

ANALISIS DE UN SINCRONIZADOR PARA DATOS CODIFICADOS CONVOLUCIONALMENTE

Oscar N. BRIA, Hugo E. LORENTE y Carlos H. MURAVCHIK

Departamento de Electrotecnia
Facultad de Ingeniería
Universidad Nacional de La Plata

Calle 48 y 116, La Plata BA, (1900) ARGENTINA
Tel. (021) 24-3709, Telex 31151 BULAP ARG

RESUMEN

El propósito de este trabajo es analizar el desempeño de un tipo de sincronizador para datos codificados convolucionalmente.

Se modeliza la máquina de estados del sincronizador como una cadena de Markov. Se define un parámetro de medida del desempeño para cuando se usa el sincronizador en un canal donde se transmiten paquetes, y se encuentra una expresión analítica para el mismo.

Finalmente se presentan resultados numéricos en forma gráfica que muestran la variación del parámetro de desempeño en función del número de estados de la máquina de estados del sincronizador, y de la probabilidad de error al bit en el canal de comunicaciones.

1. INTRODUCCION

En la comunicación por conmutación de paquetes puede ser necesaria una codificación redundante de los datos a transmitir con el propósito de detectar y corregir los posibles errores surgidos durante la transmisión a través de un canal ruidoso. Normalmente se utilizan para la codificación redundante códigos de bloque (Ref.1). Otra forma de verificar los datos y corregir los errores hace uso de códigos convolucionales. Estos últimos pueden resultar aptos en ciertas aplicaciones donde el canal utilizado es de radiofrecuencia.

Si se recurre a un código convolucional (Ref.2) los bits de datos y los que servirán para la verificación se envían de forma intercalada. En un código convolucional sistemático de tasa de datos $1/2$ se envían en forma alternada un bit de dato y uno calculado por una función convolucional de la información fuente que llamamos de verificación o paridad. Esto obliga a que en el receptor se provean los medios para: a) separar en dos subsecuencias la secuencia recibida, y b) decidir cuál de las subsecuencias corresponde a los datos y cuál a los bits de verificación.

En la Ref.2 se describe una máquina digital decodificadora (MD), de un código convolucional sistemático de tasa de datos $1/2$. MD demultiplexa la secuencia compuesta de bits que recibe,

en dos subsecuencias cada una de ellas formada por una de las dos posibles secuencias de bits alternados. En el procesamiento posterior de verificación y corrección, una de las secuencias es interpretada como secuencia de bits de datos y la otra como secuencia de bits de verificación de la primera.

Cuál de las dos secuencias es interpretada como secuencia de datos y cuál como de verificación depende exclusivamente de la fase de la señal de muestreo de la secuencia compuesta original, señal que llamaremos, como en la Ref.2, reloj de datos. Este reloj de datos se deriva de un reloj llamado de bit que está sincronizado con la secuencia de bits de entrada (y que tiene el doble de frecuencia que el reloj de datos).

Para cada una de las dos posibles fases del reloj de datos, MD puede estar en uno de dos posibles estados. Llamaremos a esos estados enganchado (E) y desenganchado (D). En el primero de ellos se discrimina correctamente cuales son los bits de datos y cuales los de verificación; en el segundo se hace la asignación incorrecta (Fig.1.1).

En Ref.2 se muestra como se relaciona una variable interna de la MD llamada síndrome con la fase del reloj de datos y la probabilidad de error al bit en el canal de comunicaciones (pe). Esta variable interna tiene la frecuencia del reloj de bit, y se genera en el decodificador para detectar y corregir errores.

El síndrome tiene las siguientes características (Ref.2):

- Es una variable binaria que se actualiza con la llegada de cada bit de la cadena compuesta (es decir su reloj es el de bit).
- Se calcula en forma automática en la misma MD, pues tiene otra aplicación interna en la misma MD, ver la Ref.2.
- El síndrome de cada uno de los bits de datos depende solamente de los errores introducidos por el canal, y en ausencia de estos es cero.
- Para el dispositivo citado el síndrome de cada uno de los bits de verificación depende no sólo de los errores sino también de los bits de datos y/o verificación.
- La probabilidad de que s sea igual a uno durante la recepción de un bit, dado que se calcula el síndrome con los bits de datos es:

Por lo tanto es de esperar, que el comportamiento del síndrome calculado en sincronismo con los datos (síndrome de los bits de datos) sea diferente del calculado en sincronismo con los bits de paridad (síndrome de los bits de paridad). En efecto, en la Ref.2 se demuestra que:

$$\Pr(s=1) = 3*pe \quad (1.1)$$

La probabilidad de que s sea igual a uno durante la recepción de un bit dado que se calcula el síndrome en sincronismo con los bits de paridad es:

$$\Pr(s=1) = 1/2 \quad (1.2)$$

Durante la recepción de un paquete de datos la evolución de la señal de síndrome puede interpretarse como un proceso aleatorio de tiempo discreto (sucesión de variables aleatorias) con espacio de estados $(0,1)$. A este proceso lo llamaremos S_n .

