

ДОСЛІДЖЕННЯ КАСКАДНИХ КОДІВ ТА ТУРБОКОДІВ

В роботі проведений аналіз побудови завадостійких кодів, заснованих на послідовному або паралельному з'єднанні складових кодів. Представлено порівняння ефективності каскадних кодів і турбокодів з відокремленим використанням їх складових. Показано, що кращим характеристикам відповідає каскадне з'єднання декількох складових кодів. Досліджено алгоритм ітеративного декодування турбокодів, ефективність турбокоду у порівнянні зі згортковим. Зроблений висновок, що турбокод ефективніше послідовного каскадного на 2,5 дБ.

Ключові слова: каскадний код, турбокод, завадостійке кодування.

I.S. PYATIN, V.V. MISHAN, D.B. VERSHKOV

Khmelnytskyi National University

RESEARCH OF CASCADE CODES AND TURBO CODES

To ensure the minimum probability of mistaken reception of information, it is necessary to use interference-encoding. The purpose of the paper is to investigate the efficiency of cascading codes and turbo codes compared to other methods of noise immunity encoding. In this work an analysis of the construction of noise immunity codes based on the sequential or parallel connection of component codes is carried out. The method of implementing cascading codes consists in the sequential connection of two or more codes. The most common is a circuit with two levels of coding. The paper investigates the cascading coding system, where the Reed-Solomon code is used as an external code, and as an internal - convolutional code. The internal code is decoded by the Viterbi decoder, and the external one is the decoder of Reed-Solomon. It is found that cascading code is more efficiently converting at 5 dB at the level of BER = 10^{-4} . The turbo code is found to be 5.5 dB more efficiently. Simulink-model of the communication system with turbo coding and iterative decoding is researched. It is found that the use of five iteration of decoding a posterior probability (APP) turbo code is more effective than another iteration of 1.7 dB for BER = 10^{-6} . The comparison of the efficiency of cascading codes and turbo codes with the separate use of their constituents is presented. It is shown that the cascading connection of several constituent codes corresponds to the best characteristics. The efficiency of turbo code in comparison with conveying is studied. It is concluded that the turbo code is more efficient than the successive cascade at 2.5 dB.

Keywords: concatenated codes, turbo codes, antinoise coding.

Постановка задачі

Завдяки інтенсивному розвитку цифрових систем зв'язку актуальним завданням може бути досягнення мінімальної ймовірності помилкового приймання інформації. Ефективним способом досягнення цього є застосування завадостійкого кодування. Відповідно до теореми Шенона найкращим кодом є код, який передає повідомлення за нескінченно великий час, формуючи в кожен момент часу випадкові кодові елементи. У приймача є нескінченні версії повідомлення, спотвореного випадковим чином. З цих копій декодер повинен вибрати копію, найбільш близьку до переданого повідомлення. Першим кроком в побудові ефективних кодів було відкриття в 1964 р. каскадних кодів, другим кроком – відкриття в 1993 р. турбокодів [1].

Турбокод – паралельний каскадний або блоковий, або згортковий код, пов'язаний перемешувачем інформаційних символів, здатний виправляти помилки, що виникають при передачі цифрової інформації по каналу зв'язку з шумами (завадами). Турбокоди (ТК) відносять до випадкових кодів. Складові коди називаються компонентними. Під час ітеративного декодування інформація з виходу однієї ітерації надходить на вхід другої ітерації декодування і так далі. Кількість ітерацій декодування залежить від забезпечення заданої достовірності передачі інформації. Для декодування складових кодів звичайно використовують декодування по максимуму апостеріорної ймовірності (МАР-алгоритм). ТК дозволяють практично наблизитися до так званої межі Шеннона: відхилення становить не більше 0,5 дБ. Завдяки цій властивості ТК знаходять в даний час широке застосування в цифрових системах зв'язку різного призначення і застосування.

Аналіз досліджень та публікацій

Метод реалізації каскадних кодів полягає у послідовному з'єднанні двох або більше кодів. При застосуванні таких кодів необхідно збільшувати загальну довжину блоку при одночасному зменшенні складності устаткування [1]. Найбільш поширеною є схема з двома рівнями кодування. Одним з кодів, він називається зовнішнім, буває код Ріда-Соломона (РС); а в якості внутрішнього можна вибрати один з багатьох різноманітних кодів. Основний принципи каскадного кодування з двома рівнями показаний на рис. 1 [1].

