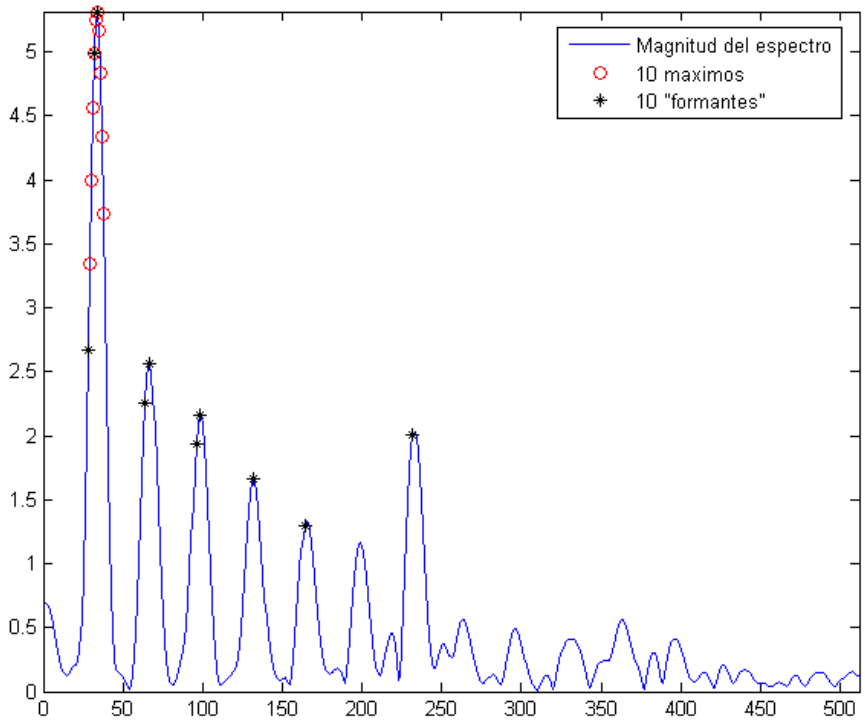


Figura 44: Espectro de la DFT y amplitudes seleccionadas

El proceso de síntesis no es el proceso inverso a la DFT. En el decodificador aproximaremos la señal original como la suma de las senoides generadas a partir de las tripletas procedentes del análisis.

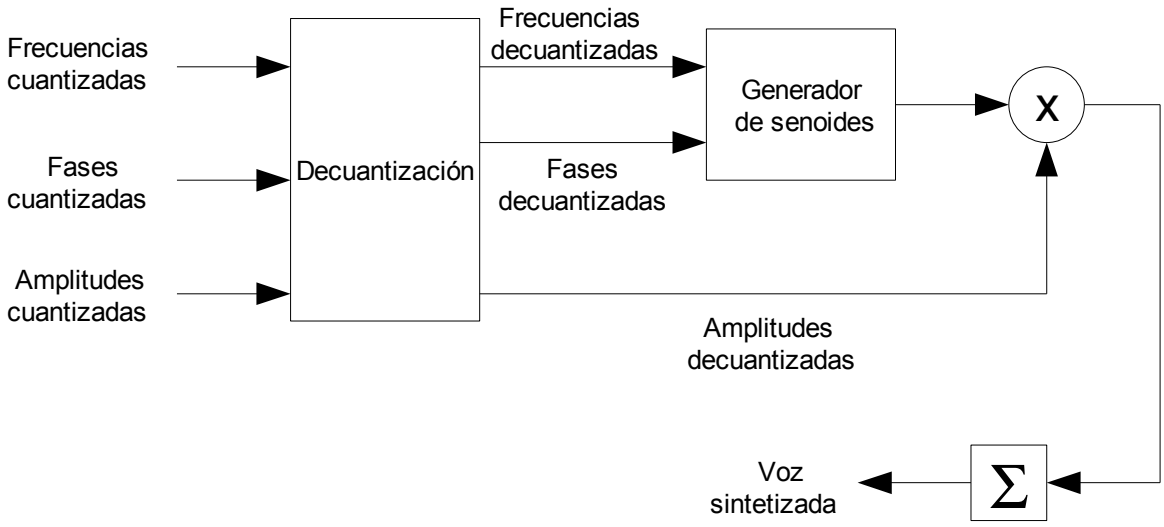


Figura 45: Decodificador Armónico

En los experimentos se han seleccionado 20 tripletas por trama. Para la codificación de la amplitud se han asignado 10 bits, para la frecuencia 16 bits y para la fase 6 bits, lo que nos lleva a la siguiente tasa de salida:

$$Tasa = (10+16+6)\frac{bits}{tripleta}\,20\,\frac{tripletas}{trama}\,1\,\frac{trama}{160\,muestras}\,8000\,\frac{muestras}{s} = 32000\,\frac{bits}{s}$$

3.3. CODIFICADOR ADPCM.

El codificador ADPCM es un codificador de propósito general. En principio no posee ninguna propiedad que le haga especialmente adecuado para el tratamiento de la voz. El codificador realiza operaciones de diferencias entre muestras correlativas, así que supondremos que la diferencia entre muestras será un valor cercano a cero que podremos codificar con baja precisión.