En la Ref.2 se describe una máquina de número de estados finito, diseñada con el propósito de que MD alcance el estado de sincronismo con los bits de datos, que llamaremos enganchado (E) y permanezca en él durante la recepción de un paquete. Aquí definimos una máquina parecida (ver Ref.8), llamada sincronizador de datos (en adelante MQ). Esta máquina tiene como entrada al proceso S_n y como salida la variable binaria que controla la fase del reloj de datos. A ésta última la llamaremos r y cada uno de los dos valores (r_1 y r_2) que puede tomar corresponde a una de las dos posibles fases del reloj de datos. En la Fig.1.2 se muestra como se relacionan funcionalmente MD y MQ.

En realidad existe una familia de MQs, cada una de ellas con un número de estados distinto. En la fig.1.3 se muestra el diagrama de estados, como la máquina de Moore de una MQ genérica de N estados MQ(N), el cual sirve como descripción de toda la familia. En el diagrama de estados de la Fig.1.4 se ha supuesto que r_1 es la fase correcta para el reloj de datos.

De las fórmulas 1.1 y 1.2 y de las figuras 1.2 y 1.3 se deduce, que si la fase del reloj de datos es la correcta, MQ incrementa su cuenta con probabilidad $3 \cdot p_e$, y la decreta con probabilidad un medio. Como se prohíbe que MQ disminuya su cuenta por debajo del estado uno, es de esperar que MQ tienda a permanecer en el estado uno o sus inmediaciones, para el caso usual de que $p_e \ll 1/2$.

De igual manera se deduce, que si la fase del reloj de datos es la incorrecta, MQ incrementa su cuenta con probabilidad un medio y la decreta con probabilidad $3 \cdot p_e$. En esta situación es de esperar si $p_e \ll 1/2$; que MQ aumente en promedio su cuenta. Pero al llegar al estado $N/2$ (umbral de decisión) se invierte la fase del reloj de datos y se pasa a un estado similar al uno pues se supone que la nueva fase es la correcta.

Expongamos ahora otro aspecto del problema que nos ocupa. Si por ejemplo r_2 es la fase correcta para un paquete en particular, no lo será necesariamente para el próximo (o más generalmente para otro cualquiera). Es decir que la MQ debe estimar cuál es la fase a utilizar al comienzo de cada paquete. A tal efecto la información útil del paquete viene precedida por unos cuantos bits de sincronismo, de modo que la MQ pueda estimar en ese lapso la fase del reloj de datos que mantendrá a la MQ en el estado E.

El propósito de este trabajo es estudiar analíticamente el desempeño de las MQs excitadas por el proceso aleatorio S_n durante la recepción de un paquete; suponiendo que los paquetes están compuestos por $L+1$ bits de información útil (bits de datos y bits de verificación) precedidos por $M-1$ bits de sincronismo. Se utiliza como herramienta analítica la teoría de las cadenas de Markov.

En la sección 2 se demuestra que se puede describir la evolución de MQ como una cadena de Markov. En la sección 3 se discute un índice que califica el desempeño de las MQ. Los resultados obtenidos para la máquina de la Ref.2 son presentados en la sección 4.

2. CADENAS DE MARKOV DE LAS MQs.

Dadas las características de la familia de máquinas de estado finito MQs, abordaremos el estudio del comportamiento de las MQs excitadas por el proceso S_n . Para ello en esta sección definimos el proceso aleatorio que describe la evolución de los estados de las MQs y demostramos que se trata de una cadena de Markov.

Definimos un proceso aleatorio de tiempo discreto:

- X_n , $n=1,2,\dots$; es un proceso aleatorio (sucesión de variables aleatorias) donde cada X_i es una variable aleatoria cuyos valores pertenecen al conjunto de estados de MQ. Este conjunto es $X=(x_1, x_2, \dots, x_N)$.

Veamos como se relaciona el proceso X_n con el hecho de que MD se halle enganchada (fase correcta del reloj de datos) o no:

- Durante toda la recepción de un paquete sólo una de las dos fases será la correcta.
- Al conjunto $X=(x_1, x_2, \dots, x_N)$ por comodidad lo desdoblamos en $(e_1, e_2, \dots, e_{N/2}) \cup (d_1, d_2, \dots, d_{N/2}) = E \cup D$. E es el conjunto de todos aquellos estados de MQ que tienen como salida la fase correcta, y D es el conjunto de todos aquellos estados de MQ que tienen como salida la fase incorrecta.
- Si suponemos en lo que sigue que r_1 es la fase correcta. Entonces cuando X_n tome valores en E la fase será r_1 . Y cuando X_n tome valores en D la fase será r_2 , que es la fase incorrecta, y por lo tanto será $Z_n=D$.

