Universidad de la República

Facultad de Ingeniería

Codigos para Correccion de Errores

Proyecto Final

Autores:
Federico Bello
Rodrigo Torrado

28 de septiembre de 2025

UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Índice

1. Introducción

Los códigos *Reed-Solomon* (RS) son una clase de códigos correctores de errores utilizados en variados ámbitos, desde la trasmisión en comunicaciones digitales, hasta el almacenamiento en dispositivos como CD's, e incluso en comunicaciones espaciales [1]. Estos códigos, a diferencia de los códigos de Hamming, operan con conjuntos de bits, a los cuales se los denomina *símbolos*. Para operar con estos símbolos, se utiliza aritmética de cuerpos finitos o campos de *Galois* (GF). En este caso particular, se trabajara en un cuerpo finito de 929 elementos, es decir GF(929). Este tipo de códigos se utiliza para los códigos de barra PDF417, el cual es utilizado en diversos lugares como puede ser en transporte o el código postal estadounidense,

La figura 1 muestra un diagrama de bloques del sistema completo. Al codificador entra un mensaje $\mathcal{K}$, de largo k símbolos, y sale una palabra de código $X_{\mathcal{K}}$ de largo $n = k + r$, donde r es la redundancia del código. El canal le agrega un ruido a la palabra de código enviada, modificando los símbolos del mensaje y agregando errores y borraduras. Debido a esto, el objetivo del decodificador es obtener una estimación $\tilde{\mathcal{K}}$ que sea igual al mensaje enviado originalmente, sin errores.

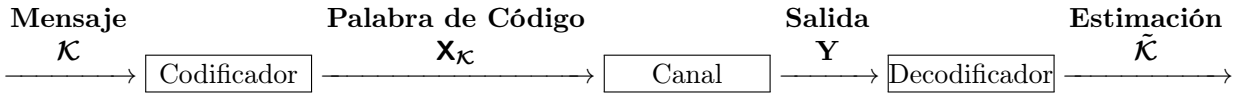


Figura 1: Diagrama de bloques de la trasmisión simulada

2. Canal

En este caso, el canal simulado es tal que agrega ráfagas de borraduras y errores aleatorios e independientes. Dado un bloque de n símbolos en la entrada, el canal borra con una probabilidad ρ los siguientes B símbolos. El símbolo de inicio se elige con una probabilidad uniforme entre el primero y el $(n - B)$-ésimo símbolo. Respecto a los errores, el canal actúa como un canal q-ario simétrico, con una probabilidad δ de error.

Al canal se lo denominara EEC(n,δ, ρ, B), donde los parámetros son los antes explicados. A continuación, se derivaran algunos resultados teóricos sobre la probabilidad de error en función de los parámetros del canal. Los mismos serán de utilidad luego para compararlos con los valores empíricos, además, de esta forma se podrá verificar la correctitud del decodificador.

Por el Teorema 1.7 de [4], se tiene que, dado un código cualquiera de distancia d, existe un codificador que corrige hasta τ errores y μ borraduras, siempre y cuando se cumpla que:

$$2\tau + \mu \leq d - 1 = r \tag{1}$$

Es importante aclarar que la igualdad $d - 1 = r$ se cumple para este caso en particular, dado que los códigos RS son códigos de distancia maxima (MDS).

2.1. Resultados Teóricos

2.1.1. Parte a

Asumiendo que se utiliza un código RS de parámetros $[n, n-r, r+1]$, se tiene que deducir la probabilidad de que un bloque codificado, que haya pasado por el canal EEC(n, δ, ρ, B), no sea decodificado correctamente por un decodificador convencional del código.

Para esto, primero se propuso encontrar la probabilidad de que dicho bloque si sea decodificado correctamente. Teniendo esto en consideración, se decidió estudiar dos casos por separado: cuando hubiesen borraduras y cuando no.

Para el caso sin borraduras hace falta deducir la probabilidad de que se encuentren errores que no impidan la decodificación correcta del bloque que pasa por el canal. Una de las asunciones que se realizan para llevar a cabo el desarrollo del algoritmo de decodificación para códigos RS (cuando no hay borraduras), es que la cantidad de errores en el bloque no puede superar la mitad de la redundancia: $\#Errores \leq \lfloor \frac{r}{2} \rfloor$. Si la cantidad de errores es mayor, el algoritmo de decodificación tendrá un comportamiento impredecible, nunca corrigiendo el bloque correctamente. Este resultado sale fácilmente de fijar la cantidad de borraduras en 0 en el resultado 1.

