Métodos Numéricos Avanzados
Un sistema de comunicaciones

María de la Puerta Echeverría (50009), Teresa Fontanella De Santis (52455) y Tomás Mehdi (51014)
Grupo 2 - Instituto Tecnológico de Buenos Aires (ITBA)

Resumen—En el presente artículo se describe la estimación de un canal de un sistema de comunicación mediante el método de cuadrados mínimos. El entorno de programación que se utiliza es Matlab.

Palabras claves—Sistema de comunicación, estimación de canal, respuesta al impulso, cuadrados mínimos, QR.

1 Introducción

Un sistema de comunicaciones consta, como mínimo, de tres componentes: un transmisor (que envía los datos), un receptor (que los recibe) y el medio por donde se transmiten los datos o *canal*. Dado que éste último modifica la información transmitida, es necesario que el receptor pueda conocerlo, para poder interpretar los datos de forma adecuada. Como generalmente no se conoce, resulta necesario estimarlo. A eso se le llama *estimación del canal*. En el presente informe se estima el canal, utilizando el método QR para cuadrados mínimos.

En la siguiente sección, Metodología Utilizada, se describe la misma, así como las pruebas realizadas. En la última sección se muestran los resultados obtenidos y se extraen conclusiones.

2 Metodología utilizada

Para la realización de este trabajo, se toman en cuenta las siguientes etapas:

2.1 Modelo empleado

El transmisor envía un dato s_k cada T segundos, donde s_0 es enviado en $t = 0$, s_1 en $t = T$, s_2 en $t = 2T$, ...

La modificación de los datos por el canal se presenta por la denominada *respuesta al impulso del canal* $\{h_k\}_{k=0}^{L}$, donde L es la longitud de la respuesta al impulso.

Además de ser modificados por el canal, los datos son afectados por ruido blanco Gaussiano aditivo $N_k \sim cN(0, \sigma)$.

Teniendo todo esto en cuenta, cada T segundos el receptor observa:

$$r_n = \sum_{k=0}^{L-1} h_k s_{n-k} + N_n \tag{1}$$

También podemos expresar la Ec. (1) en forma matricial como

$$\vec{r} = H\vec{s} + \vec{N} \tag{2}$$

donde

$$\vec{r} = \begin{pmatrix} r_0 \\ r_1 \\ ... \\ r_M \end{pmatrix}, \vec{s} = \begin{pmatrix} s_0 \\ s_1 \\ ... \\ s_{M-1} \end{pmatrix}, \vec{N} = \begin{pmatrix} N_0 \\ N_1 \\ ... \\ N_{M-1} \end{pmatrix} \tag{3}$$

$$H = \begin{pmatrix} h_0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ h_1 & h_0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & 0 & 0 & \dots & 0 & 0 \\ h_{L-1} & h_{L-2} & \dots & h_1 & h_0 & 0 & \dots & 0 & 0 \\ 0 & h_{L-1} & \dots & h_2 & h_1 & h_0 & \dots & 0 & 0 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_1 & h_0 \end{pmatrix} \in \mathbb{R}^{MXM}. \tag{4}$$

y $M \geq L$.

Otra forma equivalente es la siguiente:

$$\vec{r} = S\vec{h} + \vec{N} \tag{5}$$

donde $\vec{r}$ y $\vec{N}$ son como antes y

$$\vec{h} = \begin{pmatrix} h_0 \\ h_1 \\ ... \\ h_L \end{pmatrix}, \tag{6}$$

$$S = \begin{pmatrix} s_0 & 0 & \dots & \dots & 0 \\ s_1 & s_0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ s_{L-2} & s_{L-3} & \dots & s_0 & 0 \\ s_{L-1} & s_{L-2} & \dots & s_1 & s_0 \\ s_L & s_{L-1} & \dots & s_2 & s_1 \\ \vdots & s_{M-1} & s_{M-2} & \dots & s_{M-L} \end{pmatrix} \in \mathbb{R}^{MXL}. \tag{7}$$