Con las definiciones, suposiciones y asignación numérica secuencial hecha a los estados (ver Fig.1.3) podemos ahora escribir:

$$E = (e_1, e_2, \dots, e_{N/2}) = (x_1, x_2, \dots, x_{N/2}) \quad (2.1)$$

$$D = (d_1, d_2, \dots, d_{N/2}) = (x_{N/2+1}, x_{N/2+2}, \dots, x_N) \quad (2.2)$$

E y D se definen así si y sólo si r_1 es la mejor respuesta.

En lo que sigue se tomarán las siguientes convenciones:

- El signo " " significa intersección lógica.
- $CD = 1$ significa que el flanco negativo del reloj de datos ocurre durante un bit de de datos.
- $CD = 0$ significa que el flanco negativo del reloj de datos ocurre durante un bit de paridad

Recordando las características del proceso aleatorio S_n podemos escribir (ver fórmulas 1.1 y 1.2):

$$Pr(S_n=1/X_n \text{ pertnc } E \cdot CD=1) = p = 3p_e \quad (2.3)$$

$$Pr(S_n=1/X_n \text{ pertnc } D \cdot CD=1) = 1/2 \quad (2.4)$$

Como así también:

$$\Pr(S_n=1/X_n \text{ pertnc } E \cdot CD=0) = 1/2 \quad (2.5)$$

$$\Pr(S_n=1/X_n \text{ pertnc } D \cdot CD=0) = p = 3pe \quad (2.6)$$

Como lo que interesa es caracterizar el proceso (X_n) , deberíamos conocer la distribución conjunta de las X_n o lo que equivalente (Ref.5), para todo n :

$$\Pr(X_{n+1}=x_{n+1}/X_n=x_n, X_{n-1}=x_{n-1}, X_{n-2}=x_{n-2}, \dots, X_0=x_0) \quad (2.7)$$

con x_i pertncnt a X .

El estado futuro de una máquina de número de estados finito depende únicamente del estado actual (X_n) y la entrada actual (S_n) (ver fig. 2.1). Como el valor de S_n depende del estado presente de X y CD (2.3-2.6), es fácil demostrar (Ref.8) que:

$$\begin{aligned} & - \Pr(X_{n+1}=x_{n+1}/X_n=x_n, X_{n-1}=x_{n-1}, X_{n-2}=x_{n-2}, \dots, X_0=x_0) = \\ & \Pr(X_{n+1}=x_{n+1}/X_n=x_n) \end{aligned} \quad (2.8)$$

Un proceso con la propiedad dada por la fórmula 2.8 es un proceso de Markov de tiempo discreto y como tiene un espacio de estados denumerable es una cadena.

Una cadena de Markov tiene la propiedad de que, la probabilidad del comportamiento futuro de la sucesión dado que el estado presente es conocido exactamente, es independiente del conocimiento del comportamiento pasado.

A la probabilidad de que X_{n+1} sea el estado j , dado que X_n es el estado i o probabilidad de transición de un paso, la denotamos:

$$P_{i,j}^{n,n+1} = \Pr(X_{n+1}=j/X_n=i) \quad (2.9)$$

Cuando las probabilidades de transición de un paso (2.9) son independientes de n decimos que la cadena de Markov es homogénea. Podemos entonces simplificar la notación:

$$P_{i,j}^{n,n+1} = P_{i,j} \quad (2.10)$$

Una cadena de Markov homogénea queda completamente definida por:

- La distribución de probabilidades iniciales, que puede ponerse en forma de vector.
- El conjunto de probabilidades de transición de un estado a otro, para todos los posibles pares de estados, que puede arreglarse en forma de matriz. Esta matriz es la matriz de transición de Markov.

El proceso X_n que describe la evolución de los estados de MQ resulta ser una cadena finita de Markov homogénea. A continuación hallaremos la matriz de transición para una MQ genérica.

Si seguimos considerando a r_1 como la fase correcta y tenemos presentes las las fórmulas 2.3, 2.4, 2.5 y 2.6, podemos escribir el siguiente conjunto de relaciones (2.11), con $p=3pe$; $m=1/2$ y $q=1-p$:

Si i es par,

Para $j=1+1$:

-Si i pertnc a $(EE)-eN/2$,

entonces

$$\Pr(X_{n+1}=j/X_n=i)=p$$

-Si i pertnc a (DD)-dN/2,

entonces

$$\Pr (X_{n+1}=j / X_n=i) = m$$

Para $j=1-1$:

-Si i pertnc a (EE),

entonces

$$\Pr(X_{n+1}=j/X_n=i)=q$$

-Si i pertnc a (DD),

entonces

$$\Pr(X_{n+1}=j/X_n=1)=m$$

Si i es impar,

Para $j=1+1$:

-Si i pertnc a (EE)-e1,

entonces

$$\Pr(X_{n+1}=j/X_n=1)=m$$

-Si i pertnc a (DD)-d1,

entonces

$$\Pr(X_{n+1}=j/X_n=1)=q$$

Para $j=i-1$:

