

МІНІСТЕРСТВО ОСВІТИ И НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА АЕРОКОСМІЧНИХ СИСТЕМ УПРАВЛІННЯ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ О.М. Тачиніна

«_____» _____ 2021 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

**ЗА СПЕЦІАЛЬНІСТЮ 151 «АВТОМАТИЗАЦІЯ ТА
КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ»**

Тема: «Завадостійка система зв'язку для управління космічним роботом. Реалізація у пакеті MATLAB.»

Виконавець: студент групи СУ-401 Воронков Іван Янович

Керівник: професор Гаев Євгеній Олександрович

Нормоконтролер: Дивнич М.П.

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Навчально-науковий інститут аеронавігації, електроніки та телекомунікацій

Кафедра аерокосмічних систем управління

Напрямок 151 «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри АКСУ

_____ О.М. Тачиніна

«_____» _____ 2021 р.

ЗАВДАННЯ

на виконання дипломної роботи

Воронкова Івана Яновича

1. Тема дипломної роботи: «Завадостійка система зв'язку для управління космічним роботом. Реалізація у пакеті MATLAB.» затверджена наказом ректора від 12.04.2021р. №584/ст.

2. Термін виконання роботи (проекту): з 17.05. по 15.06.2021р.

3. Вихідні дані до роботи:

Використати можливості мови MATLAB та комп'ютерного моделювання для продовження розробки “цифрової лабораторії” з теорії інформаційних процесів та кодування.

Кінцева мета – забезпечення надійного дистанційного управління космічним роботом шляхом створення та випробування однієї з технологій «розумного коду», що виправляє помилки, які виникають у кодованих командах під час їх проходження зашумленого шляху.

4. Зміст пояснювальної записки:

Розділ 1: Огляд сучасних засобів управління роботами на великій відстані, застосування теорії інформаційних процесів та кодування.

Розділ 2: Підготовчі розділи теорії інформаційних процесів, Метод «розумного кодування».

Розділ 3: Випробування створеної програми, аналіз результатів.

5. Перелік обов'язкового графічного (ілюстрованого) матеріалу:

Блок-схеми алгоритмів і програм, таблиці і графіки результатів статистичного випробування алгоритмів.

6. Календарний план-графік

№	Завдання	Термін виконання	Відмітка про виконання
1	Огляд літературних джерел.	25.03.2021 — 03.04.2021	
2	Розробка РС-програм	15.04.2021	
3	Написання першого та другого розділів диплому.	30.04.2021	
5	Аналіз отриманих результатів. Написання третього розділу та висновку	29.05.2021	
6	Підготовка презентації і виступу, отримання рецензій.	08.06.2021 — 16.06.2021	

7. Дата видачі завдання: “_____” _____ 2021 р.

Керівник дипломної роботи _____

(підпис керівника)

Є. Гаєв

(П.І.Б.)

Завдання прийняв до виконання _____

(підпис випускника)

И. Воронков

(П.І.Б.)

Реферат

Пояснювальна записка до дипломної роботи по темі «Завадостійка система зв'язку для управління космічним роботом. Реалізація у пакеті MATLAB.» 45 сторінок, 3 малюнків, 11 посилань.

Ключові слова: ДИСТАНЦІЙНЕ УПРАВЛІННЯ РОБОТОМ, ІНФОРМАЦІЙНИЙ ПРОЦЕС, ПОВІДОМЛЕННЯ, КОДУВАННЯ, ПЕРЕДАЧА, ДЕКОДИРУВАННЯ, “РОЗУМНІ КОДИ”, ЦИФРОВА ЛАБОРАТОРІЯ.

Метою дипломної роботи було проектування коду, здатного самостійно знаходити помилки передачі та виправляти їх, що потрібно для дистанційного управління космічним роботом.

Об'єктом дослідження є інформаційні процеси та теорія кодування.

Предметом дослідження є метод створення і використання “розумного коду”.

Актуальність даної дипломної роботи

Ця дипломна робота є комплексною, паралельна дипломній роботі дипломника Миколенка Миколи. Однак, на відміну від його роботи, моя виконана за допомогою використання пакетів MATLAB, а не Java. Обидві наші роботи продовжують дипломників минулих років Бойко А. і Бабориگی А.[7,8].

По мірі росту виробництва, у будь-якій сфері, розумним рішенням стала часткова або, навіть, повна автоматизація процесів. Тому дуже важливо забезпечити ефективну середу передачі команд та використовуючи метод розумного кодування звести кількість помилок передачі до мінімуму.

Розділ 1 присвячується принципам управління роботами на відстані та застосуванню теорії інформаційних процесів та кодування.

Розділ 2 розкриває поняття простого кодування.

Розділ 3 описує процес реалізації «Розумного кодування» у пакеті MATLAB.

