

Práctica 2. Modelado de la señal de voz como un proceso aleatorio y su aplicación en la transmisión sobre un canal con pérdidas (Continuación parte 2 - Anexo I)

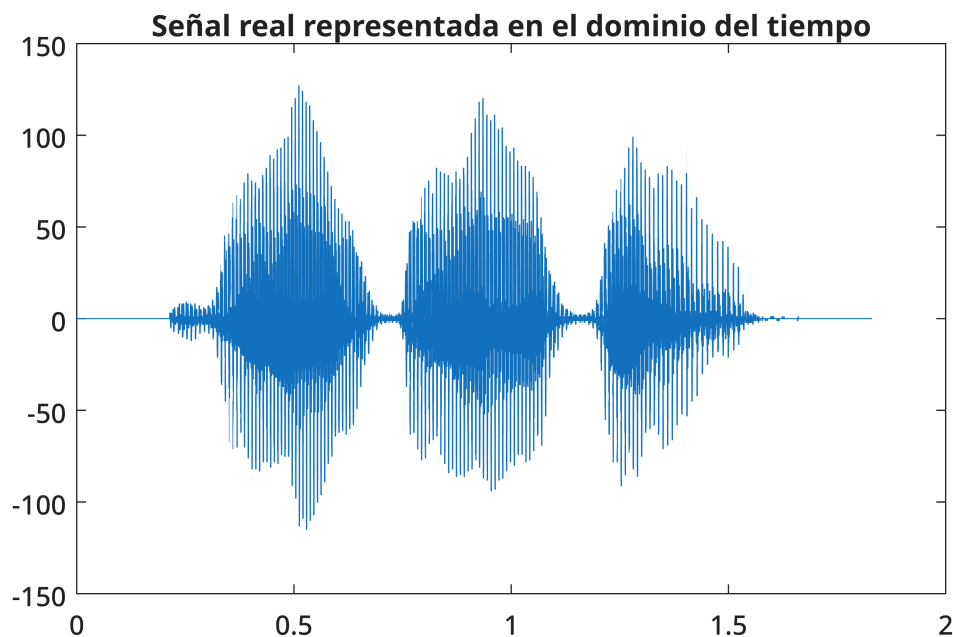
¿QUÉ PASARÍA SI PIERDO DOS MUESTRAS?

La última parte de la realización de la práctica 2 - modelo 2 nos propone este interesante reto. Partiendo de mi señal real *realSignal*, voy a insertar pérdidas de muestra tanto al principio de cada bloque de 10 muestras, como al final:

```
lossySignal2 = realSignal;  
  
for row_index = 1:10:length(lossySignal2)  
    lossySignal2(row_index, 1) = randi([1,10]);  
  
    if (row_index ~= 1)  
        % Añadimos las pérdidas al final de cada bloque de 10 muestras.  
        lossySignal2(row_index - 1, 1) = randi([1,10]);  
    end  
end
```

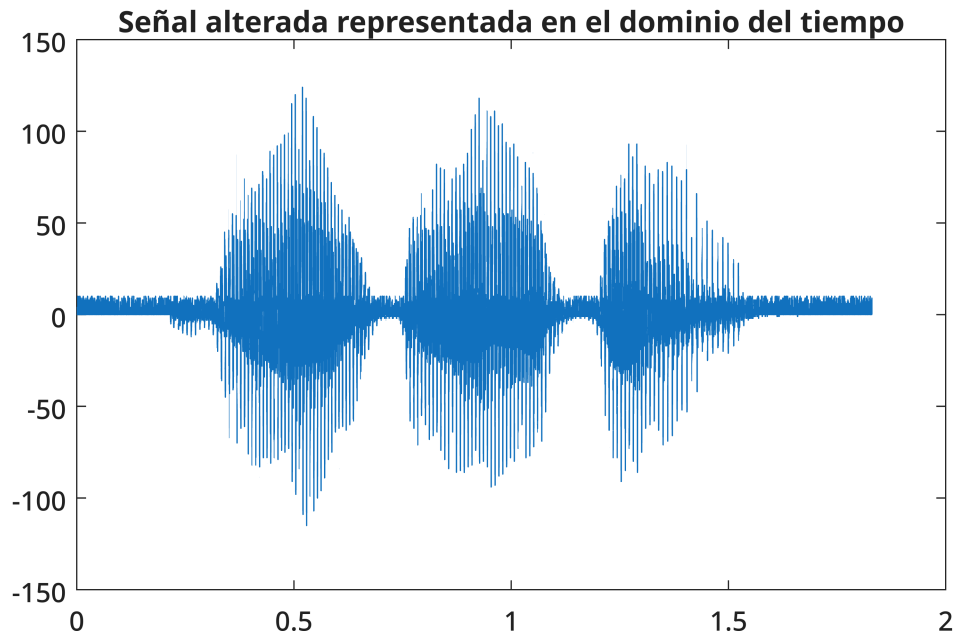
Veamos ambas señales para compararlas gráficamente:

```
figure;  
plot([1:length(realSignal)]/frequency, realSignal)  
title("Señal real representada en el dominio del tiempo")
```



```
figure;  
plot([1:length(lossySignal2)]/frequency, lossySignal2)
```

```
title("Señal alterada representada en el dominio del tiempo")
```