Por lo tanto, se puede corregir correctamente cualquier cantidad entre 0 y $\lfloor \frac{r}{2} \rfloor$ errores inclusive. Dado que son errores independientes de probabilidad fija δ, la probabilidad de encontrar i errores en un bloque sigue una distribución binomial, es decir:

$$P_E(i) = \binom{n}{i} \delta^i (1-\delta)^{n-i}$$

donde n es el tamaño del bloque en cuestión y δ es la probabilidad de que en un símbolo del mismo se produzca un error.

Entonces, a partir de lo anterior, la probabilidad de que se pueda decodificar correctamente un bloque que haya atravesado dicho canal es simplemente la suma de las probabilidades de que ocurran entre 0 y $\lfloor \frac{r}{2} \rfloor$ errores:

$$P_{SB} = \sum_{i=0}^{\lfloor r/2 \rfloor} \binom{n}{i} \delta^i (1-\delta)^{n-i}$$

Para el caso con borraduras es necesario seguir un razonamiento análogo, realizando algunas modificaciones. Primero, como se mencionó anteriormente, las borraduras se producen en ráfagas de B símbolos seguidos, por ésta razón, los errores en el bloque se pueden provocar en solamente $n-B$ símbolos, a diferencia del caso anterior. Debido a esto, la probabilidad de que en un bloque se den i errores, para el caso actual, es la siguiente:

$$P_E(i) = \binom{n-B}{i} \delta^i (1-\delta)^{n-B-i}$$

Por otro lado, utilizando el resultado 1, la cantidad de errores debe cumplir la siguiente inecuación para que se pueda decodificar el bloque de manera correcta: $\#Errores \leq \lfloor \frac{r-B}{2} \rfloor$.

Tomando en consideración lo anterior y siguiendo el razonamiento descrito previamente, se obtuvo una expresión para la probabilidad de que un bloque se pueda decodificar correctamente, en el caso donde hayan borraduras:

$$P_{CB} = \sum_{i=0}^{\lfloor (r-B)/2 \rfloor} \binom{n-B}{i} \delta^i (1-\delta)^{n-B-i}$$

Finalmente, para calcular la probabilidad de que un bloque se pueda decodificar correctamente utilizando un decodificador convencional del código RS en cuestión, alcanza con realizar la suma ponderada por la probabilidad de los dos resultados anteriores:

$$P_{DECOD} = (1-\rho) P_{SB} + \rho P_{CB}$$

$$P_{DECOD} = \rho \sum_{i=0}^{\lfloor (r-B)/2 \rfloor} \binom{n-B}{i} \delta^i (1-\delta)^{n-B-i} + (1-\rho) \sum_{i=0}^{\lfloor r/2 \rfloor} \binom{n}{i} \delta^i (1-\delta)^{n-i}$$

Por ultimo, la probabilidad de que no sea decodificado correctamente por un decodificador convencional del código es (1 - P_{DECOD}) :

$$p_{block} = 1 - \rho \sum_{i=0}^{\lfloor (r-B)/2 \rfloor} \binom{n-B}{i} \delta^i (1-\delta)^{n-B-i} - (1-\rho) \sum_{i=0}^{\lfloor r/2 \rfloor} \binom{n}{i} \delta^i (1-\delta)^{n-i} \tag{2}$$

2.1.2. Parte b

Asumiendo lo mismo que en la pregunta anterior, se tiene que demostrar que cuando δ es nulo, es decir, cuando solo hay borraduras, la probabilidad de que un bloque codificado, que haya pasado por el canal EEC(n, δ, ρ, B), no sea decodificado correctamente, está dada por la siguiente ecuación:

$$p_{block} = \rho[B > r]$$

donde $[B > r]$ se interpreta como 1 si se cumple la desigualdad, o 0 si no se cumple.