Зміст

Вступ	
Розділ 1 Теорія інформаційних процесів та кодування	
1.1 Теорія інформаційного процесу	
1.2 Концепція передачі інформації	
1.3 Прості коди.	
1.4 Концепція надмірності та коди із надмірністю	
Розділ 2 Завадостійке кодування повідомлень, проблема їх викривлення	
2.1 Причини виникнення шумів та їх види	
2.2 Коригувальні коди для виявлення помилок	
2.3 Роль в авіації	
Розділ 3 Розробка алгоритмів у пакеті MATLAB, здатних виправляти помилки	
Блок – схема програми	
Кодуючий пристрій	
Генератор помилок	
Декодуючий пристрій	
Висновки	
Література	

Вступ

Управління, теорія управління, космічні роботи.

Розглянемо поняття що ж таке робот в цілому. Робот – це створений по принципу живого організму автоматичний пристрій, головною функцією якого є виконання виробничих або інших операцій за певною програмою, заздалегідь порученою людиною.

За останнє століття роботи не тільки еволюціонували, вони стали частиною нашого повсякденного життя. Після того, як у 1920 році у світ вийшла п'єса чеського письменника Карела Чапека «R.U.R.» про штучних людей – поняття слова «робот» міцно увійшло у побут.

Протягом подальших десятиліть людство не зупинилося у розвитку робототехніки. Робилися видатні відкриття в самих різних дисциплінах, на яких саме й базується робототехніка, а саме: кібернетика, електроніка, механіка, інформатика, кібернетика, тощо. Приблизно на початку 30 – х років ХХ століття були винайдені та сконструйовані перші у світі андроїди, вони були здатні рухатися та вимовляти прості фрази. Саме роботи американського інженера Генрі Форда по створенню виробничої лінії з певною мірою автоматичності стали поштовхом для конструювання перших механізмів з маніпуляторами у 1937 році у США. На кінці 40 – х років у Радянському Союзі з'явилися повністю автоматизовані виробництва поршнів для тракторних двигунів. Всі процеси виконувались виключно автоматично, від завантаження сировини та до пакування готової продукції. У 1950 – х роках були створені маніпулятори, які були здатні копіювати рухи рук людини. Згодом їх почали використовувати для роботи з радіоактивними матеріалами і не тільки.

На початку 2000 – х велика кількість компаній по виробництву

робототехніки представила світу нове покоління гуманоїдних роботів. Наприклад робот-андроїд Asimo. У масове виробництво були запуснені побутові роботи – пилососи та навіть кіберсобаки. Було також представлено розробку вчених Стенфордського університету – робота STAIR, який був здатен до прийняття нестандартних рішень, керуючись закладеними в нього знаннями. Роботів почали використовувати в медицині для допомоги у проведенні складних хірургічних операцій [1].

Простіше кажучи, людство створило собі унікальний інструмент, котрий спрощує та суттєво прискорює як дослідницькі так і виробничі процеси. У виробничих процесах в більшості випадків сучасні роботи являються маніпуляторами, а саме так званими «руками» для переміщення або складання та структурування певних ресурсів.

Кажучи про роботів, яких використовують в дослідницьких сферах – це різноманітні датчики, мобільні пристрої та роботи, які здатні працювати у середовищах, перебування в яких небезпечно або навіть неможливе для людини, наприклад у водних глибинах, зонах термального впливу або навіть у космосі. Але слід пам'ятати, що керування такими роботами є виключно дистанційним. Це означає, що інформація підлягає дії так званих шумів середовища, через яке проходить повідомлення й його подальшому спотворенню. Тому важливою частиною роботи з такими роботами стали методи кодування й захисту інформації від спотворення. Прикладом такого робота та необхідності в забезпеченні надійного каналу передачі даних хочу привести вертоліт NASA «Ingenuity» для дослідження поверхні Марсу та виявлення слідів існування води на планеті.

Було проведено кілька тестових польотів, та під час шостого польоту при передачі інформації сталося спотворення. Як потім було зазначено в деталях польоту, приблизно на 54 секунді у програмі бортового комп'ютера сталася помилка, через яку таймер камери став

показувати помилковий час, що в свою мить досить важливо для системи навігації робота. Ось як про це повідомлено:

"Шостий політ відбувся 22 травня, - зазначив відомство. - Вертольоту була передана команда піднятися на висоту 10 м і переміститися на 150 м в південно-західному напрямку зі швидкістю 4 м / с. Далі він повинен був просунути ще на 15 м на південь, виробляючи при цьому кольорову фотозйомку місцевості, а потім ще на 50 м на північний схід і здійснити посадку. Як показала телеметрія, перший етап - 150 м - пройшов без перешкод. Але потім щось сталося: Ingenuity почав то втрачати, то набирати швидкість і при цьому гойдатися. І так тривало до самого кінця польоту. Перед посадкою, яка завершилася без проблем, бортові датчики показували, що відхилення по крену і тангажу у вертольота склали понад 20 градусів ", - вказав NASA, підкресливши, що апарат зумів опуститися в 5 м від заданого місця [2].

