

UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE INGENIERIA



PROYECTO MATLAB

CODIFICACION SHANON-FANO

MATERIA: Telecomunicaciones 2 ETN-1016

ESTUDIANTE: UNIV. CONDORI YUJRA MOISES MARTIN

DOCENTE: ING. Jose Campero

CODIFICACION SHANON-FANO

1. OBJETIVOS

- Realizar la codificación shanon-fano para un conjunto variable de caracteres con distinta probabilidad de ocurrencia.
- Realizar la medida de la entropía de la fuente de información codificada con el criterio shanon-fano.
- Realizar la medida de la longitud de la palabra del código resultante.
- Realizar el cálculo de la eficiencia del código.

2. INTRODUCCION

Como se conoce sobre la teoría de la información se puede construir códigos binarios para un determinado conjunto de símbolos, cuya longitud de palabra será lo más eficiente posible en base a su probabilidad.

Para un respectivo conocimiento de lo que es la codificación shanon-fano se lo explicara de la siguiente manera:

- Para un conjunto determinado de símbolos se tendrá una probabilidad para cada uno se debe verificar que la suma de las probabilidades debe ser igual a uno.
- Luego se procede a ordenar los símbolos de mayor a menor en relación a su probabilidad en columna.
- Una vez ordenados los símbolos se debe dividir en dos grupos cuya suma de probabilidades en cada grupo sea lo más distribuida posible y asignar a los símbolos de arriba un "1" pero también a los símbolos de abajo un "0".
- Se realiza el anterior paso hasta que ya no se pueda separar en ahí cada símbolo fue codificado.

3. DESCRIPCION FUNCIONAL DEL PROYECTO

El programa sigue la siguiente secuencia:

- Únicamente como datos de ingreso se le dio números de mensajes y probabilidades de cada uno.
- Si los datos cumplen con las especificaciones del programa continua.
- se procede a dividir los subgrupos haciendo una comparación entre todos los mensajes ya ordenados y realizando una suma consecutiva.
- Una vez dividido hace la asignación de ceros y unos.
- Realiza el paso anterior hasta que ya no se pueda subdividir cada grupo.
- Una vez finalizada la asignación se procede a calcular la entropía, longitud de la palabra y la eficiencia.
- Se muestran los resultados y el programa finaliza.

4. DESCRIPCION DE LAS FUNCIONES USADAS DE MATLAB

- **Input:** función que permite introducir algún carácter por el usuario ya sea número o letra depende a lo requerido.

- **Else if:** ambas son condiciones en este caso se lo usa para que el usuario únicamente pueda introducir un numero en el menú principal.
- **Case:** es para crear distintos casos a la hora de correr el programa.
- **Ceil:** sirve para redondear un numero al siguiente próximo.
- **Condicional &&:** en el cual se puede poner condiciones que se cumpla para que siga corriendo el programa.
- **Condicional ||:** se utiliza de igual manera que la anterior condicional se pueden poner condiciones para su respectivo seguimiento.
- **Elseif:** agrega mas condiciones.
- **For:** se lo utilizo para el manejo de ciclos en las condicionantes.

4.1 DESCRIPCION DE PARAMETROS DE ENTRADA Y SALIDA

- **Numero de símbolos:** con los cuales trabajara el programa.
- **Probabilidad:** es la probabilidad que tiene cada símbolo ingresado.
- **Probabilidad Ordenada:** es la probabilidad de los símbolos ingresados.
- **Codificación:** por cada símbolo mediante shannon-fano.
- **Entropía:** promedio de información en función de la probabilidad.
- **Longitud de la palabra:** numero de bits por cada símbolo multiplicado por su probabilidad de ocurrencia.
- **Eficiencia:** la relación entre entropía y longitud de palabra en porcentaje.

5. INGENIERIA DEL PROYECTO

5.1 ANALISIS

Primeramente, antes de empezar a programar en Matlab se hizo un análisis de lo que trata la codificación shannon-fano siguiendo los pasos para su codificación.