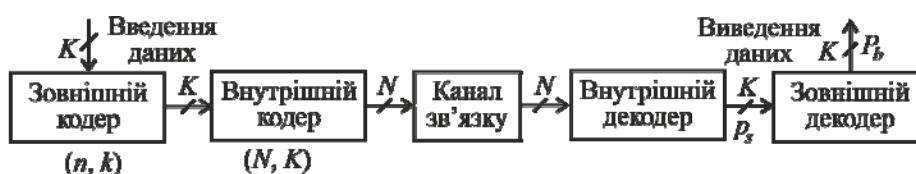


Рис. 1. Структурна схема системи з каскадним кодуванням

Припустимо, що канал має двійковий вхід. Зовнішнім кодом є недвійковий код, який використовує

K -бітові символи. Ці символи надходять в кодер зовнішнього коду від джерела даних, як показано на рис. 1. Передбачається, що зовнішній код є блоковим і його блок складається з n символів, причому k з них є інформаційними. Далі K – бітові символи, що виходять з кодера зовнішнього коду, кодується кодером внутрішнього коду. При цьому додаються $N - K$ перевірочних двійкових символів, так що довжина блоку внутрішнього коду дорівнює N . В канал надходять N -бітові символи і для можливості здійснення двійкової передачі паралельні символи переводяться в послідовні.

На приймальному кінці можуть використовуватися як жорсткі, так і м'які рішення. У будь-якому випадку результати цих рішень паралельно подаються на декодер внутрішнього коду. Цей декодер дає оцінку кожного K -бітового символу зовнішнього коду з імовірністю помилки p_s . Декодер зовнішнього коду має ймовірність помилки двійкового символу P_b .

Довжина каскадного коду $N' = nN$ двійкових символів, причому $K' = kK$ символів кодового слова – інформаційні, і швидкість коду $R' = rR = kN / nK$, де $R = K / N$ і $r = k / n$. Хоча загальна довжина коду дорівнює nN , каскадування забезпечує таку структуру коду, що декодування може здійснюватися за допомогою двох декодерів для кодів з довжинами N і n відповідно. Це дозволяє істотно знизити складність в порівнянні з тією, яка була б для отримання тієї ж ймовірності помилки при одному рівні кодування. Краще в якості зовнішніх вибирати коди Ріда - Соломона. Ці коди найбільш широко поширені, оскільки є кодами з максимальною відстанню ($d = n - k + 1$) і відносно просто реалізуються. Нехай система використовує зовнішній код Ріда - Соломона з декодуванням лише при наявності помилок. Це означає, що декодер внутрішнього коду в якості рішень видаватиме лише символи, не вказуючи ані стирань ані достовірності кожного символу. Найбільш перспективним є використання згорткового кодування з малою довжиною кодового обмеження і алгоритмом декодування Вітербі (з м'якими рішеннями демодулятора).

Розглянемо систему, внутрішнім кодом якої є згортковий код. K -бітові символи, що надходять із зовнішнього кодера, повинні бути знову переведені в послідовні для того, щоб можна було здійснити згорткове кодування. Однак при цьому слід зберігати блокову синхронізацію, тобто збіг початку блоку зовнішнього коду і кожного символу коду РС. Після декодування алгоритмом Вітербі, декодовані дані знову повинні бути переведені у K -бітові символи, що надходять на вхід декодера РС. Помилки на виході декодера Вітербі групуються у пакети, тому помилки в послідовних символах коду РС будуть некорельованими. Оскільки кореляція може викликати погіршення параметрів декодера Вітербі, слід застосувати перемешування. Це перемешування має бути таким, щоб ніякі два символи на виході декодера Вітербі, відстань між якими менше глибини декодування, не належали до одного кодового слова коду Ріда-Соломона.

Для визначення ймовірності помилки двійкового символу на виході декодера РС слід застосувати вираз ймовірності помилки двійкового символу на виході декодера Ріда-Соломона:

$$P_b < \frac{2^{K-1}}{2^K - 1} \sum_{j=t+1}^n \frac{j+t}{n} C_n^j p_s^j (1-p_s)^{n-j}, \quad (1)$$

де p_s – ймовірність помилки символу на вході декодера Ріда-Соломона, t – кількість помилок символів, що виправляє код Ріда-Соломона.

При цьому передбачається ідеальне перемешування, так що помилки символів на вході декодера РС вважаються незалежними. При кодах з $R = 1/3$ вираш від кодування збільшується приблизно на 0,6 дБ в порівнянні з кодами з $R = 1/2$, в той час як при кодах з $R = 3/4$ вираш зменшується приблизно на 1,2 дБ.