Задля забезпечення надійного управління дистанційного робота потрібен надійний, безперебійний зв'язок і виконання тих команд та завдань, які ми надсилаємо зі станції управління. Для вирішення таких завдань слід застосувати арсенал технологій теорії інформації і кодування. Цьому й присвячена дана дипломна робота. Розглянемо це більш детально.

Досить важко уявити сучасний світ без гаджетів або інших пристроїв передачі інформації. Всі вони здійснюють інформаційних процес. Вчені й досі припускають, що саме Клауд Шеннон, американський математик, започаткував створення теорії інформації своєю публікацією «Математичної теорії зв'язку». Інформаційні процеси стали дуже природними для нас, оскільки людство кожен секунду обмінюється, отримує та використовує інформацію. Засоби зв'язку, соціальні мережі, банківська сфера, Інтернет - все це наші сучасні реалії.

Кожен день ми робимо маленький крок у напрямку високих технологій. Кожного року винаходяться та виробляються все потужніші комп'ютерні компоненти, обчислюючі можливості котрих вже в найближчому майбутньому не знатимуть меж.

Дуже важливу роль інформаційні процеси відіграють і в авіації, адже аеронавігація повністю покладається на точність та швидкість передачі даних, оскільки від цього залежить життя як пасажирів так і екіпажу.

Розділ 1 Теорія інформаційних процесів та кодування

Як було вказано раніше, надійне управління забезпечується, зокрема, надійністю інформаційного зв'язку. Тому, для більш глибокого дослідження наведемо потрібну інформацію.

1.1 Теорія інформаційного процесу

Обговорюючи тему інформаційних процесів, ми, очевидно, торкаємося й теми теорії інформації. Доленосною подією заснування такої дисципліни стала публікація статті Клода Шеннона під назвою «Математична теорія комунікації» у журналі BELL SYSTEM липня 1948 року.

В своїй роботі, завершену у лабораторіях Белл до кінця 1944 року, американський математик вперше запропонував якісну та кількісну модель спілкування як статичний процес, котрий лежить в основі теорії, регламентуючи, що «Основною проблемою спілкування є відтворення в одній точці, точно або приблизно, повідомлення, вибраного в іншій точці [3].

Попередником статті Шеннона була робота Гаррі Найквіста «Фактори, що впливають на швидкість телеграфу». В цій роботі вперше розглядалася кількісна оцінка інформації, а також швидкості лінії, з якою дані можуть передаватися системою. Звідси було отримано співвідношення:

$$W = \frac{K}{\log m'}$$

					НАУ. 21.01.55 000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Воронков І.Я..			<i>Завадостійка система зв'язку для управління космічним роботом. Реалізація у пакеті</i>	Літ.	Аркуш	Аркушів
Перевір.		Гаєв Е.А.						
Реценз.								
Н. Контр.		Дивнич М.П.						
Затверд.		Тачиніна О.М.						
						12 ФАЕТ- 401 гр.		

Де W – це швидкість передачі інформації, m – кількість рівнів напруги на кожному часовому кроці. K – константа.

Під час дослідження інформаційних процесів постійно фігурують три поняття, а саме:

а) Первинне джерело інформації – визначається як елемент середовища, відомості про який є об'єктом інформації.

б) Сигнал – визначається як зміна фізичної величини, яка використовується для відправки даних.

в) Споживач інформації – елемент, що отримує та використовує отриману інформацію.

При моделюванні процесів передачі інформації завжди потрібно мати на увазі, що данні відсилаються за допомогою повідомлень. Для спрощення та прискорення процесу обробки та відсилення інформації використовують кодуєчий та декодуєчий пристрої.

Кодуючий пристрій – це пристрій для перетворення інформації в цифровий сигнал відповідно до визначеної таблиці кодування. Слід пам'ятати, що кодуванню підлягає лише та інформація, яка представлена у формі дискретного сигналу.

Декодуєчий пристрій – це пристрій для перетворення цифрового коду у вигляді, зрозумілого для отримувача. Вхідна закодована комбінація на виході перетвориться в сигнал.



Рис 1.2 Загальна схема передача інформації в інформаційній системі [10]

Канал зв'язку – це засіб передачі сигналу між двома пристроями на відстані.

Каналами зв'язку можуть бути велика кількість речей: вода для підводного човна, провід для аналогової телефонії, атмосфера у разі оптичного зв'язку. У нашому випадку це космічний простір, тобто наш космічний робот отримує команду, яка проходить у вигляді електромагнітної хвилі через радіоефір.

Також пристрої передачі поділяють на два типи, а саме:

а) Симплексні, тобто передача інформації відбувається лише в одну сторону. Прикладом такого каналу є радіо та телебачення.