- Para un conjunto determinado de símbolos se tendrá una probabilidad para cada uno se debe verificar que la suma de las probabilidades sea igual a 1.
- Posteriormente se procede a ordenar los símbolos de mayor a menor en relación a su probabilidad y columna.
- Una vez ordenados los símbolos se debe dividir en dos grupos cuya suma de probabilidades en cada grupo sea lo mas distribuida posible y asignar a los símbolos de arriba un "1" pero también a los símbolos de abajo "0".
- Posteriormente se realiza el anterior paso hasta que ya no se pueda separar en ahí cada símbolo fue codificado.

Para que los datos sean analizados por el programa estos deben encontrarse entre un numero de símbolos que es de 2 a 16.

También debe verificar la suma de probabilidad de los mismos que será de 1.

Cuando el programa haya realizado la codificación procesada a calcular lo que es la eficiencia.

Las fórmulas que se emplearon para el cálculo de la entropía.

$$H = P_1 * \log_2 \left(\frac{1}{P_1} \right) + P_2 * \log_2 \left(\frac{1}{P_2} \right) + P_3 * \log_2 \left(\frac{1}{P_3} \right) + \dots P_n * \log_2 \left(\frac{1}{P_n} \right)$$

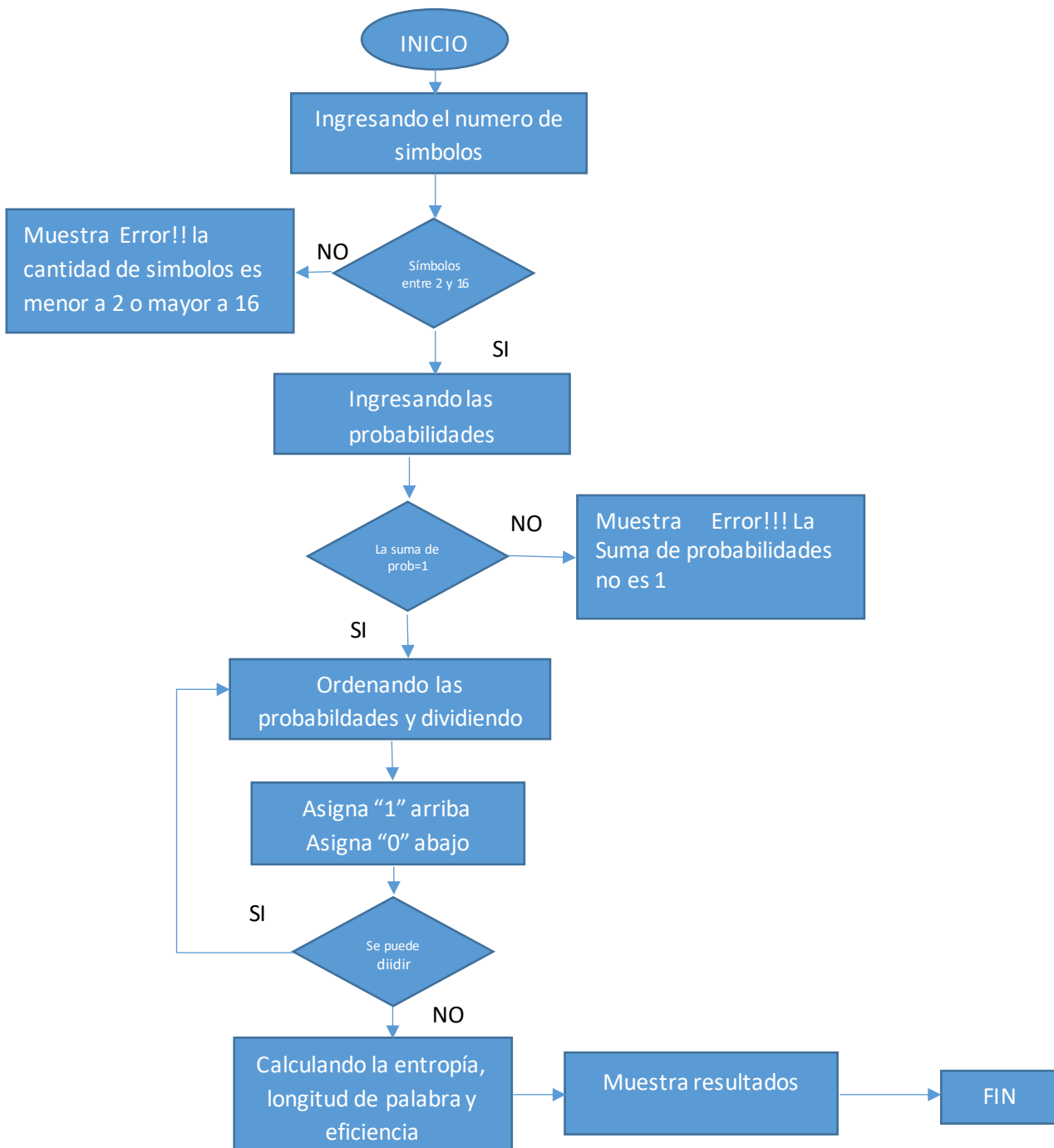
Para la longitud de palabra

$$L = \sum_{i=1}^n (N^{\circ} \text{ bits}) * P_{x_i}$$

Para la eficiencia

$$E = \frac{H}{L} * 100\%$$

5.2 FLUJOGRAMA



5.4 CODIGO MATLAB

```
%*****ETN 1016*****
%***+TELECOMUNICACIONES II***
%CODIFICACION SHANNON-FANO
%FLORES FLORES ROLY ALEX
%limpiando la ventana de comando
clc;
clear all;
close all;
%ingrsando la cantidad de simbolos y sus respectivas
probabilidades
m=input('Ingresar la cantidad de Simbolos: ');
if m<2 || m>16 % condicion que limita la cantidad de simbolos
que se debe introducir
    fprintf('\nError!! la cantidad de simbolos es menor a 2 o
mayor a 16\n\n');
else
    display('Ingresar las probabilidades de los simbolos:');
    for i=1:m % condicion para el ingreso de las respectivas
probabilidades
        fprintf('Simbolo %d: ',i);% el valor de la probabilidad se
muestra en pantalla
        pin(i,1)=input(' ');
    end
    for uv=1:m
        prob(uv)=pin(uv,1);
    end
    for i=1:m %condicion para que la suma de
