ΑΣΚΗΣΗ 5η

Υπολογισμός πιθανότητας λάθους αποκωδικοποίησης σε απλούς κώδικες επανάληψης

5.1 Σκοπός της άσκησης

Σκοπός της άσκησης είναι να γραφτεί στο MATLAB πρόγραμμα υπολογισμού και αναπαράστασης της *πιθανότητας λάθους* (error probability, P_e) στο αποκωδικοποιημένο bit για *απλούς κώδικες επανάληψης (*simple repetition codes).

5.2 Θεωρητικό μέρος

Η μετάδοση δεδομένων μέσω ενός καναλιού επικοινωνίας το οποίο παρουσιάζει θόρυβο έχει ως χαρακτηριστικό την εμφάνιση *λαθών* (errors) κατά τη λήψη των εκπεμπόμενων συμβόλων (bit). Θα πρέπι να ειπωθεί ότι στην πράξη δεν υάρχει κανάλι επικοινωνίας χωρίς να παρουσιάζει θόρυβο (αθόρυβο) (noiseless). Το προηγούμενο πρόβλημα είναι δυνατό να λυθεί μερικά με την πρόσθεση επιπλέον bits αλλά με την συνεπακόλουθη μείωση του *ρυθμού εκπομπής δεδομένων* (transmission rate). Η πρόσθεση επιπλέον bits μεσκοπό τη μείωση του ρυθμού λαθών στην αποκωδικοποίηση ονομάζεται *κωδικοποίηση* (coding). Η κωδικοποίηση χωρίζεται γενικά σε δύο κατηγορίες: τη *γραμμική*, *μπλοκ* (block) κωδικοποίηση και τη *συγκεραστική* (convolutional) κωδικοποίηση, [1,2].

Στη μπλοκ κωδικοποίηση, οι ακολουθίες της εξόδου της πηγής πληροφορίας (μήνυμα πληροφορίας), με μήκος k απεικονίζονται σε δυαδικές ακολουθίες (*κωδικές λέξεις*) (codewords) με μήκος n έτσι ώστε ο *ρυθμός* του παραγομένου κώδικα να είναι ίσος με $\frac{k}{n}$ ανά εκπομπή, [1]. Ένας τέτοιος κώδικας ονομάζεται *κώδικας (n,k)* και αποτελείται από 2^k (πλήθος κωδικών λέξεων) κωδικές λέξεις με μήκος n, οι οποίες πολλές φορές συμβολίζονται με $c_1, c_2, .. c_{2^k}$.

Στους κώδικες μπλοκ, η απεικόνιση των διαφόρων ακολουθιών εξόδου της πηγής πληροφορίας στις κωδικές λέξεις πραγματοποιείται ανεξάρτητα και η έξοδος του *κωδικοποιητή* (encoder) εξαρτάται μόνο από την τρέχουσα ακολουθία εξόδου της πηγής (ακολουθία εισόδου στον κωδικοποιητή) μήκους k και σε καμία περίπτωση από τις προηγούμενες ακολουθίες εισόδου στον αποκωδικοποιητή, [1,2,5].

Αντίθετα, στους συγκεραστικούς κώδικες, αντίθετα, οι ακολουθίες εξόδου της πηγής μήκους k_0 απεικονίζονται σε ακολουθίες μήκους n_0, αλλά σε αυτή τη περίπτωση οι ακολουθίες εξόδου εξαρτώνται όχι μόνο από τις πιο πρόσφατες k_0 ακολουθίες εισαλλά και από τις $(L-1)\ k_0$ ακολουθίες εισόδου στον κωδικοποιητή, [1,2,5].

Μία πολύ απλή περίπτωση μπλοκ κωδικοποίησης είναι ο *απλός κώδικας επανάληψης* (simple repetition code). Σε έναν απλό κώδικα επανάληψης με τον οποίο επιθυμούμε να εκπέμψουμε τα δυαδικά σύμβολα “0” και “1” σε ένα δυαδικό συμμετρικό κανάλι (BSC), αντί να εκπέμψουμε τα “0” και “1”, εκπέμπουμε αντίστοιχα μία ακολουθία από “0” και “1” στην θέση αντίστοιχα του “0” και “1”. Το μήκος των δύο αυτών ακολουθιών επιλέγεται να είναι ένας *περιττός* (odd) αριθμός n. Έτσι, ένας απλός κώδικας επανάληψης μπορεί να παρασταθεί με την παρακάτω αντιστοιχία:

$$
\begin{aligned}
0 &\rightarrow \overbrace{00\ \ldots\ 00}^{n \text{ περιττός}} \\
1 &\rightarrow \overbrace{11\ \ldots\ 11}^{n \text{ περιττός}}
\end{aligned}
\tag{5.1}
$$