В наведених публікаціях дослідженню Simulink-моделей каскадних кодів і турбокодів та визначенню їх ефективності приділено недостатню увагу. Мета статті полягає у дослідженні ефективності каскадних кодів і турбокодів у порівнянні з іншими способами завадостійкого кодування

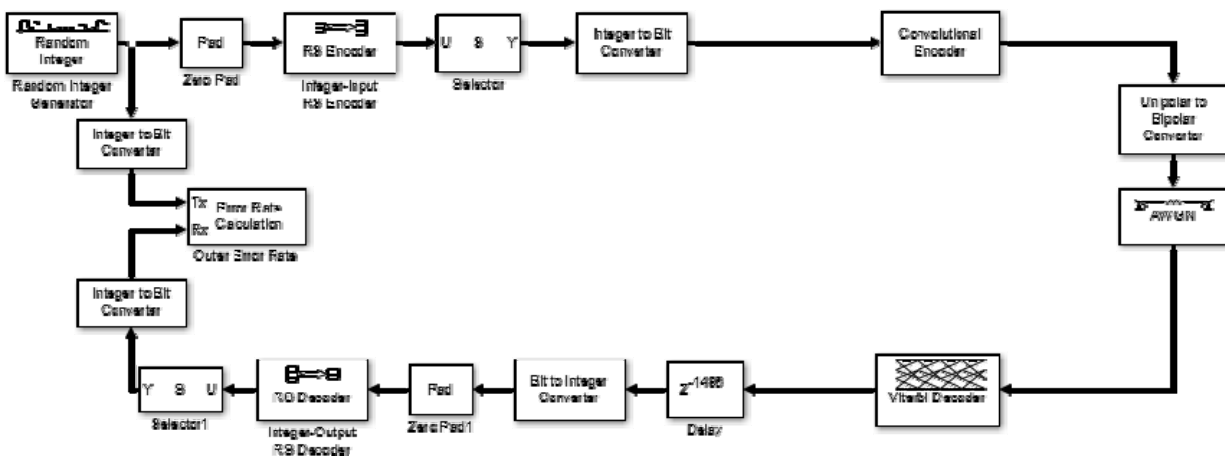


Рис. 2. Simulink-модель системи зв'язку з послідовним каскадним кодуванням

Дослідження ефективності каскадних кодів

Порівняємо за енергетичним критерієм системи зв'язку з каскадним коруванням і згортковим кодуванням. Simulink-модель системи зв'язку з послідовним каскадним кодуванням приведена на рис. 2. В якості джерела вхідного сигналу використаний Random Integer generator, що генерує рівномірно розподілені числа в діапазоні $[0, M-1]$, де M задається параметром Set size.

На рис. 3 зображено вікно налаштувань параметрів блоку Zero Pad.

Блок Pad розширює або звужує розміри сигналу на вході шляхом заповнення або вилучення даних за його стовпцями або рядками, або будь-якими зазначеними параметрами. Вилучення даних відбувається, коли ви вкажете розміри сигналу на виході, які коротше, ніж відповідні розміри сигналу на вході. Ви можете ввести значення блоку Pad в масці блоку або через вхідний порт.

Pad over – вказуються розміри, які необхідно розширити або звужити.

Dimensions to pad – вказується масив, елементи якого треба розширити або звужити.

Pad value source – вказується джерело значень блоку Pad.

Pad value – вказуються значення блоку Pad.

Коди Ріда-Соломона – це недвійкові циклічні коди, символи яких m -бітові послідовності, де m – додатне ціле число, більше 1. Коди Ріда-Соломона ефективні при виправленні пакетів помилок в каналах з пам'яттю, а також в каналах з великим набором вхідних символів. Особливістю кодів Ріда-Соломона є те, що до коду довжиною n можна додати два інформаційних символи, не зменшуючи мінімальної відстані. Такий розширений код має довжину $n + 2$ і ту же кількість символів контролю парності, що і початковий код. Simulink-модель системи зв'язку з згортковим кодуванням зображена на рис. 4.