probabilidades sea igual a 1
        for j=1:m-i
            if pin(j,1)<pin(j+1,1) %realiza la suma de probabilidades
de cada
                aux=pin(j,1); %simbolo ingresado
                pin(j,1)=pin(j+1,1);
                pin(j+1,1)=aux;
            end
        end
    end
    for uv=1:m
        prob2(uv)=pin(uv,1);
    end
    s=0;
    for i=1:m
        s=s+pin(i,1);
    end
    if (0.99999>s) || (1.00001<s) %verifica que la suma se igual a
1 haciendo cumplir la anterior condicion
        no=fprintf('\n\n Error!!! La Suma de probabilidades no es
1\n\n'); %caso contrario mostrara un mensaje de error
    else
        %%cuando la suma se igual a 1 prosigue con el ordenamiento de
las
        %%probabilidades
        variable=1;
        for k=1:m
            p(k,1)=variable;
            variable=variable+1;
        end
        resu=1;
        %se analiza el orden de las probabilidades
        for i=1:m
```

```

a=0;
b=0;
for j=1:i
    a=a+pin(j,1);
end
for k=i+1:m
    b=b+pin(k,1);
end
res2=abs(a-b);
if res2<resu
    resu=res2;
for jj=1:i
    p(jj,2)=0;
end

    for kk=i+1:m
        p(kk,2)=1;
    end
else
    resu=resu;
end
end

for i=1:m
    for j=3:m+5
        p(i,j)=5;
    end
end
for ll=1:m
    p(m+1,ll)=0;
end
for ii=3:m
    nn=0;
    for kk=1:m
        %mediante la creacion de un vector se hace la comparacion
        if p(kk,ii-1)~=p(kk+1,ii-1) %y realiza la
respectiva asignacion de unos y ceros

            if nn>0; % condicion la cual nos indica que ya no se puede
diividir y se iso la
            % asignacion
            res3=1;
            for iii=kk-nn:kk
                d=0;
                e=0;
                for jjj=kk-nn:iii
                    d=d+pin(jjj,1);
                end
                for kkk=iii+1:kk
                    e=e+pin(kkk,1);
                end
                res4=abs(d-e);
                if res4<res3
                    res3=res4;
                end

                for jjjjj=kk-nn:iii
                    p(jjjjj,ii)=0;
                end
                for kkkkk=iii+1:kk
                    p(kkkkk,ii)=1;
                end
            end
        end
    end
end