Ahora, para reconstruirla, vamos a usar los dos métodos de estimación que preparamos. Pero, estamos asumiendo que nuestra cadena de Markov es homogénea y de primer orden, lo que implica que, para reconstruir la segunda pérdida, haremos uso de la reconstrucción de la primera pérdida.

Esto da lugar a un nuevo riesgo, la propagación de errores. Si la primera estimación no fue muy exacta, la estimación de la segunda muestra será afectada.

Vamos a arreglar nuestra señal con pérdidas. Adaptaré el código de antes:

```
trend_reconstructedSignal_Markov = lossySignal2;
weightProbability_reconstructedSignal_Markov = lossySignal2;

for row_index = 1:10:length(lossySignal2)
    if (row_index > 2)

        % Reconstruyo la pérdida anterior (fin del bloque anterior de 10
        % muestras)
        trend_reconstructedSignal_Markov(row_index - 1, 1) =
trendProbabilityValues(round(trend_reconstructedSignal_Markov(row_index - 2)
+ 129));
        weightProbability_reconstructedSignal_Markov(row_index - 1, 1) =
weightProbabilityValues(round(weightProbability_reconstructedSignal_Markov(ro
w_index - 2) + 129));

        % Reconstruyo la pérdida actual (inicio del bloque actual de 10
        % muestras)
        trend_reconstructedSignal_Markov(row_index, 1) =
trendProbabilityValues(round(trend_reconstructedSignal_Markov(row_index - 1)
+ 129));
```

```

        weightProbability_reconstructedSignal_Markov(row_index, 1) =
weightProbabilityValues(round(weightProbability_reconstructedSignal_Markov(ro
w_index - 1) + 129));
    else
        % Hago este else para asignar un valor a la muestra número 1. Cómo
        % antes de este no hay ninguna muestra, asumo que es 0.
        trend_reconstructedSignal_Markov(row_index, 1) =
trendProbabilityValues(0 + 129);
        weightProbability_reconstructedSignal_Markov(row_index, 1) =
weightProbabilityValues(0 + 129);
    end
end

```

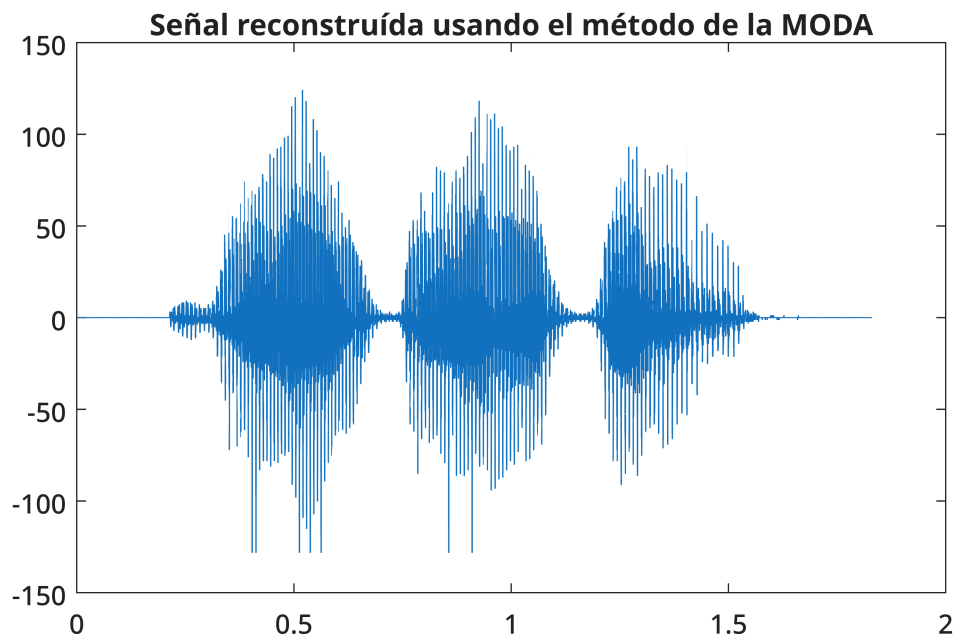
Nota: He tenido que meter la función **round** en el código, para evitar índices como estos: "129.0008".