б) Дуплексні, тобто інформація в таких каналах здатна передаватися в обох напрямках. Прикладом такого каналу є різноманітні телефони, рації, тощо.

По каналах може передаватися багато повідомлень в один й той самий час. Максимальна кількість повідомлень залежить від пропускної здатності каналу.

В свою чергу пропускна здатність визначається максимальною

кількістю надісланих символів, котрі передаються по каналу за відсутністю перешкод (шумів).

Кожна система зв'язку може привести до помилкового прийому сигналу через вплив на неї шумів та спотворень. Для виправлення плачевної ситуації з виникаючими після шумів помилками в сигнал вводиться спеціальний код, що дозволяє приймаючій стороні ідентифікувати та виправити певну кількість помилок. Виправлення помилок та контроль цілісності переданих даних – це дуже важливе завдання на багатьох рівнях роботи з даними.

У системі зв'язку існують три стратегії боротьби з цією проблемою, а саме:

А) Виявлення помилок у блоках і запит повторної передачі пошкоджених.

Б) Виявлення помилок і відкидання пошкоджених блоків.

В) Виявлення помилок у блоках і їх подальше виправлення за допомогою завадостійкого кодування, що, власне, і є головною темою цієї дипломної роботи.

Як можна зрозуміти, ця дисципліна є відносно молодого і її головною метою є створення набору законів, інструментів та закономірностей як результат управління інформаційними процесами.

1.2 Концепція передачі інформації

Для подальшого вивчення слід ознайомитися з декількома поняттями. Поняття інформації та даних пов'язані. Перш за все, дані – це сукупність значень якісних або кількісних змінних. Інакше кажучи, дані стосуються факту, що інформація або знання представлені чи закодовані в формі,

придатній для операцій. Як правило, дані вимірюються, збираються та передаються для подальшої обробки.

Дані можуть бути передані різними способами, способи класифікуються за тисячами критеріїв. За середовищем, яким воно передається(вода, земля, повітря, дрiт), за аспектом синхронізації, за типом кабелю(оптичне волокно, мідь), за типом підключення або хвилями, що використовуються для передачі (Звукові, інфрачервоні, світлові, тощо).

Неопрацьовані дані – це набір символів або цифр, які були отримані від вимірювальних приладів, до відправки. Основною причиною обробки вихідних даних є усунення відхилень та помилок, адже, наприклад, фізика повністю базується на даних, отриманих в якість результатів вимірювання. Обробка інформації розподілена на декілька етапів.

Інформація – це сутність або форма, яка відповідає на питання або невизначеність. Таким чином, це пов'язано з даними та знаннями, оскільки дані представляють значення, що приписуються параметрам, а знання означають розуміння реальних речей та абстрактних понять[8].

Інтерфейс – діалог між користувачем та пристроєм шляхом перетворення сигналів у той тип, який є зрозумілим для користувача.

Обмін інформацією є дуже важливим процесом для збільшення знань. Цей процес потребує відсилання інформації одержувачу для її подальшої інтерпретації. Для відправки вона кодується у різні форми, вона може передаватися у вигляді кодованої послідовності знаків або через послідовність сигналів. Також інформація може бути зашифрованою, це робиться для її безпечного зберігання.

Зменшення невизначеності є одним із завдань інформації. Ця невизначеність вимірюється ймовірністю настання і є обернено пропорційною. Це, в свою чергу, означає, що чим більше невизначена подія – тим більше інформації потребує її вирішення. Типовою одиницею вимірювання інформації є біт.

Теорія передачі інформації є частиною теорії інформації. Предметом вивчення цієї теорії є набуття оптимальних методів передачі повідомлень. Вони передаються за допомогою сигналів, які мають певні фізичні властивості. Отже, сигналом може бути будь-яка зміна початкового стану об'єкта, здатного викликати реакцію людини чи пристрою. В часі та просторі сигнали поділяються на статичні та динамічні. Статичним сигналам властиві постійні зміни в об'єкті. Динамічні сигнали – сигнали, що представляють постійні зміни в об'єкті. За структурою сигнали поділяються на безперервні та дискретні.

Основними параметрами, які визначають сигнал є тривалість сигналу, стійкість, потужність та ширина частотного спектру. Для надійної передачі ці характеристики враховуються, оскільки одна одну доповняє. Відсутність потужності можна компенсувати збільшенням тривалості і так далі.

Тривалість сигналу – характеристика, що вказує час сигналу в каналі.

Частотний спектр сигналу – характеристика, що показує смугу частот, на якій сигнал передається по каналу. Теоретично, ширина спектра сигналу необмежена. З практики запису на магнітні носії ми знаємо, що чим ширший діапазон частот використаного обладнання, тим вища якість запису. Але майже кожен конкретний тип передачі має свій спектр, при якому спотворення сигналу лежить в допустимих межах.