En este nuevo escenario, para que el bloque sea decodificado correctamente se tiene que cumplir la desigualdad 1 fijando la cantidad de errores en 0, es decir $\mu = B \leq r$. Por lo tanto, si existen borraduras y la cantidad de éstas es menor o igual a la redundancia del código, siempre se podrá decodificar correctamente. Análogamente, el bloque que atraviese el canal ya mencionado correrá el peligro de no poder ser decodificado **solo** si existen borraduras y las mismas superan estrictamente la redundancia del código, por ende, la probabilidad de que el bloque no sea decodificado correctamente, en este caso, es $\rho[B > r]$.

Además, se puede verificar el resultado imponiendo la restricción $\delta = 0$ en la ecuación 2. Utilizando la convención de que $0^0 = 1$, se ve como los únicos términos no nulos en las sumatioras de la ecuación 2 son los correspondientes a $i = 0$. En particular, la primer sumatoria es nula sii $\lfloor (r-B)/2 \rfloor < 0$, es decir, si $B > r$. Por lo tanto, se tiene que:

- Si $r >= B$:
$$p_{block} = 1 - \rho \binom{n-B}{0} 0^0 - (1-\rho) \binom{n}{0} 0^0 = 1 - \rho + \rho - 1 = 0$$
- Si $B < r$:
$$p_{block} = 1 - (1-\rho) \binom{n}{0} 0^0 = 1 - 1 - \rho = \rho$$

Logrando así verificar el resultado.

3. Decodificador

El decodificador a implementar es un decodificador de códigos RS convencional, con las modificaciones adecuadas para, además de poder corregir errores, poder corregir también borraduras. Se le llamará K al conjunto de índices cuyo símbolo fue borrado y J los índices cuyos símbolos fueron eliminados. Se definen el síndrome $S(x)$ y el polinomio localizador de errores $\Lambda(x)$ como es usual:

$$S(x) = \sum_{l=0}^{r-1} S_l x^l \qquad \Lambda(x) = \prod_{j \in J} (1 - \alpha_j x)$$

El polinomio evaluador de errores $\Gamma(x)$ se modifica, y también se define el polinomio localizador de borraduras $M(x)$ de la siguiente manera:

$$\Gamma(x) = \sum_{j \in (K \cup J)} e_j v_j \prod_{m \in (K \cup J) \{j\}} (1 - \alpha_m x) \qquad M(x) = \prod_{j \in K} (1 - \alpha_j x)$$

Como es de esperar, los polinomios $S(x)$ y $M(x)$ son conocidos de antemano, mientras que $\Gamma(x)$ y $\Lambda(x)$ serán calculados durante el algoritmo de decodificación.

Como se vio en la sección 2.1.2, si la cantidad de borraduras es mayor a la redundancia el bloque es incorregible, por lo que el primer paso del algoritmo es evaluar esta condición. De esta forma, si el bloque en consideración tiene mas de r borraduras, se interpreta como incorregible.

Luego, se calculan el síndrome y el polinomio localizador de errores, para calcular el síndrome modificado: $\tilde{S}(x) = M(x)S(x) \; mod \; x^r$. Para esto, alcanza con realizar la multiplicación y eliminar los coeficientes cuyo grado sea mayor a r.

Posteriormente, se aplica el algoritmo extendido de Euclídes, tomando como entradas $a(x) = x^r$ y $b(x) = \tilde{S}(x)$ y como condición de parada el único índice h tal que: $\deg r_h < \frac{1}{2}(r + \rho) \leq \deg r_{h-1}$. Un pseudo-algoritmo puede encontrarse en el capitulo 6 de [4], donde para adaptarlo alcanza con modificar la condición de parada. Una vez finalizado el algoritmo, se toma $\Lambda(x) = t_h$ y $\Gamma(x) = r_h$.

A partir de lo anterior, se calcula un nuevo polinomio localizador como $\tilde{\Lambda}(x) = M(x)\Lambda(x)$. Las raíces del mismo indican las ubicaciones de los errores y de las borraduras. Para hallarlas se computa el algoritmo de *Chien*, el cual se basa en realizar una búsqueda exhaustiva de las raíces recorriendo, en el peor caso, todos los elementos del cuerpo finito.