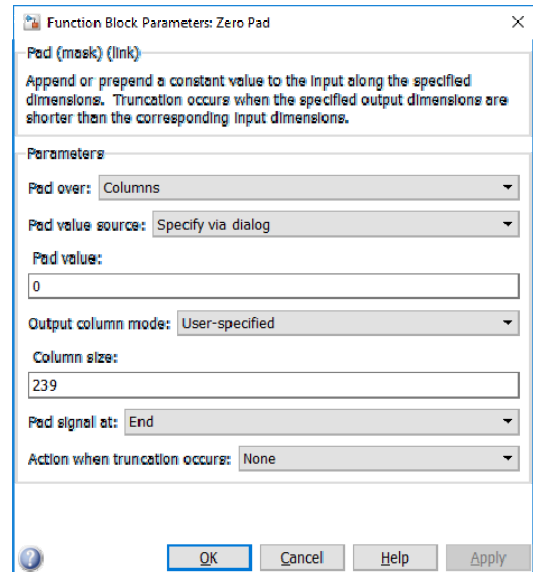


Рис. 3. Налаштування параметрів блоку Zero Pad

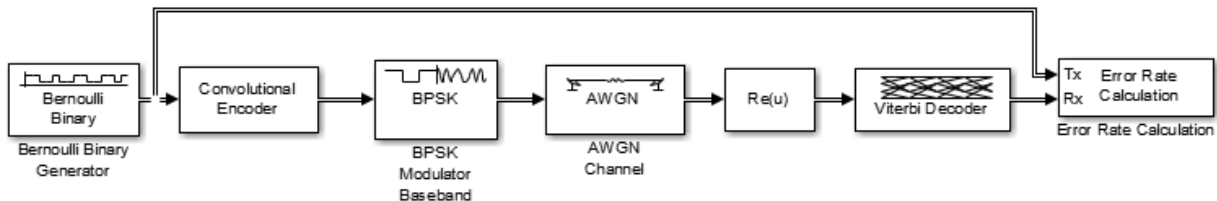


Рис. 4. Simulink-модель системи зв'язку з згортковим кодуванням

Згортковий код – це код, що виправляє помилки і використовує безперервну або послідовну обробку інформації короткими фрагментами (блоками). Згортковий кодер має пам'ять в тому сенсі, що символи на його виході залежать не тільки від (чергового фрагмента) інформаційних символів на вході, а й попередніх символів на його вході. Іншими словами, кодер представляє собою послідовну машину або автомат з кінцевим числом станів. Стан кодера визначається вмістом його пам'яті.

Згортковий кодер представляє собою пристрій, що приймає за кожний такт роботи в загальному випадку k вхідних інформаційних символів і видає на вихід за той же такт n вихідних символів, що підлягають передачі по каналу зв'язку. Параметром згорткового коду, що характеризує його завадостійкість, є мінімальна вільна відстань - d_c , яка визначається як мінімальна відстань за Хемінгом між послідовностями згорткового коду на довжині кодових обмежень по виходу. Кодове обмеження щодо виходу – це число символів на виході кодера, в формуванні яких бере участь один вхідний біт. Ефективність згорткового коду визначається тим, яким чином пов'язані суматори з комірками регістра зсуву [2].

Відношення $R = k/n$ називають відносною швидкістю коду. Вихідні символи, що створюються кодером на даному такті, залежать від k інформаційних символів, що надійшли на цьому і попередньому тактах. Вихідні символи згорткового кодера однозначно визначаються його вхідним сигналом і станом, що залежать від $m - k$ попередніх інформаційних символів. На рис. 5 наведені залежності ймовірності помилки на біт від відношення енергії біта до спектральної густини потужності шуму для каскадного коду і згорткового коду.

Основними елементами згорткового кодера є: регістр зсуву, суматори за модулем 2 і комутатор.

Якщо зафіксувати $BER=10^{-4}$, то каскадний код ефективніше згорткового на 5 дБ. Проведені також дослідження ефективності турбокоду. На рис. 6 наведені залежності ймовірності помилки на біт від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для турбокоду (turbo) і згорткового коду (conv), за якими можна зробити висновок, що турбокод ефективніше згорткового на 5,5 дБ.