Vamos a ver gráficamente a reconstrucción de la señal:

```

figure;
plot([1:length(trend_reconstructedSignal_Markov)]/frequency,
trend_reconstructedSignal_Markov)
title("Señal reconstruída usando el método de la MODA")

```

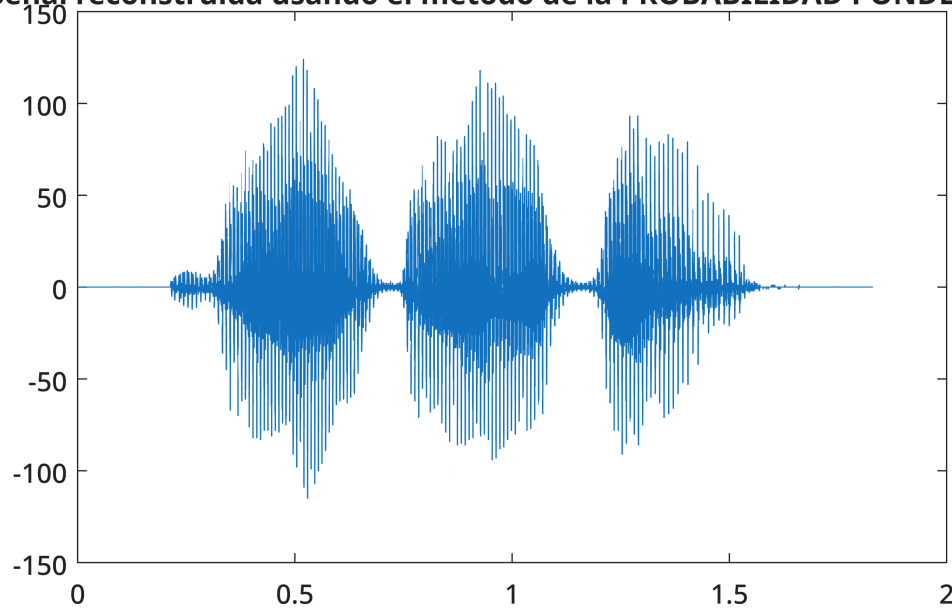


```

figure;
plot([1:length(weightProbability_reconstructedSignal_Markov)]/frequency,
weightProbability_reconstructedSignal_Markov)
title("Señal reconstruída usando el método de la PROBABILIDAD PONDERADA")

```

Señal reconstruída usando el método de la PROBABILIDAD PONDERADA



Y calculemos las respectivas SNR:

```
SNR1 = 10*log10(((realSignal'*realSignal))/  
((realSignal-trend_reconstructedSignal_Markov)'*(realSignal-  
trend_reconstructedSignal_Markov)))
```

```
SNR1 =  
10.4511
```

```
SNR2 = 10*log10(((realSignal'*realSignal))/  
((realSignal-weightProbability_reconstructedSignal_Markov)'*(realSignal-  
weightProbability_reconstructedSignal_Markov)))
```

```
SNR2 =  
10.5397
```

Nuevamente, vemos que la estimación a través de la probabilidad ponderada nos reporta mejores resultados. También observamos que la señal está afectada por la propagación de errores que mencioné.

```
% Redactado por Alexandru Theodor Muntenaş.
```