Lo que nos interesa es recuperar correctamente la señal enviada $\vec{s}$ a partir de la señal recibida $\vec{r}$. Esto no sería tan difícil si se conociesen L y los $\{h_k\}$. En efecto, tomando $M = L$ en la Ec. (2), podemos buscar $\vec{s}$ que minimice:

$$\|H\vec{s} - \vec{r}\|_2^2. \tag{8}$$

El problema es, sin embargo, que en general ni L ni $\{h_k\}$ son conocidos: estimarlos corresponde al problema de estimación del canal. Si L es conocido, una forma de estimar el canal es mediante la Ec. (5) y el método de cuadrados mínimos. En efecto, tómese $M > L$ y envíese una señal conocida por el receptor $\vec{s} \in \mathbb{R}^M$ (denominada secuencia de entrenamiento). Luego, se estima el canal planteando el siguiente problema de cuadrados mínimos: encontrar $\vec{h} \in \mathbb{R}^L$ que minimice

$$\|S\vec{h} - \vec{r}\|_2^2. \tag{9}$$

2.2 Pruebas realizadas

Para efectuar las pruebas correspondientes, se considera utilizar un $L = 1, 10, 30, 50$[2] , aplicando un ruido gaussiano $\sigma = 1$. Se considera este valor por ser lo suficientemente chico para no distorsionar demasiado los datos a transmitir. También se prueba con diferentes longitudes de la cadena de entrenamiento $M = 32, 512, 1024$.
Para analizar cómo se estima en ”condiciones óptimas” (verbigracia: sin ruido, etc). también se prueba con $M = 512$ y $\sigma = 0$ (con los valores de L ya mencionados).

[2] Resolviendo dudas con el profesor, llegamos a la conclusión de que a valores de L mayores a 10, el resultado obtenido no es satisfactorio. Por este motivo, para que la prueba sea exitosa los valores de L deben ser $L = 1, 2, 3, 4, 5$.

3 Resultados y conclusiones

Para $L = 1$, $M = 32$ y $\sigma = 1$ se observa que la imagen de Lena se recupera de manera confiable a pesar de que el canal era inestable.

Para $L = 2$, $M = 32$ y $\sigma = 1$ la imagen se recupera en forma errónea dadas dos imágenes con distinto ruido.

Para $L = 3$, $M = 32$ y $\sigma = 1$ la imagen se recupera de manera bastante confiable. Dadas dos imágenes con distinto ruido, se obtienen imágenes recuperadas similares.

Para $L = 4$, $M = 32$ y $\sigma = 1$ las imágenes recuperadas son completamente distintas y distorsionadas, habiendo transmitido dos imágenes con ruido similar.

Para $L = 5$, $M = 32$ y $\sigma = 1$ y una imagen transmitida con poco ruido, ésta se recupera de manera confiable. Para una imagen con más ruido, se recupera una imagen errónea.

Para $L = 1$, $M = 64$ y $\sigma = 1$ no se recupera la imagen perfecta, pero es bastante confiable con respecto a la imagen original que contenía mucho ruido.

Para $L = 2$, $M = 64$ y $\sigma = 1$ se nota una variación importante para dos imágenes con poco ruido. Se recupera una imagen confiable y una muy distorsionada.

Para $L = 3$, $M = 64$ y $\sigma = 1$ se da una situación similar al caso anterior.

Para $L = 4$, $M = 64$ y $\sigma = 1$ y tres imágenes de Lena con distinto ruido, se recuperan dos de manera confiable y una distorsionada.

Para $L = 5$, $M = 64$ y $\sigma = 1$ y tres imágenes de Lena con distinto ruido, se recupera una de manera confiable y dos distorsionadas.

Para $L = 1$, $M = 256$ y $\sigma = 1$ se observan resultados muy similares al caso de igual L y $M = 64$.

Para $L = 2$, $M = 256$ y $\sigma = 1$ y tres imágenes con distinto ruido, se obtienen imagenes confiables.

Para $L = 3$, $M = 256$ y $\sigma = 1$ y tres imágenes con distinto ruido, ninguna se recupera de forma correcta.

Para $L = 4$, $M = 256$ y $\sigma = 1$ y tres imágenes con distinto ruido, se obtienen dos imágenes distorsionadas y una confiable.

Para $L = 5$, $M = 256$ y $\sigma = 1$ se recupera una imagen confiable a partir de una cuyo ruido es mayor a otras dos transmitidas y sus correspondientes no se recuperan con éxito.

Para $L = 1$, $M = 512$ y $\sigma = 1$ se recuperan dos imágenes defectuosas pero claras, dadas dos fotografías con gran nivel de ruido.

Para $L = 2$, $M = 512$ y $\sigma = 1$ se transmiten tres imágenes, dos con nivel de ruido similar y una con mayor cantidad. Para la imagen con mayor ruido, se obtiene una buena recuperación, aunque no perfecta y lo mismo sucede con una de las imagenes con menor cantidad de ruido. Con la otra, se obtiene una imagen defectuosa, aunque el nivel de ruido sea similar a la imagen que fue recuperada con éxito.

Para $L = 3$, $M = 512$ y $\sigma = 1$ y tres imágenes con distinto ruido aunque similar en nivel, dos se recuperan defectuosas y una confiable.