Η διαδικασία της *αποκωδικοποίησης* (decoding) βασίζεται σε μία πλειοψηφική απόφαση: αν η πλειοψηφία (σε πλήθος) των λαμβανομένων συμβόλων, είναι τα "0" τότε αποφασίζεται ότι το εκπεμπόμενο bit είναι το "0". Αν αντίθετα, η πλειοψηφία των λαμβανομένων συμβόλων είναι οι "1" τότε αποφασίζεται ότι το εκπεμπόμενο bit είναι το "1".

Στην διαδικασία της αποκωδικοποίησης, λάθος παρατηρείται όταν τουλάχιστον $(n+1)/2$ από τα εκπεμπόμενα bit έχουν ληφθεί λάθος. Αν το κανάλι επικοινωνίας είναι ένα BSC κανάλι, το οποίο εμφανίζει *πιθανότητα λάθους* (crossover probability) στο εκπεμπόμενο bit ίση με ε, τότε η πιθανότητα λάθους αποκωδικοποίησης για έναν απλό κώδικα επανάληψης (n,k) αποδεικνύεται ότι δίνεται από την παρακάτω σχέση, [4,5,11]:

$$
p_e = \sum_{k=(n+1)/2}^{n} \binom{n}{k} \cdot \varepsilon^k \cdot (1-\varepsilon)^{n-k} \tag{5.2}
$$

Για παράδειγμα, σε έναν απλό κώδικα επανάληψης με n=5 και $\varepsilon=0.001=10^{-3}$, η πιθανότητα λάθους είναι ίση με:

$$
p_e = \sum_{k=3}^{5} \binom{5}{k} \cdot 0.001^k \cdot (0.999)^{5-k} = 9.99 \times 10^{-10} \cong 10^{-9} \tag{5.3}
$$

Το αποτέλεσμα της σχέσης (5.3), δείχνει ότι αν εκπέμψουμε 5 bits στη θέση του ενός, τότε πετυχαίνουμε μείωση της πιθανότητας λάθους από $0.001(=10^{-3})$ σε 10^{-9}. Οπωσδήποτε μείωση της πιθανότητας λάθους έχει σαν αντίτιμο την μείωση του ρυθμού εκπομπής στο κανάλι και την αύξηση της πολυπλοκότητας στο συγκεκριμένο σύστημα. Δηλαδή, στο συγκεκριμένο παράδειγμα, ο ρυθμός εκπομπής μειώνεται από 1 bit κάθε μία φορά που χρησιμοποιούμε το κανάλι επικοινωνίας σε 1 bit κάθε 5 φορές που χρησιμοποιούμε το κανάλι επικοινωνίας. Η πολυπλοκότητα του συστήματος αυξήθηκε διότι η αποκωδικοποίηση σε έναν απλό κώδικα επανάληψης απαιτείται, στην περίπτωση αυτή, να γίνει μέσω ενός αποκωδικοποιητή πλειοψηφικής λογικής. Τέλος θα πρέπει να ειπωθεί ότι η πιθανότητα λάθους p_e μειώνεται με την αύξηση του n. Για παράδειγμα, για n=9, για τη προηγούμενη περίπτωση, έχουμε εφαρμόζοντας τη σχέση (5.2):

$$
p_e = \sum_{k=5}^{9} \binom{9}{k} \cdot 0.001^k \cdot (0.999)^{9-k} = 9.97 \times 10^{-16} \cong 10^{-15} \tag{5.4}
$$

Από το προηγούμενο παράδειγμα, μπορούμε να συμπεράνουμε ότι αν θέλουμε να μειώσουμε την πιθανότητα λάθους σε έναν απλό κώδικα επανάληψης στο μηδέν, θα πρέπει να αυξήσουμε το μήκος του κώδικα επανάληψης (n) στο άπειρο και αντίστοιχα να μειώσουμε έτσι το ρυθμό εκπομπής δεδομένων στο κανάλι στο μηδέν. Όμως ο C. Shannon απέδειξε, [6,7], ότι μπορούμε να διατηρήσουμε το ρυθμό εκπομπής σε τιμή μικρότερη από τη χωρητικότητα του καναλιού επικοινωνίας, με πιθανότητα λάθους η οποία να τείνει σε μηδενική τιμή $(p_e \rightarrow 0)$ αλλά χρησιμοποιώντας κατά την εκπομπή στο κανάλι επικοινωνίας, πιο σύνθετης δομής κώδικες, από ότι ένας απλός κώδικας επανάληψης.