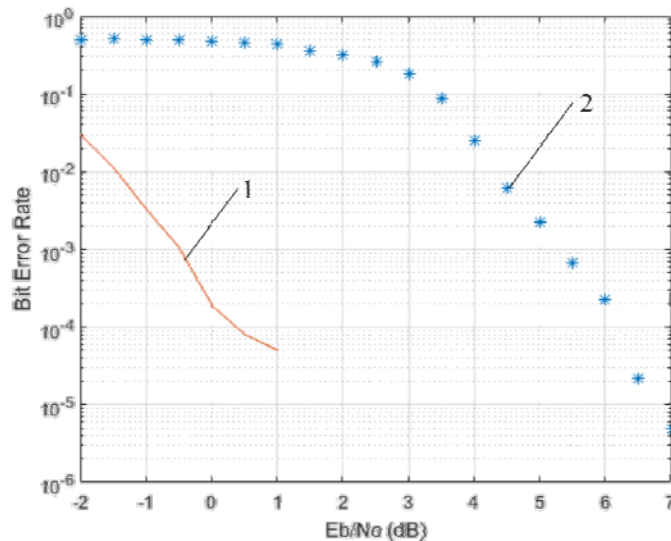


Рис. 5. Залежності ймовірності помилки на біт (BER) від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0): 1) каскадний код; 2) згортковий код

Дослідження ефективності турбокодів

Широке розповсюдження отримали паралельні каскадні згорткові турбокоди (ТК). Для побудови турбокодів використовують паралельне об'єднання рекурсивних систематичних згорткових кодів з перемешуванням і ітеративне декодування з м'яким входом і виходом.

Турбокодер – це схема паралельної конкатенації з декількома складовими конверторного коду. Перший кодер працює з використанням вхідної бітової послідовності, в той час як інші працюють з вхідними послідовностями, що отримані шляхом чергування вхідних бітів по довжині блоку. На рис. 6 наведені залежності ймовірності помилки на біт (BER) від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для турбокоду і згорткового коду. Блок Turbo Encoder використовує два ідентичних рекурсивних систематичних згорткових кодера з 8 станами. Об'єкт comm.ConvolutionalEncoder System використовує параметр «Завершення» для властивості TerminationMethod. Це відновлює коди до початкового стану всіх нулів для кожного кадру даних, який обробляє блок. Внутрішній блокувий перемешувач використовує попередньо обчислені індекси перестановок на основі заданого користувачем параметра довжини блоку коду. Підсистема перевпорядкування біт видаляє додатковий набір систематичних бітів з другого виходу енкодера і реалізує закінчення решітки. Залежність ймовірності помилки на біт від відношення E_b/N_0 для різної кількості ітерацій турбокоду приведена на рис. 7.

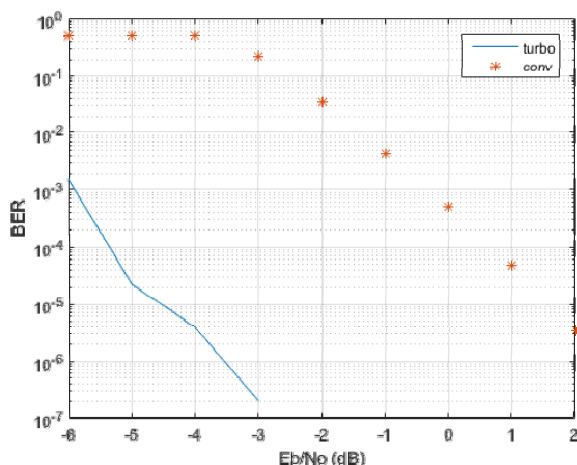


Рис. 6. Залежності ймовірності помилки на біт (BER) від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для турбокоду і згорткового коду

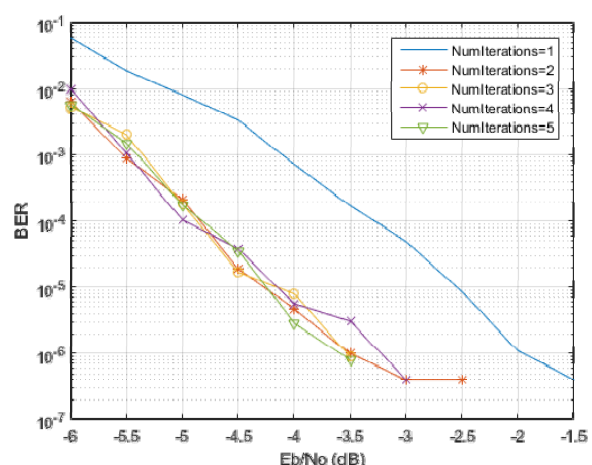


Рис. 7. Залежності ймовірності помилки на біт (BER) від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для різної кількості ітерацій турбокоду

Використання п'яти ітерацій декодування апостеріорної ймовірності (APP) турбокоду ефективніше однієї ітерації на 1,7 дБ для $BER=10^{-6}$.