-Si i pertnc a (EE)-e1,

entonces

$$\Pr(X_{n+1}=j/X_n=1)=m$$

-Si i pertnc a (DD)-d1,

entonces

$$\Pr(X_{n+1}=j/X_n=1)=p$$

Además:

$$\Pr(X_{n+1}=e_2/X_n=e_1)=1$$
$$\Pr(X_{n+1}=d_2/X_n=d_1)=1$$
$$\Pr (X_{n+1}=d_2 / X_n=e_{N/2})=p$$
$$\Pr (X_{n+1}=e_2/X_n=dN/2)=m$$

Si tenemos presentes las fórmulas 2.1 y 2.2

$$E = (e_1, e_2, \dots, e_{N/2}) = (x_1, x_2, \dots, x_{N/2})$$

$$D = (d_1, d_2, \dots, d_{N/2}) = (x_{(N/2+1)}, x_{(N/2+2)}, \dots, x_N)$$

se puede ver fácilmente que la matriz de Markov correspondiente a una MQ genérica, de N estados, con probabilidad de error de un bit pe, es (2.12):

[illegible]

Otra forma de representar las probabilidades de transición consiste en construir el diagrama de estados y transiciones de la cadena. En la figura 2.2 se ve el diagrama de estados y transiciones de la cadena de Markov que representa a MQ(12). Los nodos del diagrama representan los estados y los arcos dirigidos las probabilidades de transición de un estado a otro. Si bien las máquinas de número de estados finito también admiten una representación de diagramas de estados y transiciones, ambas representaciones no deben confundirse; ni tampoco pensarse que una representación se infiere directamente de la otra.

Por simple inspección (Refs.3 y 4) se pueden ver algunas características de las cadenas de Markov de las MQs. En la Ref.8 se arriba a la conclusión que las cadenas Xns que describen la sucesión de los estados de las MQs, son irreducibles, periódicas de período dos, y recurrentes.

3. INDICE DE DESEMPEÑO PARA LAS MQs

Con el objeto de evaluar el desempeño de las MQ debemos definir una función de costo o índice que permitan juzgar su comportamiento.

A primera vista parece que un buen objetivo de funcionamiento, para cada una de las realizaciones, es que la cantidad de bits recibidos con la MQ enganchada sea máximo, para lo cual debe darse que la MQ permanezca enganchada (estado E) la mayor cantidad de tiempo posible (no olvidemos que la mínima unidad de tiempo es el tiempo de duración de un bit).

Si bien la proporción de tiempo en que la máquina está enganchada puede ser un índice adecuado para comparar el desempeño de distintas MQs; su bondad o utilidad dependerá del contexto en que sean utilizadas.

No debemos olvidar que estamos interesados en recibir paquetes de información (conjuntos de bits en secuencia y no bits individuales); por lo tanto

La MQ debe permanecer en el estado E durante toda la recepción (todos los bits) de información del paquete para que sea considerado aceptable. (3.1)

En consecuencia definimos los eventos (disjuntos y complementarios) PE y PD:

PE: El procesamiento de cada uno los bits de información del paquete se hace con la MQ enganchada.

PD: El procesamiento de al menos uno de los bits de información (uno o más bits de información) del paquete se hace con la MQ desenganchada.

Claramente PE y PD son mutuamente exclusivos y su unión es el evento cierto; por lo tanto:

$$\Pr(PE) = 1 - \Pr(PD) \quad (3.2)$$

Dado que para considerar aceptable un paquete debemos recibir todo el paquete engachando, concluimos $Pr(PE)$ es mejor que la proporción de tiempo que MQ está enganchada como índice de desempeño.

A continuación calcularemos $Pr(PE)$ valiéndonos del hecho que el comportamiento de MQ puede ser descripto por la cadena de Markov X_n .

Precisando un poco más la sentencia 3.1 la MQ debe:

- Estar en E durante la recepción del primer bit de información del paquete.
- Permanecer en E durante todo el tiempo que demora la recepción de cada uno de los restantes bits de información del paquete.

Es conveniente hacer la precisión anterior si suponemos que la recepción de paquetes debe realizarse en forma aperiódica. Más aún si se pierde el sincronismo durante el tiempo en que no se reciben paquetes.

Habitualmente los paquetes de comunicaciones tienen un "encabezamiento" en su comienzo, parte del cual está destinado a lograr sincronización. A continuación de dicho encabezamiento se transmiten los bits de información. Por lo tanto la MQ debe arribar al estado E antes de que finalice la parte de sincronización del encabezamiento de cada paquete.