En la práctica se ha optado por un esquema que integra una parte de ganancia adaptativa hacia adelante y otra diferencial de modelo autoregresivo. El proceso de análisis segmenta la voz en tramas. El objeto de ello es el cálculo de una ganancia coherente para un conjunto de muestras en un lapso corto de tiempo. La ganancia de la trama se ha calculado de la siguiente manera:

$$g = max(v_w|n|) + 0.001$$

Cada trama se normaliza con dicho parámetro. Al resultado se le aplica un esquema diferencial AR con un predictor simple de retraso de una muestra.

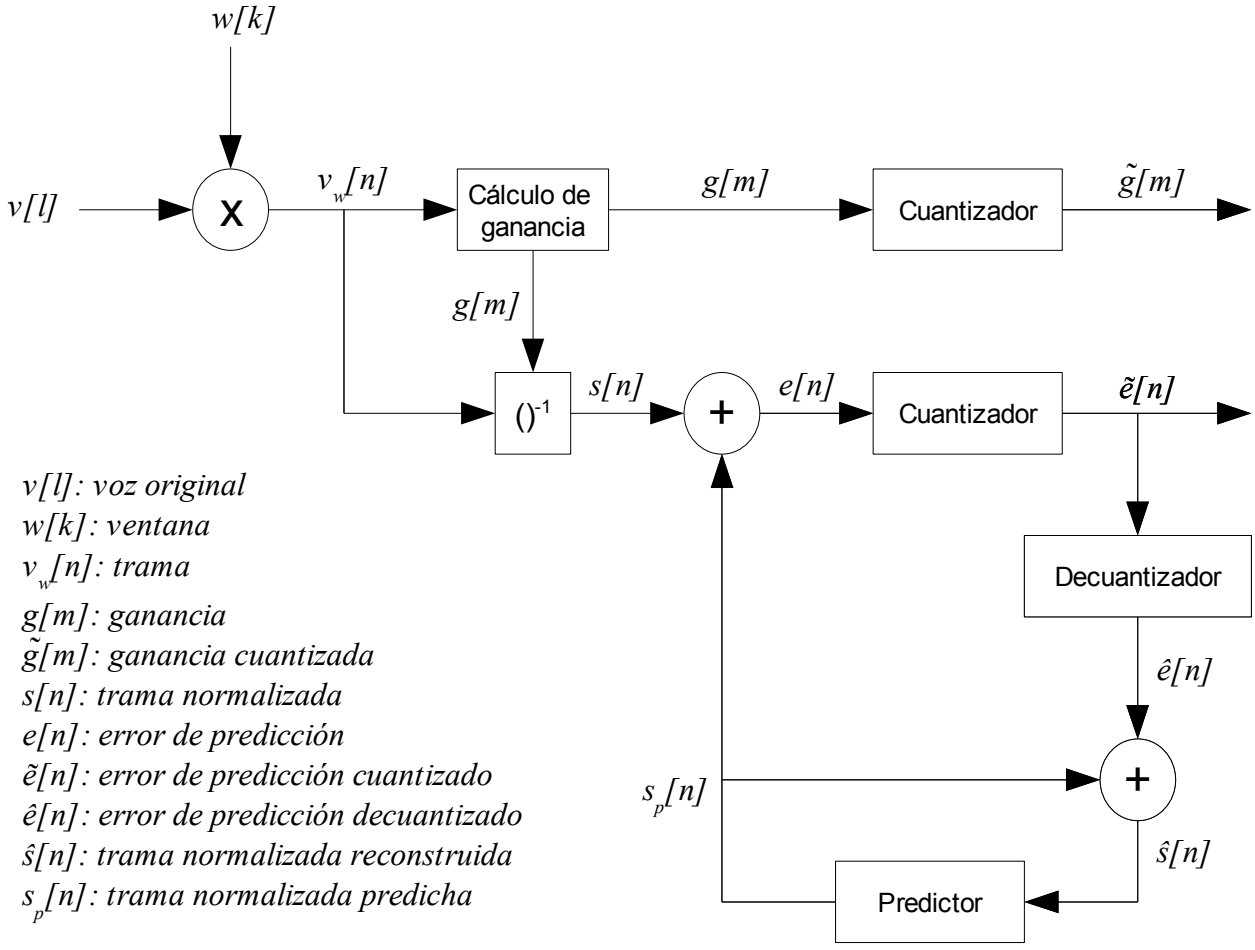
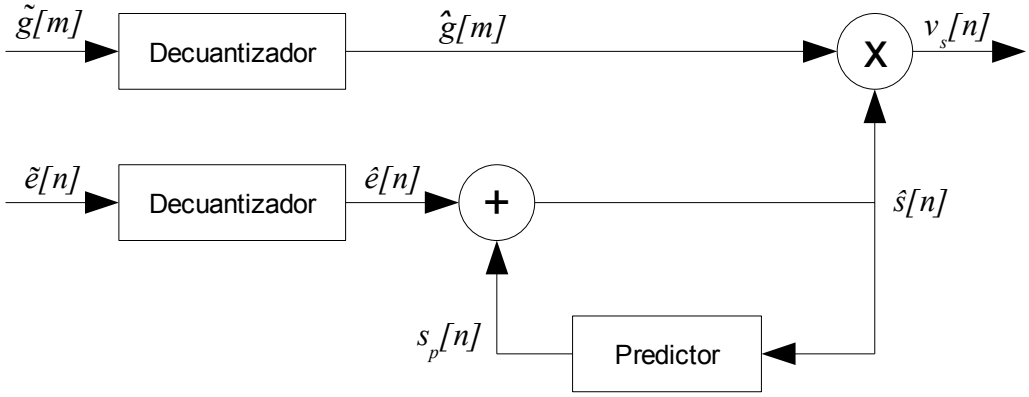