Finalmente, se aplica el Algoritmo de *Forney* [3] con $\tilde{\Lambda}(x)$ y $\Gamma(x)$, esto devuelve las posiciones y valores de los errores y las borraduras. Para obtener el mensaje enviado, simplemente se resta el vector de error de la palabra recibida y se descartan los últimos r símbolos de redundancia. Aunque es posible intercambiar el orden de estas dos últimas operaciones para mejorar ligeramente la eficiencia, este cambio no tiene un impacto significativo en el tiempo de ejecución del programa.

Un comentario no menor respecto a la búsqueda de Chien es que en los últimos años han surgido resultados los cuales optimizan las búsquedas de las raíces de un polinomio dentro de un determinado cuerpo finito. En particular, en [5] se desarrolló un método para disminuir la búsqueda de las raíces de los polinomios de grado no mayor a 11. Por otro lado, en [2] se propuso otro método sin restricciones, logrando mejorar la eficiencia en el caso general con respecto al algoritmo de Chien.

4. Experimentos

Se realizaron distintos experimentos, no solo para verificar el correcto funcionamiento del decodificador sino también para extraer estadísticas y comparar los resultados empíricos y teóricos. Todos los experimentos fueron realizados con datos aleatorios, utilizando siempre la misma semilla (42[1]), con el objetivo de que los experimentos sean reproducibles.

El primer experimento consistió en fijar la probabilidad de borraduras y luego, hacer variar tanto el largo de las mismas como la probabilidad de error. De esta forma, se propuso analizar el comportamiento del decodificador tanto en la probabilidad de error por bloque p_{block}, como en la cantidad de bloques que se dan por correctos, corregidos o incorregibles. Se utilizó un largo de código de $n = 96$, una redundancia de $r = 32$ y 1 millón de bloques, para así disminuir el efecto de la aleatoriedad en los resultados.

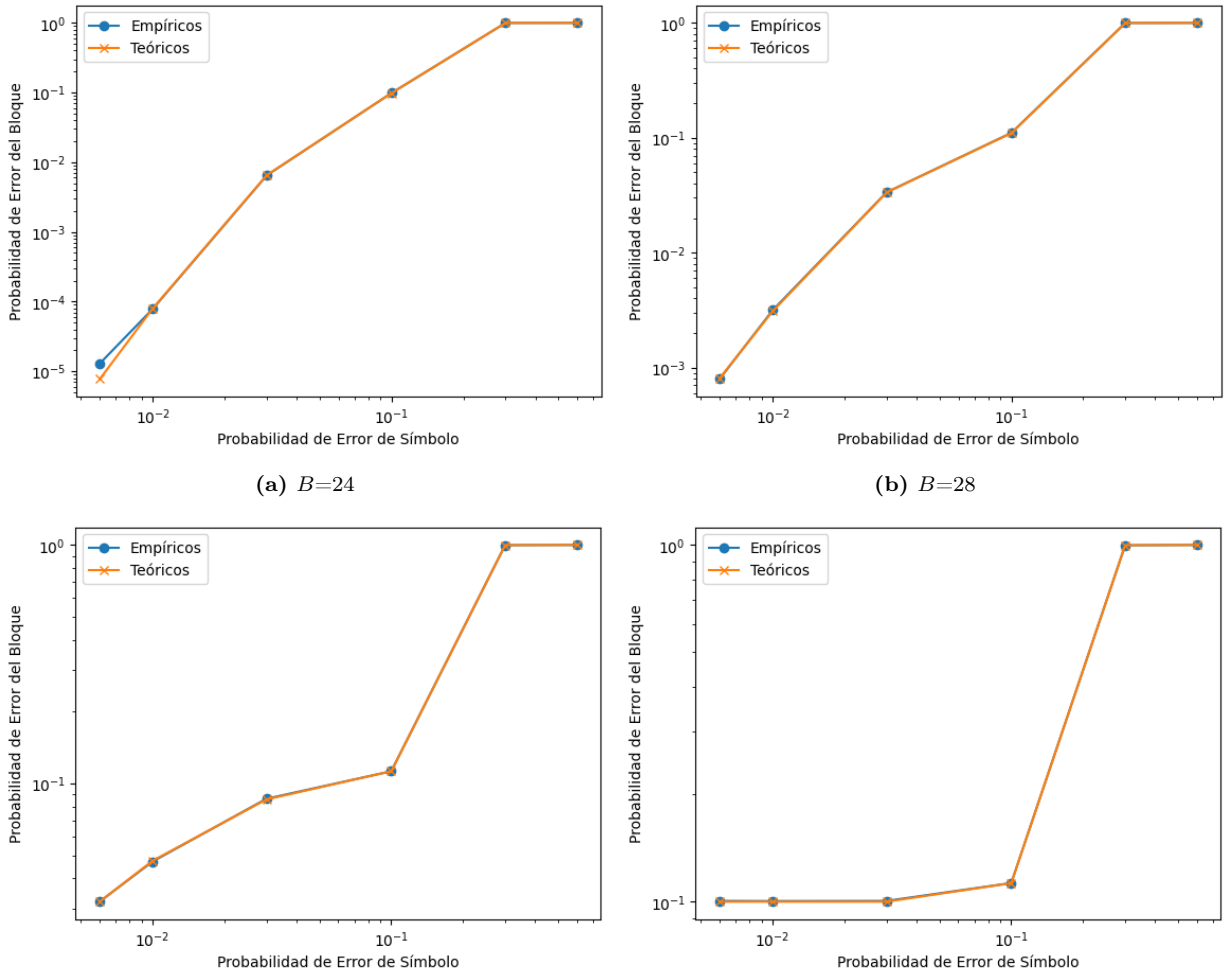
En la figura 2 se ve una comparación entre los valores teóricos y empíricos de la probabilidad de error