Стійкість розраховується за даною формулою:

$$A = 10 \log_{10} \frac{P_{\min \text{ signal}}}{P_{\text{noise}}}$$

Середня потужність сигналу показує потужність, яку передає обладнання, коли сигнал проходить через канал зв'язку.

$$D_s = \log P_s/P_i.$$

Гучність сигналу – сукупна характеристика, яка вказує на умови, які канал зв'язку повинен задовольняти для якісної передачі.

$$V_s = T_s F_s D_s$$

Схожа характеристика існує і для каналу зв'язку. Характеристика ємності каналу:

$$V_c = T_c F_c D_c ,$$

Де T_c - час використання каналу; F_c - пропускна здатність передачі сигналів по заданому каналу; D_c - динамічний діапазон рівнів сигналу, який даний канал зв'язку може забезпечити.

Якісна передача сигналу можлива, коли задовольняються наступні критерії:

$$T_s \leq T_c;$$

$$F_s \leq F_c;$$

$$D_s \leq D_c$$

Це умови повної координації каналу зв'язку з самим сигналом.

Проте, сучасне обладнання має можливість узгоджувати сигнал з каналом навіть за відсутності виконання всіх трьох умов, але для цього повинна виконуватися умова, без якої якісна передача сигналу неможлива:

$$V_s \leq V_c;$$

1.3 Прості коди.

Хотілось би спочатку розібратися, що ж таке кодування, для чого воно потрібне, його переваги та недоліки.

Кодування повідомлень – це процес зміни послідовності сигналів, іншою послідовністю, більш зрозумілу для об'єкту, котрий саме споживає цю інформацію. Слід пам'ятати, що під часу кодування повідомлення змінюється його вигляд, але зміст залишається без зміни.

Двійкове кодування використовується в сучасних комп'ютерах. Цифра 0 та 1 у двійковому коді називається бітом.

Кодування тексту означає заміну його символів двійковим кодом. У системах використовують коди фіксованої довжини. Через це відпадає необхідність використовувати спеціальні позначення для відокремлених кодів. Для подання текстів використовують коди довжиною вісім бітів. Довжина у вісім бітів дозволяє утворити $2^8 = 256$ різних комбінацій кодів., що в свою чергу дає змогу призначити індивідуальні коди літерам алфавіту – як великим, так і малим, розділовим знакам, цифрам, тощо.

При кодуванні чисел також застосовують двійковий код. Десяткове число можливо просто записати у двійковій системі числення. Спосіб кодування числа і довжина його коду залежить від того, чи є наше число додатнім чи від'ємним, цілим або дробовим. Таким чином довжину коду варується від одного до восьми байтів. Це дозволяє відтворювати в комп'ютері цілі числа в великому діапазоні та працювати з

багатозначними дробовими числами [11].

При кодуванні графічних зображень, вони створюються на екрані мозаїчним способом, а саме, складається з великої кількості кольорових крапок(пікселями). Кожен піксель має свій колір. У сучасних комп'ютерах застосовують 16 або 32 біти для подання кольору. Задається зображення, комп'ютер запам'ятовує таблицю кольорів цих самих пікселів.

Одною із самих відомих кольірних моделей – RGB. Колірна модель RGB є три кольори спектра RED, GREEN, BLUE, вона займає $3 \times 8 = 24$ біта на піксель. Дані зображення кодуються за допомогою двійкового коду. Поєднуючи ці кольори отримують різні відтінки, які під час дискретизації використовуються для створення різних палітр, що наведено в таблиці[11].

Система кодування RGB

Кольори	R	G	B
Білий	1	1	1
Жовтий	1	1	0
Фіолетовий	1	0	1
Червоний	1	0	0
Блакитний	0	1	1
Зелений	0	1	0
Синій	0	0	1
Чорний	0	0	0

1.4 Концепція надмірності та коди із надмірністю.

Дуже рідко наші коди несуть максимально можливий обсяг інформації, тому навантаження на елементи звичайно менше. Елементи кодів, які представляють повідомлення, недовантажені, а саме повідомлення має надмірність. Саме поняття надмірності в теорії інформації вводиться для оцінки інформаційного резерву коду, з якого створюється повідомлення. За

походженням існують штучні та природні надмірності. Штучна надмірність характерна для вторинних алфавітів, а природна для первинних відповідно.

Існують дві основні характеристики символів з яких було утворено повідомлення: ймовірність появи символів та залежність між символами.

Існує залежність між надмірністю та швидкістю передачі та стійкістю до перешкод. Зменшуючи надмірність повідомлення, можна збільшити швидкість його передачі. Збільшуючи надмірність повідомлення, можна зменшити ймовірність його спотворення через шумові перешкоди.