Figura 46: Codificador ADPCM

Durante el proceso de síntesis se recupera una estimación de la trama normalizada, siguiendo un proceso similar al realizado en el codificador y, posteriormente, el resultado se escala con la ganancia decuantizada.



g[m]: ganancia
$\hat{g}$[m]: ganancia decuantizada
$\tilde{e}$[n]: error de predicción cuantizado
$\hat{e}$[n]: error de predicción decuantizado
$\hat{s}$[n]: trama normalizada reconstruida
s_p[n]: trama normalizada predicha
v_s[n]: trama sintetizada

Figura 47: Decodificador ADPCM

Para codificar la ganancia se han asignado 7 bits y para cada muestra del error 7 bits, lo que nos lleva a la siguiente tasa de salida:

$$Tasa = 7\frac{bits}{ganancia}1\frac{ganancia}{trama}1\frac{trama}{160\,muestras}8000\frac{muestras}{s} + \frac{7\text{bits}}{muestra}8000\frac{muestras}{s} = 56350\frac{bits}{s}$$

3.4. CODIFICADOR LPC.

El proceso de análisis se realiza exactamente como comentamos en la teoría. Para cada trama se estiman los coeficientes LPC buscando minimizar el error. Desde el punto de vista computacional es costoso invertir matrices, así que usamos el algoritmo de Levinson-Durbin, que nos proporcionará tanto los coeficientes LPC como la ganancia. Posteriormente, el segmento de voz se pasa por un filtro construido con los coeficientes LPC, obteniéndose el error de prediccion corta.

Sobre el error resultante se realiza la búsqueda de la frecuencia fundamental o pitch.