[1] `https://en.wikipedia.org/wiki/42_(number)`

por bloque para distintos valores de B. La probabilidad de borradura considerada fue $\rho = 0{,}1$ y se varió la probabilidad de error de símbolo en los valores $\delta = [0.006\ ;\ 0.01\ ;\ 0.03\ ;\ 0.1\ ;\ 0.3\ ;\ 0.6]$.

En los gráficos se puede observar claramente como el valor empírico resulta muy cercano al valor teórico, aumentando ambos a la par a medida que δ se incrementa. Esto es indicativo de que el decodificador funciona correctamente, alcanzando los limites impuestos por la teoría.

El hecho de que algunas palabras de código no se decodifiquen correctamente no implica que el decodificador no pueda cumplir su función; si para un determinado bloque de símbolos ocurre que su cantidad de errores y borraduras no verifican la condición 1, nunca se podrá recuperar correctamente ese bloque.



(a) B=24 **(b)** B=28

(c) B=32 **(d)** B=33

Figura 2: Probabilidad de error teórica y empírica para distintos largos de ráfagas de borraduras

La figura 3 muestra los valores empíricos de las figuras anteriores, superpuestos en la misma gráfica. De esta, se pueden extraer algunas observaciones interesantes. Primero, se puede apreciar que cuanto mayor sea el largo de la ráfaga de borraduras, mayor será p_{block}. En particular, se ve claramente que cuando se cumple que $B > r$, la probabilidad de error nunca es inferior a la probabilidad de las borraduras, yendo a la par con el resultado demostrado en 2.1.2.

Además, se ve como a partir de cierto valor (cercano a 1×10^{-1}) el valor de p_{block} no parece depender de B. Esto se debe que a medida que la probabilidad de error aumenta, la esperanza de la cantidad de errores totales también, resultando así en que se supere la cota 1 independientemente de que haya borraduras o la cantidad de las mismas.