Введення поняття надмірності дозволило створити методи кодування з можливістю виявлення помилок. На сьогоднішній день уже створено безліч кодів, які здатні виявляти та виправляти помилки. Вони постійно вдосконалюються та покращують співвідношення між надмірністю, швидкістю передачі та стійкості до шумів.

Висновки до розділу 1

У цьому розділі ми описали основні поняття теорії інформації та інформаційних процесів, визначили зв'язки дисципліни з іншими науками. Дали визначення всім основним поняттям, які найчастіше фігурують у дослідженні інформаційних процесів. Написали короткий курс історії, де обговорили те, як зароджувалася ця сфера науки та хто зробив у цьому перші кроки. Також було звернено увагу на основні проблеми при передачі даних.

Було розглянуто поняття кодування: що це процес перетворення повідомлення на впорядкований набір символів та знаків.

У наступному розділі будуть розглянуті та проілюстровані шляхи, винайдені вченими в різні часи, для їх вирішення.

Розділ 2. Завадостійке кодування повідомлень, проблема їх викривлення

Саме для забезпечення стабільності системи та зведення впливу шумів на наше повідомлення до мінімум і були винайдені коригувальні коди.

2.1 Причини виникнення шумів та їх види

В каналах зв'язку перешкодами називають напругу або струм стороннього походження, який з'являється у них та обмежує якість та дальність передачі.

Залежно від характеру впливу та джерела походження, перешкоди діляться на власні перешкоду каналу зв'язку та зовнішні – вплив зовнішніх електромагнітних полів.

Причиною виникнення власних перешкод або шумів є наявність їх джерел, що знаходяться в даному каналі передачі. Ці шуми визначаються такими факторами, як пульсація напруги джерела живлення, короткими замиканнями, мікрохвильовими шумами тощо.

Проте, в даній дипломній роботі нас цікавлять саме зовнішні перешкоди, оскільки ми відправляємо команди нашому роботу через космічний простір.

					НАУ. 21.01.55 000 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>	Воронков І.Я.				<i>Завадостійка система зв'язку для управління космічним роботом. Реалізація у пакеті</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір.</i>	Гаев Е.А.							
<i>Реценз.</i>								
<i>Н. Контр.</i>	Дивнич М.П.							
<i>Затверд.</i>	Тачиніна О.М.							
						ФАЕТ- 401 гр.		22

Зовнішні перешкоди діляться на:

- Атмосферні. До атмосферних відносяться шуми, причиною виникнення яких були атмосферні явища(магнітна буря, північне сяйво, гроза, тощо).
- Індустріальні. Виникають в результаті впливу електромагнітних полів електричних пристроїв (випромінювання радіостанцій, електрообладнання підприємств, ліній електропередачі, тощо).
- Космічні. Джерелом космічних перешкод є випромінювання Сонця та зірок у відповідних діапазонах частот.

2.2 Коригувальні коди для виявлення помилок

Для виправлення помилок застосовуються коригувальні коди. Вони використовують збитковість. При записі, у дані додають спеціальним чином структуроване контрольне число, яке згодом, при отриманні даних, використовується для виявлення та виправлення помилок. Кількість помилок, які можливо виправити таким чином обмежена та цілком залежить типу застосованого коригувального коду.

З виправляючими кодами щільно пов'язані коди для виявлення помилок. Якщо перші можуть не тільки виявити присутність помилки в повідомленні, а й одразу виправити її, то другі можуть бути використані лише для виявлення помилки.

Завадостійкі коди ділять на два класи. На неперервні та блокові, що в свою чергу поділяються на роздільні та нероздільні. У роздільних кодах дуже чітко присвоєні ролі окремих символів. Деякі є виключно інформаційними, а інші перевірними, за допомогою яких і відбувається виправлення помилок. Велику кількість роздільних кодів

складають собою систематичні. В результаті проведення простих лінійних операцій над символами повідомлення визначаються перервні символи.

А в результаті перевірки груп символів на парність перевіряється інформація про наявність помилки, її позицію.

Суть блочного завадостійкого коду можна описати наступним чином. Нехай закодовані дані розділяються на частини з певною довжиною біт. (b), які згодом перетворюються в кодові слова довжиною (n) біт. Якщо, вихідне значення b наш код залишає таким самим, і додає $n - b$ перевірочних, то такий код буде систематичним.

Блоковий код можливо задати багатьма способами, але слід пам'ятати, що код потрібно написати так, щоб він виправляв максимальну кількість помилок, був простий у кодуванні та декодуванні та май малу надмірність. Саме тому я вирішив зупинитися на заданні коду таблицею. Таким чином його просто реалізувати, адже всі можливі варіанти кодування та декодування вже прописані, та кількість можливих виправлених помилок практично максимальна.

Неперервні коди не поділяють інформацію на фрагменти, як блокові, вони працюють з даними як з суцільним потоком.