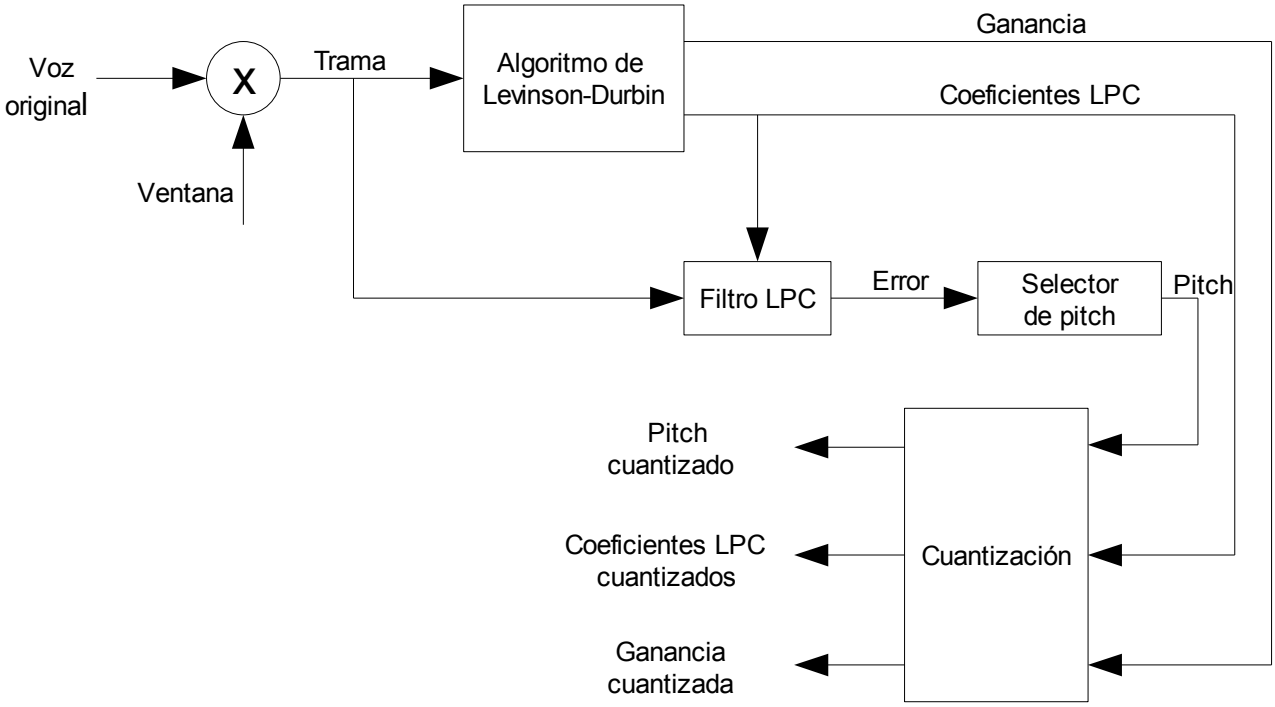


Figura 48: Codificador LPC

En la práctica se ha usado otra forma para la búsqueda del pitch distinta a la de la teoría, el método de la autocorrelación. En este método se realiza la correlación del segmento de análisis actual con un barrido de segmentos desplazados respecto a él una diferencia temporal entre 20 y 147 muestras, rango en el que supondremos que se encuentra el pitch. De entre todos los barridos, el que proporcione una mayor autocorrelación será del que se tome su desplazamiento como periodo de pitch. El pseudocódigo de este algoritmo es el siguiente:

Consideramos m como el punto inicial de la trama y N el número de muestras.

```
1. pico = 0
2. Para p desde 20 hasta 147
3.         autocorrelacion  = 0
4.         Para n desde m + 1 hasta m+N
5.                 autocorrelacion = autocorrelacion + s[n]s[n-p]
6.         Si autocorrelacion > pico
7.                 pico = autocorrelacion
```

8. *pitch = p*
9. *Devolver pitch*

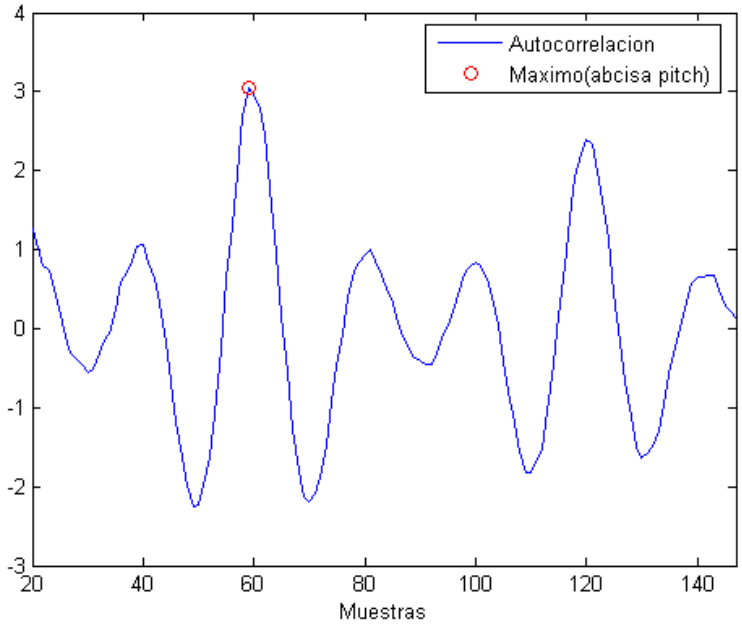


Figura 49: Selección del periodo de pitch

El proceso de síntesis varía notablemente de la teoría. Se ha tenido en cuenta el pitch estimado en el análisis de tal forma que, según su valor, se ha usado como señal de excitación bien una señal periódica de periodo el de pitch o bien un ruido gaussiano, obviando el filtro de síntesis de pitch descrito en la teoría.