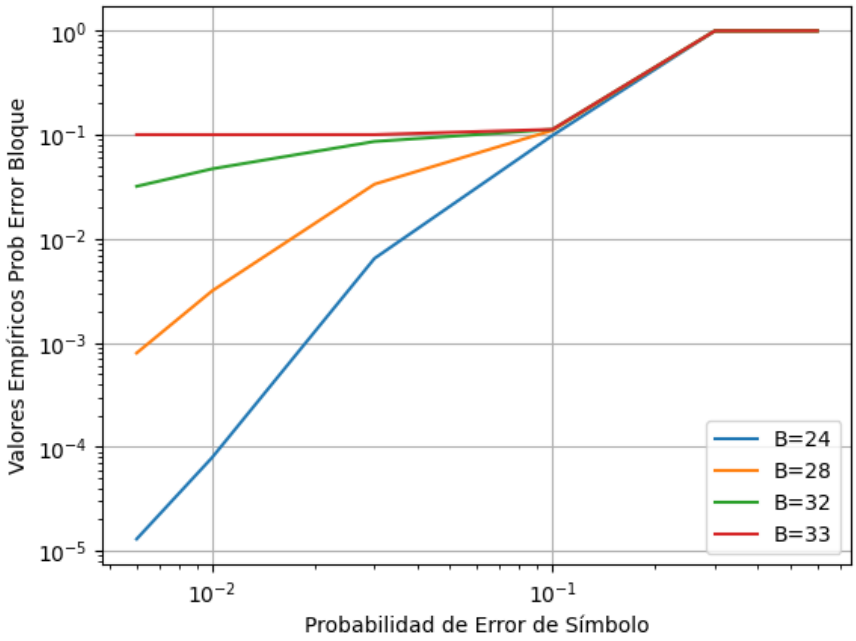


Figura 3: Probabilidad de error empírica para distintos largos de ráfagas de borraduras

Por otro lado, en la tabla 1 se ve la diferencia entre la cantidad de bloques erróneos y la cantidad de bloques que se interpretan como incorregibles por el decodificador. Un punto a destacar en los siguientes resultados es que cuando se identifica un bloque como incorregible, siempre será correcto, mientras que las diferencias surgen de los bloques que son incorregibles pero se detectan como corregibles. Esto se debe a que se intenta inicialmente corregir el bloque y solo si se encuentra algún error se clasifica como incorregible. Por lo tanto, siempre que la corrección sea posible, se realizará correctamente.

En la tabla, se ve que si la probabilidad de error es baja, se logra identificar correctamente los bloques incorregibles. Por otro lado, mientras esta aumenta se ve la tendencia de que la predicción es cada vez menos confiable. Una explicación plausible radica en el hecho de que al aumentar la probabilidad de errores, la distancia entre la palabra de código recibida y las palabras de código posibles que no fueron enviadas puede reducirse. Es decir, si el vector recibido se encuentra a una distancia $\frac{r}{2}$ de una palabra de código distinta a la original, el algoritmo seleccionará incorrectamente una palabra, considerándola como la decodificación correcta.

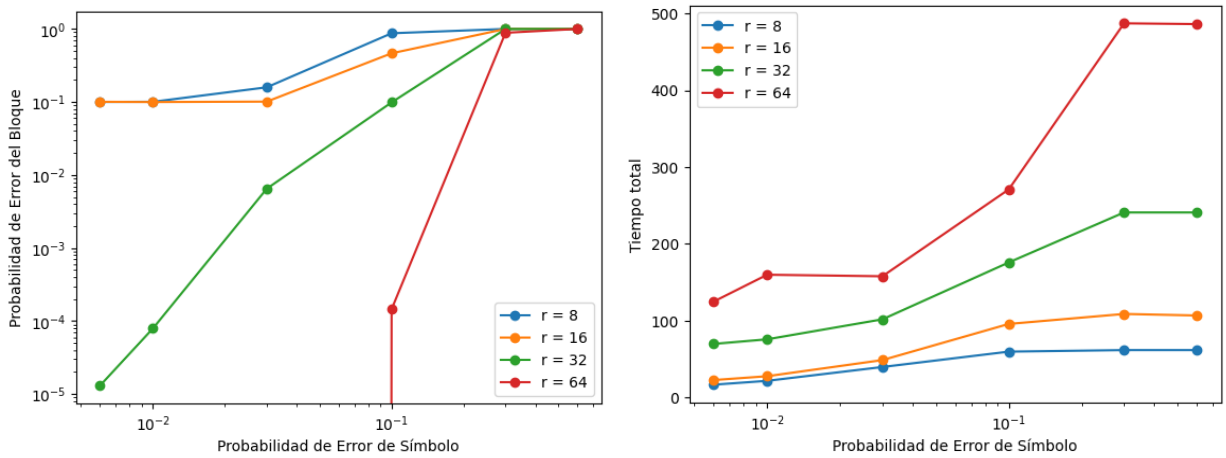
Por otra parte, cuando el largo de las borraduras supera la redundancia, la cantidad de palabras que se dan como corregidas cuando no lo fueron aumenta considerablemente. Esto es debido a que siempre que hayan borraduras, la palabra se dará como mal codificada automáticamente, disminuyendo los falsos negativos. Se ve entonces que la mayor falla del decodificador radica cuando tanto la cantidad de borraduras (sin superar cierto umbral) como la probabilidad de errores son altas. En condiciones favorables para el decodificador, no solo logra corregir una gran cantidad de errores sino que también es un confiable detector de los mismos.