Podemos considerar al paquete como dividido en $(M-1)$ bits para sincronización y los restantes $(L+1)$ bits como de información. Entonces la probabilidad de que la MQ esté desenganchada en al menos un bit de información está dada por:

$$\begin{aligned} Pr(PD) &= Pr(PD, (XM=E)) + Pr(PD, (XM=D)) = \\ &= Pr(PD, (XM=E)) + Pr(PD/XM=D) * Pr(XM=D) = \\ &= Pr(PD, (XM=E)) + Pr(XM=D) = \end{aligned} \quad (3.3)$$

pues por definición $Pr(PD/XM=0)=1$. Con (3.2) se tiene:

$$Pr(PD) = Pr(PD, XM=E) + (1 - Pr(XM=E)) \quad (3.4)$$

En la Ref.8 se calculan $Pr(PD, XM=E)$ y $Pr(XM=E)$. A continuación transcribimos las ecuaciones finales:

$$Pr(PD, XM=E) = \quad (3.5)$$

$$[p_0, p_1, \dots, p_N] P^M \begin{bmatrix} 1(e_1), 1(e_2), \dots \end{bmatrix} (P(E, E) - I)^{-1} (P(E, E) - I)^L P(E, D) 1$$

y

$$Pr(XM=E) = [p_0, p_1, \dots, p_N] P^M 1(E) \quad (3.6)$$

De las ecuaciones (3.5) y (3.6) obtenemos una expresión para $Pr(PE)$:

$$\Pr(PE) =$$

(3.7)

$$[p_0, p_1, \dots, p_N] P^M$$

$$(1(E) - [1(e_1), 1(e_2), \dots]) (P(E,E) - I)^{-1} (P^L(E,E) - I) P(E,D) 1)$$

donde:

e_i Estado enganchado número i .

d_i Estado desenganchado número i .

E Unión de todos los e_i .

D Unión de todos los d_i .

$[p_0, p_1, \dots, p_N]$ Vector fila de distribución de probabilidades iniciales. Expresa la probabilidad de estar en el estado i al comienzo del primer bit de sincronismo de un paquete. Como no hay motivo para favorecer un estado de la MQ sobre otro, suponemos la situación más desfavorable (en el sentido de máxima entropía) que es la distribución uniforme.

$1(x)$ Vector columna de dimensión $N \times 1$, con todos ceros excepto un uno en la posición (x) .

1 Vector columna de todos unos, de dimensión $N/2 \times 1$.

P Matriz de Markov de la máquina de estados del sincronizador

A fin de simplificar los cálculos y la notación se ha dividido a P en cuatro submatrices:

$$P = \begin{bmatrix} P(E,E) & P(E,D) \\ P(D,E) & P(D,D) \end{bmatrix}$$

4. RESULTADOS NUMERICOS

En la fórmula 3.7 pueden distinguirse los siguientes parámetros:

- N , número de estados de la MQ.
- p_e , probabilidad de error al bit en el canal de comunicaciones
- $M-1$, número de bits de sincronismo del paquete.
- $L+1$, número de bits de información del paquete.

En la gráfica 1 se muestran los resultados de la evaluación de la fórmula 3.7 para:

- N variando entre 4 y 32 estados;

- b. p_e variando entre 10 y 10 ;
- c. $M-1$ igual a 100 bits;
- d. $L+1$ igual a 1000 bits;

o sea que la gráfica 1 muestra $Pr(PD)$ en función de p_e , con N como parámetro, y M y L fijos. Si bien el índice de desempeño $Pr(PE)$ debe estar más cerca a uno cuanto mejor sea el comportamiento de la MQ, de (3.2) se ve es lo mismo que $Pr(PD)$ sea lo más pequeño posible. Analizamos $Pr(PD)$ para facilitar la visualización de los resultados.

En la gráfica 1 se ve que $Pr(PD)$ crece en función de p_e para todos los valores de N . Esto significa que el índice de desempeño baja a medida que la probabilidad de error aumenta.

Para un valor dado de p_e y para los valores bajos de N , se ve que el desempeño mejora al aumentar en número de estados del sincronizador. Sin embargo para igual p_e y valores mayores de N el desempeño desmejora. Esta situación es más evidente para valores bajos de la probabilidad de error.

Se observa que para una cierta "calidad" del canal de comunicaciones, es decir un valor fijo de p_e , el desempeño presenta un óptimo con relación al número de estados de la máquina MQ. Esto puede visualizarse claramente en la gráfica 2.

Por lo tanto para cada aplicación específica (tipo de canal de comunicaciones) del sincronizador donde se prevea una franja de valores de p_e , existirá una franja de valores de N que obtienen los mejores desempeños.

Este comportamiento se explica del siguiente modo. Un N elevado favorece el comportamiento en estado E, adquirido durante la sincronización, a lo largo de la parte de información. Sin embargo si N es demasiado elevado el sincronizador se hace lento y no alcanzará el tramo de sincronización para alcanzar los estados de enganche.

La ventaja de usar N elevado radica en su robustez, o sea el poder desempeñarse muy bien para valores muy distintos de la probabilidad de error de bit (canales muy variables), al precio de tener que alargar el tramo de sincronización del paquete.

5. CONCLUSIONES

Se ha estudiado un tipo de sincronizadores de datos para códigos convolucionales fácilmente implementables como máquinas de número de estados finito.