Авіація займає поважну роль як у військових, так і транспортних промисловостях. Високий ажіотаж навколо такого ефективного способу ведення промисловості дає змогу виділяти на авіацію колосальний обсяг фінансування, що в свою чергу використовується для розробки та проектування більш ефективних компонентів керування (двигунів, систем управління, тощо). Результатом такого циклу є повне

оцифрування галузі. В сучасних літаках використовується величезна кількість цифрових пристроїв. Усі ці пристрої потрібні відповідати найсуворішим вимогам.

2.3 Роль в авіації

В останні роки безпілотні машини все швидше займають своє місце в нашому житті, вони були випробувані та впроваджені в багатьох сферах промисловості. З таким ростом популярності використання в ближньому майбутньому у світі може з'явитися перший безпілотний літак для перевезення пасажирів. Проте слід пам'ятати, що літак один із найскладнішим транспортів, тому стабільність системи повинна буди колосальною. У системах таким літаків не повинні бути втрати або ж спотворення даних, адже від стабільності та правильного розуміння поставленої машині задачі залежить не тільки життя пасажирів на борту, а й потенційних жертв крушіння літака. Тому необхідно забезпечити ефективну систему управління, котра буде захищена від проявів шумів та перешкод.

У цьому етапі кодування та управління інформаційними процесами є абсолютними. У наш час безпілотники найчастіше використовуються у військових цілях. Системи зв'язку таких безпілотників потребують великої кількості ресурсів. Зараз завдання полягає у створенні цифрових систем зв'язку для безпілотників для передачі високошвидкісної інформації на великі відстані.

Основними проблемами при створенні подібної системи дальнього зв'язку є:

- Забезпечення радіовидимості між літаком та системою зв'язку
- Компенсація загасання сигналу

Зона прямої видимості між наземною системою управління досягається за рахунок збільшення висоти польоту літального апарата та збільшення висоти наземної антени. На високій швидкості та на відстані понад 200 кілометрів можлива із застосування релейного обладнання або систем супутникового зв'язку.

Вже існуюча технологія передачі зображення ефективно не використовує ресурси радіоканалів. Ось чому стає актуальним вирішення таких проблем, як:

- Реалізація гарантованої функції передачі
- Створення системи передачі інформації, яка може ефективно використовувати енергію та спектральний ресурс.
- Реалізація можливості отримання останнього знімка в повному дозволі з метою уточнення.

Однак у військовій промисловості є друга велика галузь, пов'язана з безпілотниками - розробка систем, метою яких є відключення ворожих безпілотників. У багатьох арміях є війська радіоелектронної боротьби, які за допомогою передового обладнання готові перехопити управління ворожим безпілотником, Метою розробки таких інструментів є створення умов, в яких безпілотники стануть вразливими до злону, як сучасні смартфони. Для досягнення цієї мети необхідні останні розробки в теорії інформації та кодування. Високі вимоги до якості передачі інформації характеризуються системами цивільної авіації, які все ще продовжують швидко розвиватися. У наш час великі кошти вкладаються у розвиток виробництва та інтенсивну експлуатацію авіаційної техніки, будівництво та реконструкцію великих міжнародних та основних аеропортів, міжконтинентальних та принавігаційних систем, аеровокзальних комплексів, високопродуктивних систем обробки вантажів, автоматизованих систем управління з використанням обчислювальне обладнання. Незважаючи на

значне зменшення зростання вимог до збільшення швидкості, інтенсивності, висоти, дальності польоту цивільних літаків, вимоги до безпеки та безпеки бортових систем серйозно зросли[8] .

Спостерігаються такі основні тенденції розвитку інформаційних технологій цивільної авіації: розширення сфер застосування та виконуваних функцій, розвиток нових асортиментів, посилення взаємозв'язків між певними типами інформаційних систем, збільшення співвідношення пристроїв обробки інформації на борту, збільшення складності систем разом із зростаючими вимогами до якості їх функціонування, системний підхід до аналізу, синтезу та оптимізації об'єктів інформаційних технологій на основі імовірнісних та статистичних методів . Якщо безпілотні пасажирські літаки можуть бути майбутнім, на відміну від військових безпілотників, все ще існує велика потреба у розробці найточніших систем поточної обробки літаками людини (пілота) для запобігання можливим проблемам. Для розробки таких систем очевидно необхідно залучати експертів з інформаційних процесів. Вивчення інформаційних процесів та розробка нових систем на основі цього стало важливим компонентом, що забезпечує розвиток галузі та використання все більш досконалої продукції. Саме з цієї причини необхідно забезпечити найсучасніший набір інструментів для студентів з метою кращого вивчення дисципліни.