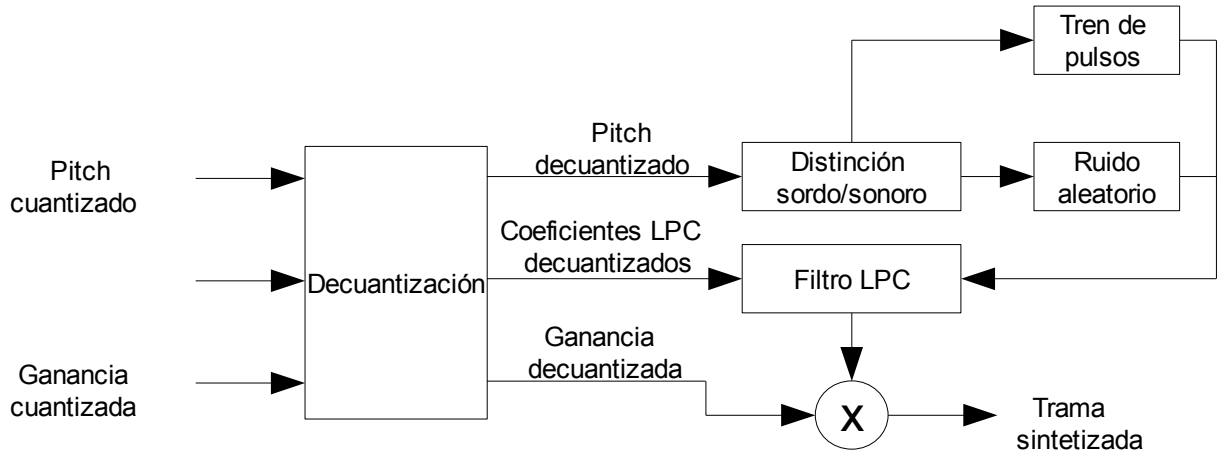


Figura 50: Decodificador LPC

Se ha considerando los pitch por debajo de 40 muestras como correspondientes a sonidos sordos y, por encima de dicho límite, a sonidos sonoros. Posteriormente, la excitación (sea cual fuere) escalada convenientemente por la ganancia, sirve de entrada para el filtro constituido por los coeficientes LPC, obteniendo finalmente la señal sintetizada.