Se demostró que la secuencia de estados del sincronizador constituía una cadena de Markov, y este conocimiento sirvió de base para el análisis del comportamiento de estos sincronizadores en función del nro de estados.

Del análisis de los resultados numéricos se concluye que, si se conoce el valor máximo de la probabilidad de error al bit en el canal, existe un valor óptimo para el número de estados del sincronizador.

Utilizando la teoría de las cadenas de Markov pueden estudiarse otros tipos de sincronizadores con distinta estructura de la máquina de estados.

REFERENCIAS

1. Sloane N.J.A.. A Short Course on Error Correcting Codes. CISM, Udine, Italia.
2. Lorente Hugo. Un sincronizador de datos para códigos convolucionales. CeTAD. Facultad de Ingeniería. U.N.L.P. Manuscrito del autor aceptado para su publicación en el boletín del Programa Nacional de Informática y Electrónica.
3. Cinlar E. Introduction to Stochastic Processes. Prentice - Hall. New Jersey. 1975.
4. S. Karlin y H.M. Taylor. A First Course in Stochastic Processes. Second Edition. Academic Press. Vol I.
5. Koroliuk, V. S.. Manual de la Teoría de Probabilidades y Estadística Matemática. Editorial Mir. Moscú. 1981.
6. Papoulis A. Probability, Random Variables, and Stochastic Processes. McGraw Hill. 1965.
7. Stern J. et.al.. Méthodes Pratiques d'Etude des Fonctions Aléatoires. Dunod. Paris. 1967.
8. O. N. Bria, H. E. Lorente y C. H. Muravchik. Análisis de un sincronizador para datos codificados convolucionalmente. Informe interno 4/88. Laboratorio de Comunicaciones. Departamento de Electrotecnia. Facultad de Ingeniería. U.N.L.P. 1988.

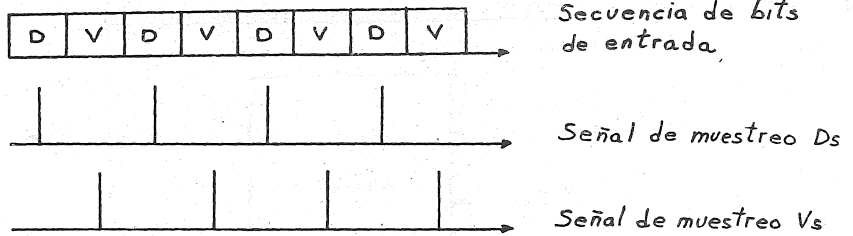


FIG 1.1. Señales de muestreo con la fase correcta para una determinada secuencia de entrada. $D \equiv$ dato, $V \equiv$ verificación.