Como ultimo experimento, se decidió variar el largo de la redundancia, con el objetivo de estudiar como varia p_{block} y el tiempo de ejecución. Es claro que se tiene un compromiso entre estos dos, ya que a mayor valor de redundancia se podrán corregir una mayor cantidad de errores, a cambio de una latencia y un tiempo de ejecución mayor. Este comportamiento es el que se puede observar en la figura 4, donde en 4a se puede ver como la probabilidad de error disminuye considerablemente a medida que la redundancia aumenta, y también como ésta aumenta de forma considerable una vez que la redundancia es mayor al largo de las borraduras. Sin embargo, un caso distinto es el de la figura 4b, donde se ve como al aumentar la

cantidad de símbolos enviados el tiempo de ejecución también aumenta. Este fenómeno, en el contexto de la transmisión de información, provoca un retardo en la entrega de la misma, mientras que en un escenario de almacenamiento, surge otra problemática: al aumentar la redundancia almacenada, se reduce el espacio disponible para datos de información.

Probabilidad Error \ Largo Borradura	B=24	B=28	B=32	B=33
$\delta = 0{,}006$	1	363	31965	43
$\delta = 0{,}01$	2	1490	47088	10
$\delta = 0{,}03$	233	15652	86317	11
$\delta = 0{,}1$	3269	45749	10076	18
$\delta = 0{,}3$	3606	47036	99921	10
$\delta = 0{,}6$	3814	46830	99912	0

Cuadro 1: Diferencia entre bloques incorrectos reales y bloques dados como incorrectos



(a) Probabilidad de error de bloque

(b) Tiempo de ejecución

Figura 4: Probabilidad de error de bloque y tiempo de ejecución en función de la probabilidad de error de símbolo para distintos valores de redundancia

Finalmente, los parámetros a elegir del código dependerán del caso de uso especifico. En escenarios donde lograr la trasmisión correcta es sumamente importante, se debería optar por un código con mayor redundancia, mientras que, si la retransmisión es efectiva, es favorable la elección de menor redundancia para así mejorar la eficiencia.

5. Conclusiones

En este informe, se ha abordado el análisis teórico y experimental de un decodificador de códigos Reed-Solomon (RS) capaz de corregir tanto errores como borraduras. Se han desarrollado resultados teóricos para

la probabilidad de error en función de los parámetros del canal y se ha presentado un procedimiento del decodificador, adaptado para enfrentar la presencia de borraduras.

Los experimentos realizados han confirmado la eficacia del decodificador, mostrando una consistencia notoria entre los resultados teóricos y empíricos. Se observó cómo la probabilidad de error por bloque aumenta con la probabilidad de error en los símbolos y el largo de las ráfagas de borraduras, siguiendo patrones coherentes con las expectativas teóricas. Además, se evidenció que cuando la probabilidad de error en símbolos es suficientemente alta, la presencia de borraduras juega un papel menos relevante en la probabilidad global de error.

En resumen, el decodificador implementado muestra un desempeño satisfactorio, adaptándose de manera efectiva a las condiciones del canal simulado, y los resultados obtenidos en los experimentos respaldan la validez de las predicciones teóricas.

Referencias

[1] Stephen B. Wicker; Vijay K. Bhargava. Reed-Solomon Codes and Their Applications, 1994.

[2] S.V. Fedorenko and P.V. Trifonov. Finding roots of polynomials over finite fields. *IEEE Transactions on Communications*, 50(11):1709–1711, November 2002.

[3] G. Forney. On decoding bch codes. *IEEE Transactions on Information Theory*, October 1965.

[4] Ron Roth. *Introduction to Coding Theory.* Cambridge University Press, February 2006.

[5] T.-K. Truong, J.-H. Jeng, and I.S. Reed. Fast algorithm for computing the roots of error locator polynomials up to degree 11 in reed-solomon decoders. *IEEE Transactions on Communications*, 49(5):779–783, May 2001.