Канал AWGN додає білий гауссовський шум до сигналу, який проходить через нього. Відносна потужність шуму в каналі AWGN описується такими величинами:

- Відношення сигнал / шум (SNR). Це фактичний вхідний параметр функції awgn.
- Співвідношення енергії біта і спектральної густини потужності шуму (E_b/N_0). Ця величина використовується BERTool і функціями оцінки продуктивності в цьому наборі інструментів.
- Співвідношення спектральної густини потужності енергії символу і енергії шуму (E_s/N_0).

Енергія біта дорівнює потужності сигналу S , помноженої на час передачі біта інформації T_b . N_0 – спектральна густина потужності шуму, що визначається відношенням потужності шуму N до смуги W .

Відношення сигнал-шум для цифрових систем визначається виразом:

$$\frac{E_b}{N_0} = \frac{S \cdot T_b}{N / W} s$$

Оскільки час передачі біта і швидкість передачі біта обернено пропорційні величини, то T_b можна замінити на $1/R$, де R – бітова швидкість, в результаті отримуємо:

$$\frac{E_b}{N_0} = \frac{S \cdot W_b}{N \cdot R}$$

Перемежувач являє собою пристрій, який змішує символи з кількох кодових слів так, що символи з будь-якого кодового слова дають випадковий ефект. Коли деперемежувач реконструює кодове слово, відновивши прийняту послідовність в її первісному вигляді, пакети помилок, що вносяться каналом виявляються розбиті і розкидані по кільком кодовим словам. Перемежувачі використовуються для того, щоб зменшити число кодових слів з малими відстанями в спектрі відстані коду, тобто генерує менше кодових слів з мінімальною відстанню Хеммінга.

Операції перемежовування / депережовування (interleaving / deinterleaving) означають перестановку символів в потоці даних. Ці операції широко застосовуються в системах зв'язку в тих випадках, коли помилки в каналі передачі виникають пакетами, тобто спотворюється велика кількість послідовних символів. Це характерно, скажімо, при завмираннях сигналу в радіоканалі або при наявності подряпин на компакт-диску. Перемежовування дозволяє перетворити пакетні помилки в одиночні, полегшивши тим самим завдання виправлення помилок блокам завадостійкого кодування / декодування.

В блоковому перемежувачі біти з виходу кодера записуються у вигляді двовимірної матриці в певному порядку. Найпростіший спосіб – записати біти у вигляді послідовних рядків. Після заповнення всієї матриці вхідними бітами починається етап зчитування. При цьому біти зчитуються в порядку, відмінному від порядку запису. Зчитування проводиться по стовпцях. В приймачі вхідні біти зберігаються в такій самій двовимірній матриці в тому ж порядку, в якому вони були зчитані в передавачі. Після заповненні матриці прийнятими бітами вони зчитуються в тому ж порядку, в якому були записані в матрицю перемежувача. Запис по рядках і зчитування по стовпцях — найпростіший спосіб розсіювання пакетів помилок. Можна визначити іншу послідовність запису і зчитування в перемежувачі і депережувачі. Обидва процеси повинні бути комплементарні один одному. Послідовність адрес і розмір матриці, що визначає так звану глибину перемежовування, повинні бути підібрані так, щоб пакетні помилки, які виникають в каналі після розсіювання виглядали випадковими. Для коректної роботи перемежувача і депережувача необхідна синхронізація. Якщо часові рамки зчитування передавача і приймача будуть зсунуті один відносно іншого, порядок бітів на виході з депережувача не дасть правильних результатів. Тому, послідовність, яка передається часто починається з короткого синхронізуючого слова. В момент часу коли одна з них використовується для зберігання вхідних даних, інша використовується для зчитування даних, записаних раніше. Після завершення операцій запису/зчитування змінюється функція матриць. Здатність розсіювати пакетні помилки визначається глибиною перемежовування. Тобто необхідно збільшувати розміри матриці, щоб ефективно боротися з пакетними помилками. Збільшення розмірів матриці приведе до використання більшого об'єму пам'яті та зростання вимоги щодо синхронізації передачі. Цікавим є питання використання матриць змінного розміру та пошуку алгоритмів запису/зчитування біт в матрицю.