```

```

        else
            res3=res3;
        end
    end
    end
    nn=0;
    else
        nn=nn+1;
    end
end
for hh=1:m
for ss=1:m+5
pp(hh,ss)=p(hh,ss);
end
end
fprintf('\n');
fprintf('Mensaje\t\tX\t\t Prob      Código');
fprintf('\n');
fprintf('\n');
ent=0;
long=0;
%se realiza el analisis de los resultados, para la creacion de
las
%longitudes de pallabras individuales para posteriormente ser
sumadas
for tt=1:m
    mm=0;
    for uu=2:m+5
        if mm==0
            if pp(tt,uu)==5
                for bb=1:m
                    if probb(bb)==probb2(tt)
                        fprintf('Simbolo %d\t',bb);
                    end
                end
                probb2(tt)=2;
                probb(bb)=3;
            end
            fprintf('X%d\t%10.3f',tt,pin(tt,1));
            for xx=2:uu-1
                fprintf(' %d',pp(tt,xx));
            end
            long=long+((uu-2)*pin(tt,1)); %se calcula la suma de la
            longitud de palabra total
            fprintf('\n');
            mm=mm+1;
        end
    end
end
for zz=1:m
    ent=ent+(pin(zz,1)*(log2(1/pin(zz,1)))); %se realiza el
    calculo de la entropia
end
fprintf('\n');
fprintf('Entropia: H =%1.3f',ent) %se muestra el valor de
la entropia en pantalla
fprintf('\n');
fprintf('\n');
fprintf('Longitud Promedio de la Palabra: L =%1.3f',long) %se
muestra en pantalla la longitud de la palabra

```

```

fprintf('\n');
fprintf('\n');
fprintf('Eficiencia: E =%1.3f %',100*(ent/long)) %se calcula y
se muestra en el valor de la eficiencia en pantalla
fprintf('\n');
fprintf('\n');
end
end

```

6. RESULTADOS

```

Ingresar la cantidad de Simbolos: 8
Ingresar las probabilidades de los simbolos:
Simbolo 1: 0.125
Simbolo 2: 0.125
Simbolo 3: 0.125
Simbolo 4: 0.125
Simbolo 5: 0.125
Simbolo 6: 0.125
Simbolo 7: 0.125
Simbolo 8: 0.125

```


Mensaje	X	Prob	Código
Simbolo 1	X1	0.125	0 0 0
Simbolo 2	X2	0.125	0 0 1
Simbolo 3	X3	0.125	0 1 0
Simbolo 4	X4	0.125	0 1 1
Simbolo 5	X5	0.125	1 0 0
Simbolo 6	X6	0.125	1 0 1
Simbolo 7	X7	0.125	1 1 0
Simbolo 8	X8	0.125	1 1 1

Entropia: H =3.000

Longitud Promedio de la Palabra: L =3.000

Eficiencia: E =100.000

Como se puede observa cumple con la condición de equiprobabilidad y numero de símbolos donde la eficiencia es del 100%

 New to MATLAB? Watch this [Video](#), see [Examples](#), or read [Getting Started](#).

```

Simbolo 1: 0.09
Simbolo 2: 0.11
Simbolo 3: 0.01
Simbolo 4: 0.005
Simbolo 5: 0.2
Simbolo 6: 0.004
Simbolo 7: 0.12
Simbolo 8: 0.001
Simbolo 9: 0.28
Simbolo 10: 0.18