Para $L = 4$ y $L = 5$ con $M = 512$ y $\sigma = 1$ suceden una situaciones similares al caso anterior.

4 Conclusiones

Luego de correr el programa con los distintos valores propuestos, se observan resultados muy variados dependiendo del ruido que se le aplica al valor transmitido. Se observan situaciones en las que dadas dos imágenes con valor de ruido similar, se recuperan imágenes completamente distintas. No podemos afirmar que este método depende de que el ruido sea menor para obtener una recuperación exitosa, ya que hemos observado situaciones en las que se obtiene una imagen muy distorsionada, habiendo transmitido una imagen con poco ruido.

References

[1] John Mattews and Kurtis Fink, *Métodos Numéricos con Matlab*

5 ANEXO CDIGO

5.1 Ejecucion

```
function ej4(E, L, sigma)

%E tamano del s de entrenamiento (es un vector)
%E > L
%Analizar casos en que E es distinto que cols(imagen) 32,1024

%PASO 1
'Enviando imagen por canal inestable'
%h random para el primer caso en el que envio s de entrenamiento
L_aux = 5;
ganancia = 1/5;
h = ganancia*(1+randn(L_aux,1));

a = imread('lena512.bmp');
%a = 255*rand(10,10);
imwrite(uint8(a),'realimg.bmp');
%a=eye(5,5);

%Agrego una fila conocida a la imagen de longitud M, cols(imagen)
sTrainSent = 255*[rand(1,E) zeros(1,size(a,2)-E)];
sTrainSent = double(sTrainSent);
a = [ a ; sTrainSent ];

%imshow(a); % muestro la imagen original

M = size(a,2);
P = size(a,1);
H = toeplitz([h.' zeros(1,M-L_aux)],zeros(1,M));
r = zeros(M,P);
N = sigma*randn(M,1); % ruido
%Envio la imagen con un s extra conocido para entrenamiendo
%Se transmite una fila de la imagen por vez
for k=1:P
    s = double(a(k,:)'); % lo que se envia
    r(:,k) = H*s+N; % lo que se recibe
end
b = uint8(r(:,1:P-1).');
%imshow(b);
imwrite(b,'imgTrans.bmp');

% r = H*s+N
% r = S*h+N

%PASO 2
'Estimando H'
%Estimo h'_x donde x es la longitud(L) de respuesta al impulso
%Tengo r y S, uso QR para estimar h
%||S*h-r||_{2}^{2}
S = toeplitz(sTrainSent, zeros(1,L)); % S
%Uso M para cortar sTrainSent por que si mando uno mas largo que
%cols(imagen) ya no me interesan los que sobran
sTrainReceived = double(r(:,P)); % r, lo que recibi del entrenamiento
```

```
[Q R] = qr(S); % S*h = r  Entonces Q'*S*h = Q'*r Entonces R*h = Q'*r
%[Q R] = ourQR(S); % Nuestra implementacion

h_estimada = pinv(R)*(Q'*sTrainReceived); % Resolvemos R*h = Q'*r
H_estimada = toeplitz([h_estimada.' zeros(1,M-L)],zeros(1,M)); % Obtenemos H con h

%PASO 3
'Recuperando'
%Recupero la imagen con el H_estimada
r2 = r.';
for k=1:P-1
    u(:,k) = H_estimada\r(:,k); %lo que se recupera
end
%u(:,P+1) = sTrainReceived; %lo que recibi ultima linea
b2 = uint8(u.');
%imshow(b2);
imwrite(b2,'imgRec.bmp');
```

5.2 Implementacion QR por G-S

```
%si estas en matlab usar el siguiente codigo
function [Q R] = ourQR (A)
   [m n]=size(A);
    Q(:,1)=A(:,1);
    Q(:,1)=Q(:,1)/norm(Q(:,1),2);
    for i=2:n
        Q(:,i)=A(:,i);
        for k=1:i-1
            Q(:,i)=Q(:,i)-((A(:,i))'*Q(:,k))*Q(:,k);
        end
        Q(:,i)=Q(:,i)/norm(Q(:,i),2);
    end
    R= (Q')*A;
end

%si estas en octave usar el siguiente codigo
%function [Q R] = ourQR (A)
%   [m n]=size(A);
%    Q(:,1)=A(:,1);
%    Q(:,1)=Q(:,1)/norm(Q(:,1),2);
%    for i=2:n
%        Q(:,i)=A(:,i);
%        for k=1:i-1
%            Q(:,i)=Q(:,i)-((A(:,i))'*Q(:,k))*Q(:,k);
%        endfor
%        Q(:,i)=Q(:,i)/norm(Q(:,i),2);
%    endfor
%    R= (Q')*A;
%endfunction
```