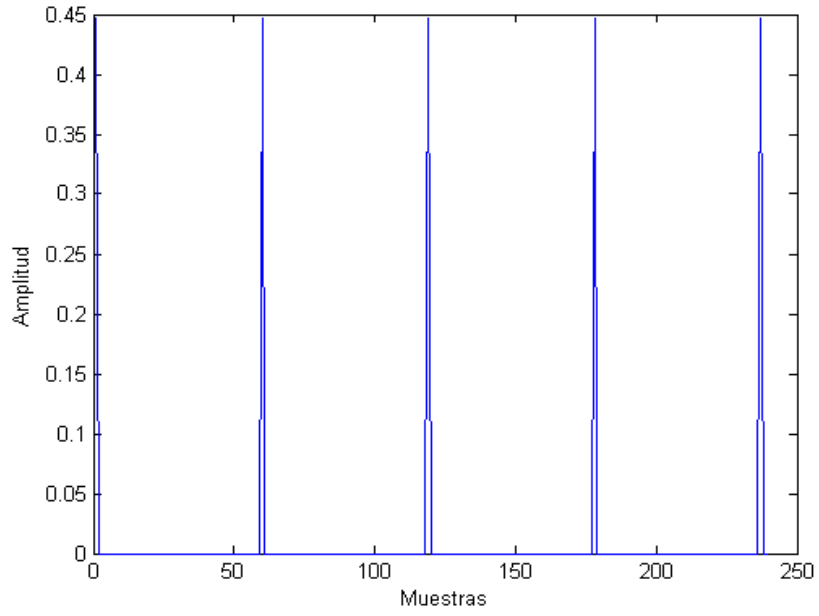


Figura 51: Señal de excitación correspondiente a un tramo sonoro

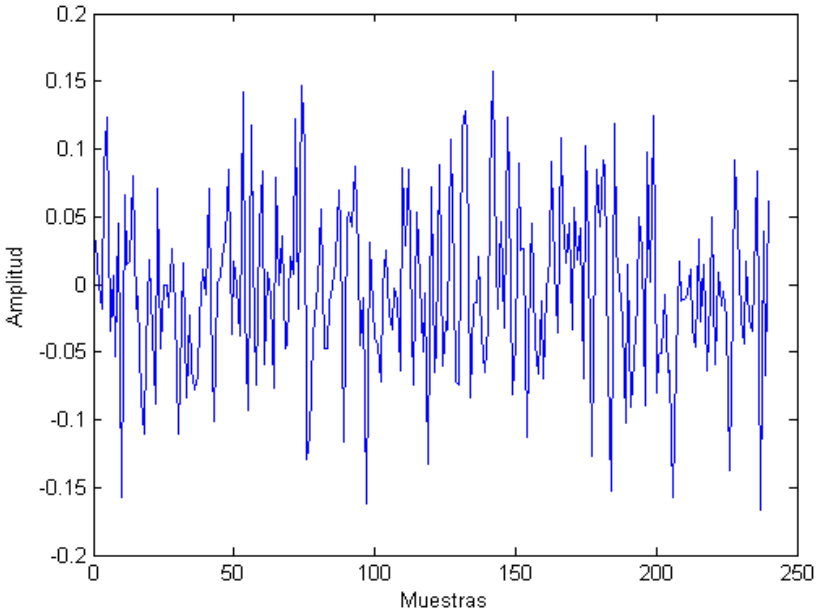


Figura 52: Señal de excitación correspondiente a un tramo sordo

Para codificar los coeficientes LPC se han asignado 10 bits, para el periodo de pitch 9 bits y para la ganancia 5 bits, lo que nos lleva a la siguiente tasa de salida:

$$Tasa = (11\text{x}10+9+5)\frac{bits}{trama}1\frac{trama}{240\,muestras}8000\frac{muestras}{s} = 4133\frac{bits}{s}$$

3.5. CODIFICADOR CELP.

El codificador CELP es un codificador de tipo híbrido, basado en una técnica de "análisis mediante síntesis". Una parte del análisis está dedicada a la extracción de parámetros de modelado de la voz, y otra parte a imitar la forma de onda.

Comentaremos las particularidades de cada bloque por separado.

3.5.1. ANÁLISIS.

Análisis LP corto.

Este bloque se ha implementado directamente con un la orden *lpc* de MATLAB, fijando un orden de 10.

Filtro de error de predicción.

Es otra aplicación directa de una función de MATLAB, en este caso la orden *filter*, usando para definir el filtro los coeficientes LPC obtenidos anteriormente. Este filtro se le aplica a la subtrama de voz, obteniéndose un residuo sobre el que se realizará el Análisis LP largo. Presenta la siguiente función de transferencia:

$$A(z) = 1 + \sum_{i=1}^{L} a_i z^{-i}$$

Análisis LP largo.

Este análisis supone que existe correlación entre muestras de voz no correlativas separadas entre sí por un intervalo de tiempo conocido como periodo de pitch.

$$\hat{e}_s = -be_s[n-T]$$

Este bloque realiza los cálculos necesarios para obtener los parámetros *b* y *T* para cada subtrama. En el intervalo de interés (en la práctica se ha considerado un pitch de entre 20 y 147 muestras) utilizaremos una medida de error cuadrático medio:

$$J = \sum_n (e_s[n] - \hat{e_s}[n])^2 = \sum_{(n)} (e_s[n] + be_s[n-T])^2$$

Derivando la ecuación anterior con respecto a b e igualando a cero:

$$b = -\frac{\sum_n e_s[n]e_s[n-T]}{\sum_n e_s^2[n-T]}$$

El pseudocódigo del algoritmo quedaría de la siguiente manera:

1. $J_{min} = \infty$
2. *Para* $T = T_{min}$ *hasta* T_{max}
3. *Calcular b*
4. *Calcular J*
5. *Si* $J < Jmin$
6. $J_{min} = J$
7. $b_{opt} = b$
8. $T_{opt} = T$
9. *Devolver* b_{opt}, T_{opt}

Filtro perceptual.

Este filtro esta conformado a partir de los coeficientes LPC hallados en el Análisis LP corto. En principio, en la teoría, se consideró una función de transferencia como la siguiente:

$$W_1(z)=\frac{A(z)}{A(z/\gamma_1)}=\frac{1+\sum_{i=1}^{L} a_i z^{-i}}{1+\sum_{i=1}^{L} a_i \gamma_1^i z^{-i}},\ \gamma_!=0.9$$

Experimentalmente se ha comprobado que los resultados obtenidos son mejores usando un filtro perceptual de la forma:

$$W_2(z)=\frac{A(z/\gamma_2)}{A(z/\gamma_1)}=\frac{1+\sum_{i=1}^{L} a_i \gamma_2^i z^{-i}}{1+\sum_{i=1}^{L} a_i \gamma_1^i z^{-i}},\ \gamma_1=0.9, \gamma_2=0.5$$