```

Mensaje	X	Prob	Código
Simbolo 9	X1	0.280	0 0
Simbolo 5	X2	0.200	0 1
Simbolo 10	X3	0.180	1 0 0
Simbolo 7	X4	0.120	1 0 1
Simbolo 2	X5	0.110	1 1 0
Simbolo 1	X6	0.090	1 1 1 0
Simbolo 3	X7	0.010	1 1 1 1 0
Simbolo 4	X8	0.005	1 1 1 1 1 0
Simbolo 6	X9	0.004	1 1 1 1 1 1 0
Simbolo 8	X10	0.001	1 1 1 1 1 1 1

Entropia: H =2.600

Longitud Promedio de la Palabra: L =2.665

Eficiencia: E =97.576

7. OBSERVACIONES Y RECOMENDACIONES

Para la sumatoria de probabilidades si se obtiene 1.001% el programa lo rechaza ya que se realizo el programa para que la suma si o si sea 1.000%.

Es necesario mencionar que en la parte de asignación cuando existe una ambigüedad de que lado tomar si ambos son simétricos, para la asignación de unos y ceros se toma los de arriba donde debe ser menor.

Se debe tener un buen conocimiento básico de Matlab para el buen desenvolvimiento, ya que el programa realiza comparaciones.

8. CUESTIONARIO

1) TEORIA DE LA INFORMACION:

También conocida como teoría matemática de la comunicación es una propuesta teórica presentada por Claude E. Shannon y Warren Weaver a finales de la década de los años 1940. Esta teoría está relacionada con las leyes matemáticas que rigen la transmisión y el procesamiento de la información y se ocupa de la medición de la información y de la representación de la misma, así como también de la capacidad de los sistemas de comunicación para transmitir y procesar información.[1] La teoría de la información es una rama de la teoría matemática y de las ciencias de la computación que estudia la información y todo lo relacionado con ella: canales, compresión de datos y criptografía, entre otros.

2) CORRECCION DE ERRORES POR REDUNDANCIA DE BITS:

Bits redundantes. Teóricamente es posible corregir cualquier fragmento de código binario automáticamente. Para ello, en pueste de los códigos detectores de errores utilizando los códigos correctores de errores, de mayor complejidad matemática y mayor número de bits redundantes necesarios. La necesidad de mayor número de bits redundantes hace que a veces la corrección de múltiples bits sea inviable e ineficiente por el elevado número de bits necesarios. Por ello normalmente los códigos correctores de error se reducen a la corrección de 1,2 o 3 bits.

3) EXPLICAR EN QUE CONDICIONES LA EFICIENCIA DEL CODIGO SHANNON – FANO ES MAXIMA (100%)

La eficiencia será máxima cuando existe equiprobabilidad y también el numero de símbolos es 2^n .

4) EXPLICAR EN QUE CONSISTE LA ENTROPIA “A PRIORI” Y “A POSTERIORI” DE LA INFORMACION

ENTROPIA:

“A PRIORI”

Se refiere a los simbolos de entrada, es decir antes de recibir un símbolo de salida determinado.

“A POSTERIORI”

Se refiere después de la recepción.

9. CONCLUSIONES

- Los resultados obtenidos con el programa en la prueba realizada fueron exactamente los mismos que los de la prueba manual con los cual se puede constatar que el programa funciona de manera satisfactoria.
- Se verifico el caso en que la eficiencia es del 100%.
- Se logró desplegar el resultado de los símbolos ordenados en base a sus probabilidades, se obtuvo la codificación correcta de los símbolos y se halló el valor de la entropía, longitud de la palabra y la eficiencia.

10. BIBLIOGRAFIA

- <https://es.wikipedia.org/wiki/codificacionshanon-fano>
- <http://ocw.upm.es/ingenieria-telematica/matlab-aplicado-a-la-ingenieria-telematica/contenido/tema5.pdf>
- <http://www.gedlc.ulpgc.es/docencia/seminarios/cd/codificacion/tsld005.htm>