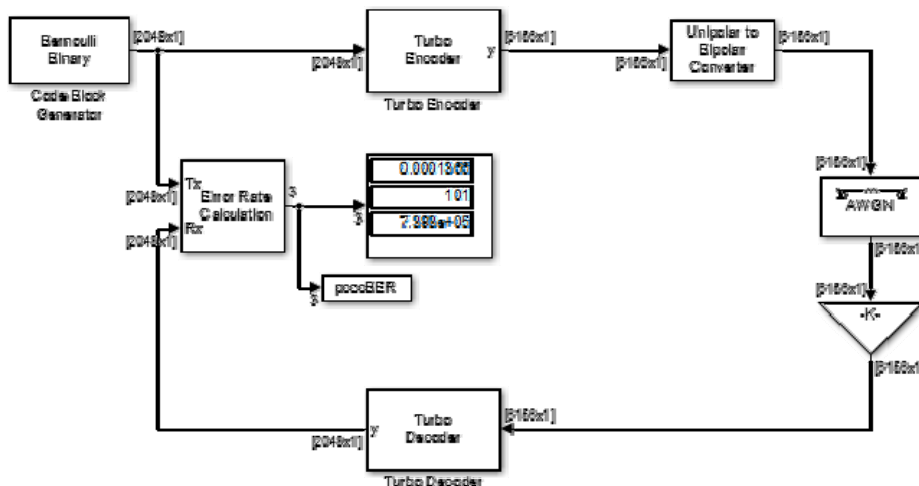


Рис. 8. Simulink-модель системи зв'язку з турбокодуванням

На рис. 8 приведена Simulink-модель системи зв'язку з турбокодуванням. В якості джерела вхідних

даних використано Bernoulli Binary Generator, що генерує випадкові двійкові числа, використовуючи розподіл Бернуллі. Розподіл Бернуллі з параметром p дає нуль з імовірністю p і один з імовірністю $1-p$. Розподіл Бернуллі має середнє значення $1-p$ і дисперсію $p(1-p)$. Імовірність нульового параметра вказує p , і може бути будь-яке дійсне число між нулем і одиницею.

На рис. 9 приведена Simulink-модель системи зв'язку з послідовним каскадним кодуванням. Виконано порівняння ефективності послідовного каскадного коду і турбо-коду. Послідовний каскадний код складається з зовнішнього згорткового кодера, перемежувача і внутрішнього згорткового кодера. Декодер каскадного коду побудований за методом максимуму апостеріорної ймовірності.

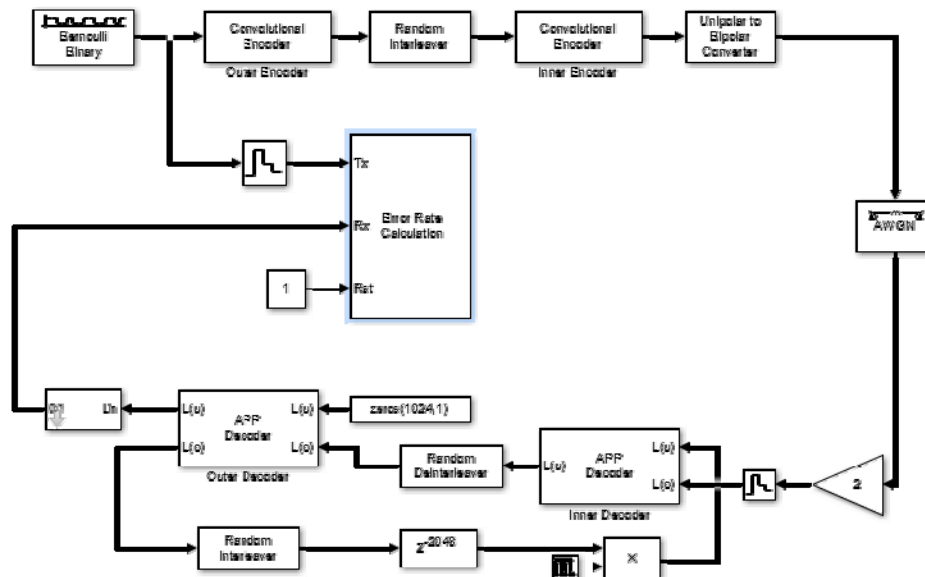


Рис. 9. Simulink-модель системи зв'язку з послідовним каскадним кодуванням

На рис. 10 наведені залежності ймовірності помилки на біт від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для каскадного коду і турбокоду.