5.3 Εργαστηριακό μέρος

Να γράψετε στο MATLAB πρόγραμμα υπολογισμού και αναπαράστασης της *πιθανότητας λάθους* στο αποκωδικοποιημένο bit για απλό κώδικα επανάληψης, στον οποίο σας δίνεται ότι η πιθανότητα λάθους σε ένα εκπεμπόμενο bit στο κανάλι επικοινωνίας είναι ίση με ε=0.3 και να γίνει γραφική παράσταση της πιθανότητας λάθους αποκωδικοποίησης p_e σε συνάρτηση με το *μήκος του κώδικα* (μήκος του μπλοκ) (block-length)(n). Να δώσετε τιμές στο n όλες τις περιττές τιμές από 1 έως και 61.

Πρόγραμμα

```
echo off
ep=0.3;
for i=1:2:61
   p(i)=0;
   For j=(i+1)/2:i
   p(i)=p(i)+prod(1:i)/(prod(1:j)*prod(1☹i-j)))*ep^j*(1-ep)^(i-j);
end
end
stem((1:2:61),p(1:2:61))
xlabel(‚n’)
ylabel(‚pe’)
title(‘Error probability as a function of n in simple repetition code’)
```

Εξήγηση Προγράμματος

Από την εκφώνηση της άσκησης μπορούμε να βρούμε την έκφραση για την πιθανότητα p_e εφαρμόζοντας τη σχέση (5.2) όπως παρουσιάζεται παρακάτω:

$$p_e = \sum_{k=(n+1)/2}^{n} \binom{n}{k} \cdot 0.3^k \cdot (1-0.3)^{n-k} = \sum_{k=(n+1)/2}^{n} \binom{n}{k} \cdot 0.3^k \cdot 0.7^{n-k} \qquad (5.5)$$

Αρχικά δίνουμε ως σταθερά τη συγκεκριμένη τιμή της πιθανότητας ε μέσω της μεταβλητής *ep*:

ep=0.3; (5.6)

Το μήκος του κώδικα n, δίνεται μέσω της μεταβλητής i. Η μεταβλητή i λαμβάνει όλες τις περιττές τιμές από 1 έως 61, για αυτό το λόγο και γράφουμε στο βρόχο επανάληψης που θα χρησιμοποιήσουμε την επόμενη εντολή:

for i=1:2:61 (5.7)

δηλαδή χρησιμοποιούμε ήμα αύξησης ίσο με 2.

Για να υπολογίσουμε το άθροισμα που απαιτείται στη σχέση (5.5), ορίζουμε έναν μονοδιάστατο πίνακα *p(i)* ο οποίος θα έχει ως περιεχόμενο μετά το τέλος κάθε εκτέλεσης του βρόχου επανάληψης την τιμή του p_e για την αντίστοιχη τιμή του n. Απαιτείται όμως μετά το τέλος κάθε εκτέλεσης του βρόχου επανάληψης, το περιεχόμενο του *p(i)* να μηδενίζεται για αυτό και δίνεται η παρακάτω εντολή:

p(i)=0; (5.8)

στην αρχή του βρόχου επανάληψης. Η μαθηματική έκφραση (συνδυασμοί των *n* ανά *k*) $\binom{n}{k} \overset{\text{ορ}}{=} \frac{n!}{k!\,(n-k)!}$ δίνεται στο MATLAB μέσω της έκφρασης:

$$\text{prod(1:i)/(prod(1:j)*prod(1:(i-j)))} \tag{5.9}$$

(το *i* αντιστοιχεί στο *n* και το *j* αντιστοιχεί στο *k*) όπου ισχύει για την έκφραση του *n!* στο MATLAB:

$$\text{n!=i!=prod(1:i)} \tag{5.10}$$

Παρατηρούμε ότι στο πρόγραμμα υπάρχουν δύο βρόχοι επανάληψης (ένας εξωτερικός και ένας εσωτερικός). Στον εξωτερικό βρόχο βρίσκεται η πιθανότητα p_e για συγκεκριμένο *n* ενώ στον εσωτερικό βρόχο κάθε φορά υπολογίζεται για το συγκεκριμένο *n* το άθροισμα που απαιτείται από τη σχέση (5.5). Η γραφική παράσταση της πιθανότητας λάθους p_e, για τις διάφορες τιμές του *n* από 1 έως και 61, πραγματοποιείται μέσω της εντολής:

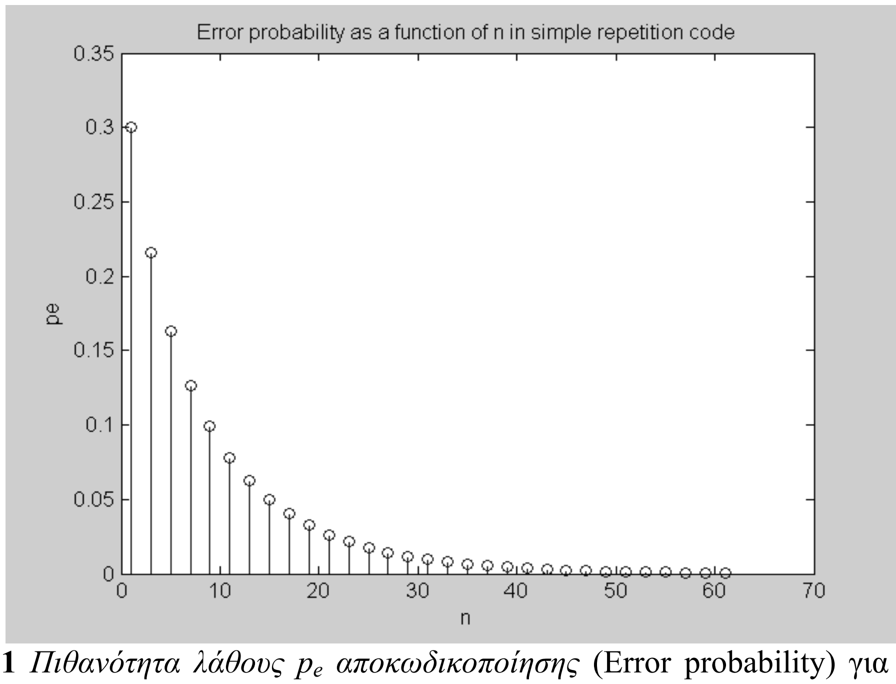
$$\text{stem((1:2:61),p(1:2:61))} \tag{5.10}$$

(γραφική παράσταση κατά βήματα)(διακριτές τιμές).

Τέλος, θέτουμε *ονομασίες* (*label*) στους άξονες x και y όπως και *τίτλο* (*title*) στη γραφική παράσταση μέσω των επόμενων εντολών:

```
xlabel('n')
ylabel('pe')                                                    (5.11)
title('Error probability as a function of n in simple repetition code')
```

Έξοδος προγράμματος



Εικόνα 5.1 *Πιθανότητα λάθους p_e αποκωδικοποίησης* (Error probability) για έναν απλό κώδικα επανάληψης σε συνάρτηση με το μήκος του κώδικα *n* (*Έξοδος από εκτέλεση του προγράμματος στο MATLAB*).

5.4 Πρόσθετες εργασίες

1. Εξηγείστε γιατί η αύξηση του n οδηγεί σε μείωση της πιθανότητας p_e;
2. Δώστε πρόγραμμα στο MATLAB το οποίο θα υπολογίζει την τιμή της πιθανότητας λάθους αποκωδικοποίησης για απλό κώδικα επανάληψης με μήκος n=7.
3. Αν για έναν απλό κώδικα επανάληψης απαιτούμε να έχουμε πιθανότητα λάθους αποκωδικοποίησης ίση με 0.05, ποιο θα πρέπει να είναι κατά προσέγγιση το μήκος του κώδικα για την περίπτωση ενός καναλιού με ε=0.3; Για το κανάλι αυτό, ποια είναι η τιμή της χωρητικότητας του καναλιού;
4. Αν για έναν απλό κώδικα επανάληψης απαιτούμε να έχουμε πιθανότητα λάθους αποκωδικοποίησης ίση με 0.05, ποιο θα πρέπει να είναι κατά προσέγγιση το μήκος του κώδικα για την περίπτωση ενός καναλιού με ε=0.1; Απαντήστε γράφοντας ένα αντίστοιχο πρόγραμμα στο MATLAB. Για το κανάλι αυτό, ποια είναι η τιμή της χωρητικότητας του καναλιού;
5. Να αποδείξετε με μαθηματικό τρόπο τη σχέση (5.2).