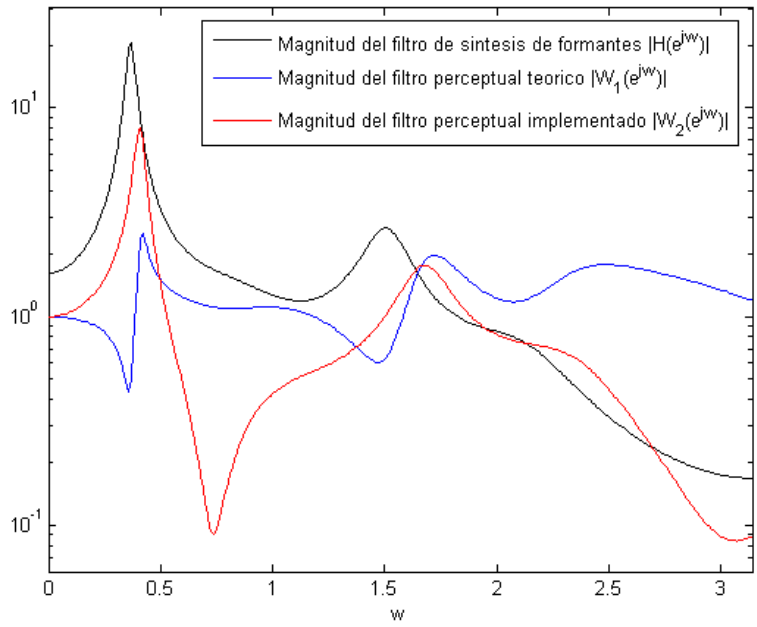


Figura 53: Filtro de síntesis de formantes y filtros perceptuales

Libro de códigos.

Para fijar el libro de códigos se han tratado dos posibilidades:

1. Usar un libro de secuencias aleatorias basándonos en una distribución gaussiana de media cero y varianza uno.

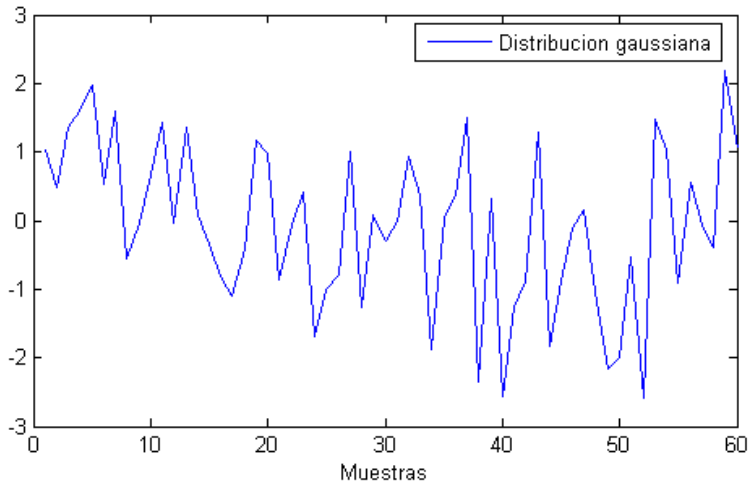


Figura 54: Secuencia de un libro de códigos aleatorio

2. Definir un libro de códigos ternario, en los que la secuencia solo puede tomar tres valores posibles: -1, 0 y 1. Para ello se ha utilizado una distribución uniforme con valores comprendidos entre 0 y 1. Después se han establecido dos umbrales en 1/3 y 2/3. A los valores comprendidos entre 0 y 1/3 se les ha asignado el valor -1, a los comprendidos entre 1/3 y 2/3 el valor 0 y, finalmente, a los comprendidos entre 2/3 y 1 el valor 1.

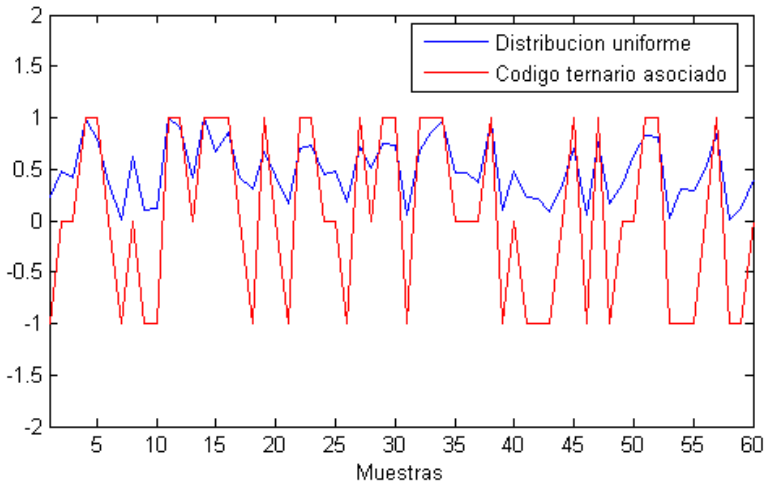


Figura 55: Secuencia de un libro de códigos ternario

En los experimentos no se ha observado una diferencia notable en el uso de uno u otro libro de códigos. Una de las ventajas del libro ternario es que, en una implementación práctica, todas las operaciones serian exclusivamente sumas, las multiplicaciones desaparecerían debido al juego de valores {-1, 0, 1}.