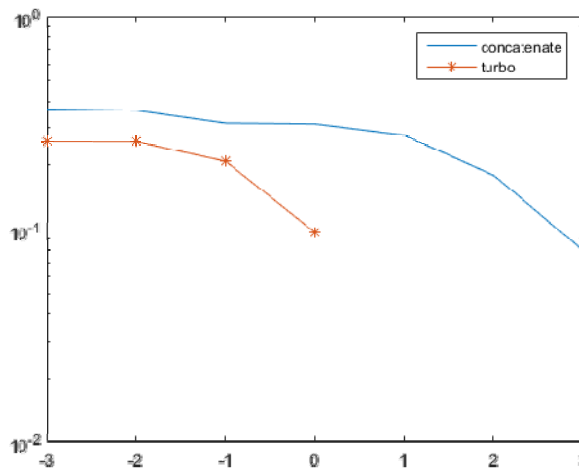


Рис. 10. Залежності ймовірності помилки на біт (BER) від відношення енергії біта до спектральної густини потужності шуму (E_b/N_0) для каскадного коду і турбокоду

З отриманого графіка залежності ймовірності помилки (BER) від відношення сигнал-шум можна зробити висновок, що турбокод ефективніше послідовного каскадного на 2,5 дБ.

Для ітеративного декодування схеми паралельного кодування Turbo Decoder використовує декодер апостеріорної ймовірності (APP) в якості складового компонента декодера. Кожен системний об'єкт comm.APPDecoder відповідає кодеру, який забезпечує оновлену послідовність значень логарифмічної правдоподібності для некованих бітів з прийнятою послідовністю за допомогою логарифмічних правдоподібностей для бітів кодованого каналу. Для кожного набору прийнятих послідовностей каналу декодер ітеративно оновлює логарифмічні ймовірності для некованих бітів до тих пір, поки не буде виконано критерій зупинки. У цьому прикладі використовується фіксована кількість ітерацій декодування, як зазначено в параметрі Кількість ітерацій декодування в блоці параметрів моделі. Число ітерацій за замовчуванням шість.

Властивість Termination Method для системного об'єкта декодера APP встановлено як «Terminated»

для відповідності кодеру. Внутрішній перемежувач декодера ідентичний внутрішньому перемежувачу, що використовує кодер. Він переупорядковує послідовності так, щоб вони були правильно вирівняні в двох декодерах.

Висновки

Турбокоди дозволяють практично наблизитися до межі Шеннона: відхилення становить не більше 0,5 дБ. Досліджено каскадний код з двома рівнями кодування. Один з кодів (зовнішній), є кодом Ріда-Соломона; а в якості внутрішнього – обраний згортковий код. Якщо зафіксувати $BER=10^{-4}$, то каскадний код ефективніше згорткового на 5 дБ. Досліджена ефективність турбокоду, що представляє собою паралельний каскадний або блоковий, або згортковий код, пов'язаний перемежувачем інформаційних символів. Турбокод ефективніше послідовного каскадного на 2,5 дБ. Досліджена залежність ймовірності помилки на біт від відношення E_b/N_0 для різної кількості ітерацій турбокоду. Послідовний каскадний код складається з зовнішнього згорткового кодера, перемежувача і внутрішнього згорткового кодера.

Література

1. Кларк Д. Кодирование с исправлением ошибок в системах цифровой связи / Д. Кларк, Д. Кейн ; [пер. с англ.]. – М. : Радио и связь, 1987. – 392 с.
2. Пятін І. С. Дослідження послідовного каскадного турбокодування / І. С. Пятін, Д.А. Макаришкін // Вісник Хмельницького національного університету. Технічні науки. – 2017. – № 3. – С. 142–145.
3. Пятін І.С. Моделювання цифрової системи зв'язку з завадостійким кодуванням / І.С. Пятін, В.В. Сергеев // Вісник Хмельницького національного університету. Технічні науки. – 2017. – № 6. – С. 89–91.

References

1. Klark D. Kodirovanie s ispravleniem oshibok v sistemah tsifrovoy svyazi / D. Klark, D. Keyn. Per. s angl. – M.: Radio i sviaz, 1987. – 392 s.
2. Pyatin I.S. Doslidzhenia poslidovnoho kaskadnoho turbokoduvannia / I.S. Pyatin, D.A. Makaryshkin // Herald of Khmelnytskyi National University. – 2017. – Issue 3. – С. 142-145.
3. Pyatin I.S. Modeliyvannia tsifrovoy sistemy zviazku z zavadostiykym koduvanniam / I.S. Pyatin, V.V. Sergeev // Herald of Khmelnytskyi National University. – 2017. – Issue 6. – С. 89-91.

Рецензія/Peer review : 6.10.2018 р.

Надрукована/Printed :22.11.2018 р.
Рецензент: д.т.н., проф. Бойко Ю.М.