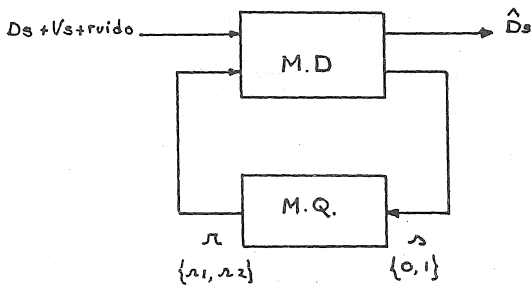


FIG 1.2. Relación funcional entre MD y MQ, y variables binarias

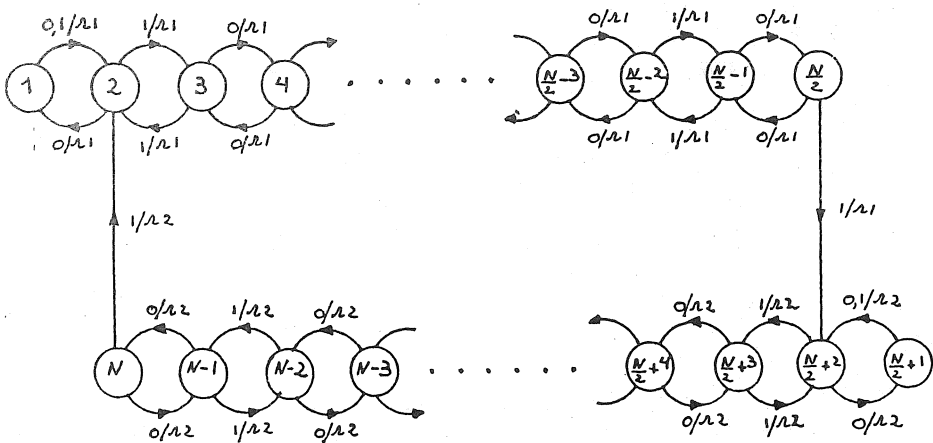


FIG. 1.3 Diagrama de estados y transiciones de una MQ genérica

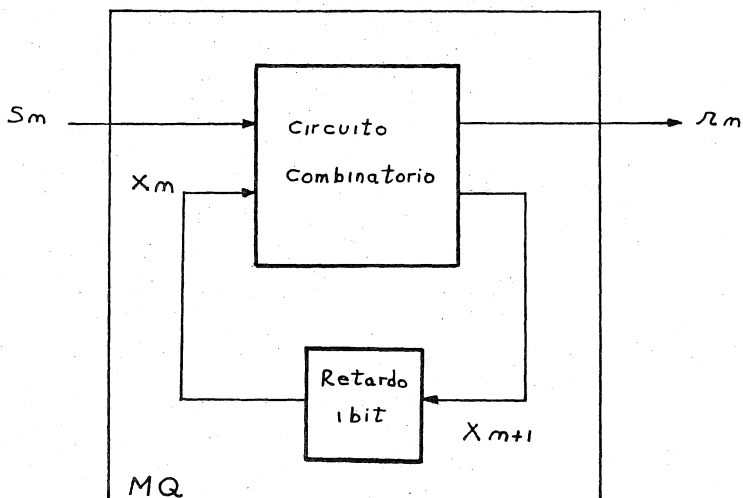


FIG. 2.1. MQ como máquina de estados donde se ponen en evidencia los procesos aleatorios intervinientes.

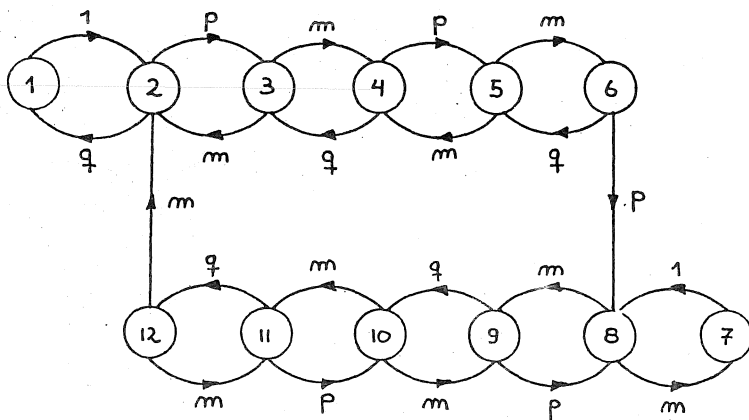
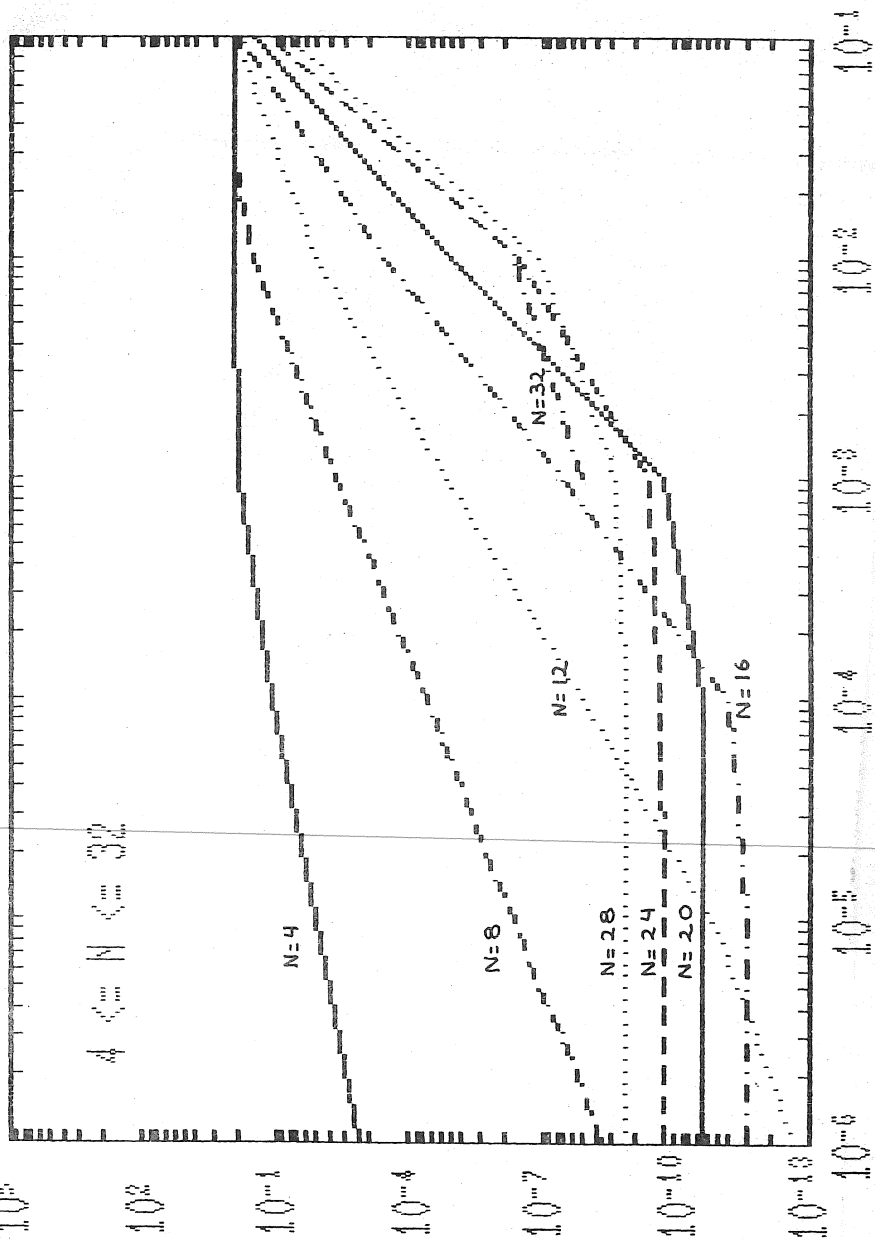