Cálculo de ganancia.

Para este apartado se han considerado dos posibilidades:

1. Calcular la ganancia para cada subtrama igualando la energía de la subtrama de voz original con la de la sintetizada:

$$ganancia = \sqrt{\frac{Energia_{original}}{Energia_{sintetizada}}}$$

2. Usar el algoritmo de Levinson-Durbin para calcular la ganancia de cada trama, como en el codificador LPC.

Experimentalmente se ha comprobado que la segunda opción proporciona mejores resultados.

Filtro de síntesis de pitch.

Hemos usado la orden *filter* de MATLAB con los parámetros obtenidos en el Análisis LP largo para implementar la siguiente función de transferencia:

$$H_p(z) = \frac{1}{1 + bz^{-T}}$$

Filtro modificado de síntesis de formantes.

Hemos usado la orden *filter* de MATLAB usando los parámetros obtenidos en el Análisis

LP corto, teniendo en cuenta que el filtro perceptual no coincide exactamente con el teórico, para implementar la siguiente función de transferencia:

$$H_{mf}(z)=\frac{1}{A(z)}\frac{A(z/\gamma_2)}{A(z/\gamma_1)}=\frac{1}{1+\sum_{i=1}^{L}a_i z^{-i}}\frac{1+\sum_{i=1}^{L}a_i \gamma_2^i z^{-i}}{1+\sum_{i=1}^{L}a_i \gamma_1^i z^{-i}}$$

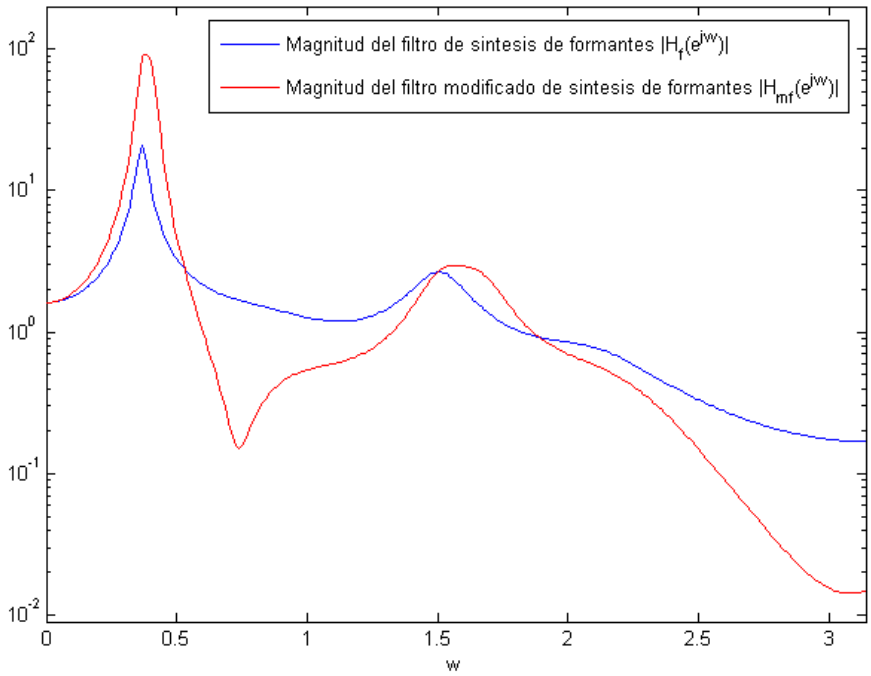


Figura 56: Filtro de síntesis de formantes y filtro modificado

Minimización del error.

Una vez fijados los parámetros del Análisis LP, sólo queda seleccionar la mejor secuencia de código. Por cada secuencia del libro se sintetiza una posible subtrama, y se compara con la subtrama original analizada mediante un criterio de error cuadrático medio:

$$D=\sum_{n}(s[n]-\hat{s}[n])^2$$

Se seleccionará aquella secuencia de código que proporcione la menor distorsión posible.

3.5.2. SÍNTESIS.

Sobre el apartado de síntesis poco hay que comentar, debido a que todos los bloques usados han sido comentados en el apartado de análisis.

En el codificador CELP se ha usado un libro de códigos de 128 secuencias de 60 muestras cada uno. Para codificar los coeficientes LPC se han asignado 12 bits , para la ganancia de predicción larga 5 bits, para el periodo de pitch 7 bits, para el índice del libro de códigos 9 bits y para la ganancia también 9 bits, lo que nos lleva a la siguiente tasa de salida:

$$Tasa = (11\text{x}12 + 9 + 4\text{x}(5 + 7 + 9))\frac{bits}{trama}\,1\frac{trama}{240\,muestras}\,8000\frac{muestras}{s} = 7500\frac{bits}{s}$$