FIG. 2.2. Diagrama de estados y transiciones de la cadena de Markov de MQ(12), considerando a r_1 como la respuesta correcta.

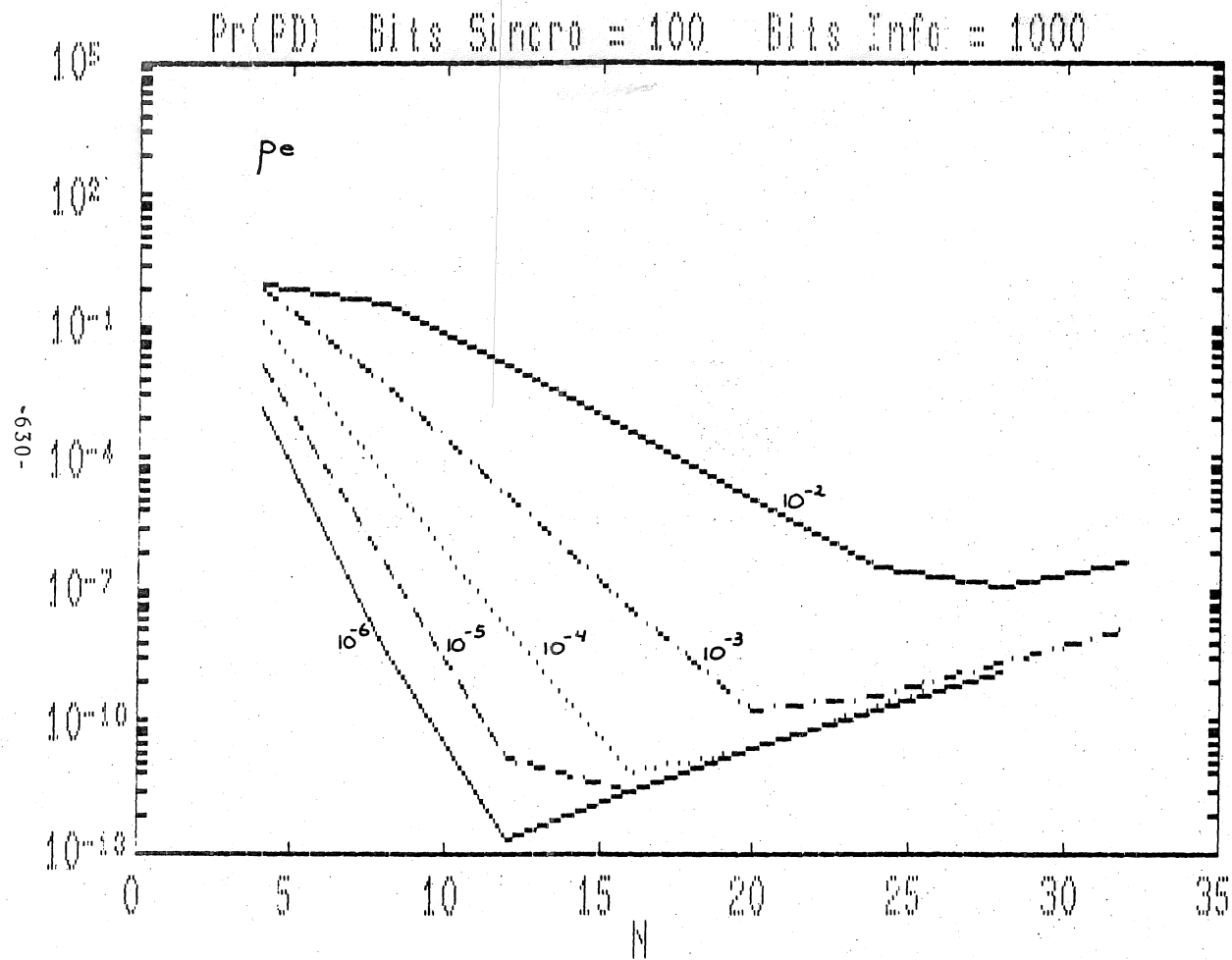
Pr(PD) Bits Sincro = 100 Bits Info = 1000

$4 \leq N \leq 32$



p_e

GRAFICA 1



GRAFICA 2