Висновок до 2 розділу

У главі 2 було описано декілька методів кодування та передачі. Ці методи зазвичай використовуються для вирішення одних і тих же проблем, але в різних ситуаціях. Розгляд різних методів кодування та передачі показує нам, як створені коди мають власний баланс між своїми характеристиками. Деякі з них легше передаються; інші краще декодувати тощо. Реалізація концепції надмірності дозволила створити «розумні коди». Ці коди відрізняються від описаних примітивних.

Механізм виявлення та виправлення помилок дозволяє зменшити кількість необхідного обладнання або вимог до якості обладнання, наприклад. Коди з виявленням та виправленням помилок мають більше різноманіття; іноді їх принципи є суперечливими.

Блок-схема Програми



Введення даних

У моєму сегменті коду, введення даних виконується вручну з клавіатури за допомогою команди:

```
CodedBinary = EnCoderVV(Text2Code)
```

Де Text2Code – набір вхідних параметрів, для прикладу введемо команду 'MOVE'.

На виході ми отримаємо набір значень двійкової системи кодування повідомлення.

Кодування даних

Метод кодування інформації в даній дипломній роботі табличний, це означає, що кожна літера має свій запрограмований код, а в разі, якщо станеться спотворення, та буде отримано код, не записаний у таблицю – такі елементи будуть записані символом '*'.

```
for i = 1:length(Text2Code)
    iChar = Text2Code(i);
    switch iChar
        case 'A'
            CodedBinary = [CodedBinary, '00000'];
        case 'B'
            CodedBinary = [CodedBinary, '00001'];
        case 'C'
            CodedBinary = [CodedBinary, '00010'];
        case 'D'
            CodedBinary = [CodedBinary, '00011'];
        case 'E'
            CodedBinary = [CodedBinary, '00100'];
        case 'F'
            CodedBinary = [CodedBinary, '00101'];
        case 'G'
            CodedBinary = [CodedBinary, '00110'];
        case 'H'
            CodedBinary = [CodedBinary, '00111'];
        case 'I'
            CodedBinary = [CodedBinary, '01000'];
```

```

case 'J'
    CodedBinary = [CodedBinary, '01001'];
case 'K'
    CodedBinary = [CodedBinary, '01010'];
case 'L'
    CodedBinary = [CodedBinary, '01011'];
case 'M'
    CodedBinary = [CodedBinary, '01100'];
case 'N'
    CodedBinary = [CodedBinary, '01101'];
case 'O'
    CodedBinary = [CodedBinary, '01110'];
case 'P'
    CodedBinary = [CodedBinary, '01111'];
case 'Q'
    CodedBinary = [CodedBinary, '10000'];
case 'R'
    CodedBinary = [CodedBinary, '10001'];
case 'S'
    CodedBinary = [CodedBinary, '10010'];
case 'T'
    CodedBinary = [CodedBinary, '10011'];
case 'U'
    CodedBinary = [CodedBinary, '10100'];
case 'V'
    CodedBinary = [CodedBinary, '10101'];
case 'W'
    CodedBinary = [CodedBinary, '10110'];
case 'X'
    CodedBinary = [CodedBinary, '10111'];
case 'Y'
    CodedBinary = [CodedBinary, '11000'];
case 'Z'
    CodedBinary = [CodedBinary, '11001'];
case ' '
    CodedBinary = [CodedBinary, '11011'];
otherwise
    CodedBinary = [CodedBinary, '*****'];
    disp(['Символ i=', ' помилковий!'])
end

```


Потроєння інформації

Наступним кроком виправки нашої інформації стане реалізація так званого методу «Розумного кодування».

```
for i = 1:L
    Class = strcat(Text2Code(i), Text2Code(i), Text2Code(i));
    TripledText2Code = strcat(TriplesText2Code, Class);
end
```

Таким чином, ми штучно змінюємо обсяг повідомлення, дублюючи кожен його елемент, через це ймовірність того, що спотворення відбудеться одразу у трьох відповідних розрядах суттєво зменшується.

Далі наша інформація відправляється нашому космічному роботу. Інформація, проходячи величезну дистанцію через космічний простір, піддається впливу шумів (випромінювання Сонця та зірок у відповідних діапазонах частот). Випадкові елементи нашого повідомлення спотворюються.

Генератор випадкових помилок (шумів)

Оскільки ми не маємо змоги використати наш код на реальній практиці, відправити нашу команду через космічний простір, то було вирішено написати код, який з випадковим шансом змінює кожен сегмент повідомлення на протилежний.

```
Occ=rand();
if Occ <= p
    r=1;
else
    r=0;
end
```

```

for i=1:L
    Err=RandVV(p);
    if Err==1
        if ErrBinaryCode(i)=='1'
            ErrBinaryCode(i)=='0';
        else
            ErrBinaryCode(i)=='1';
        end
    end
end
end

```

Декодуючий пристрій

Наше спотворене повідомлення потрапляє у декодер. Декодуючий пристрій намагається ідентифікувати помилку та її місцезнаходження та усунути її методом знаходження моди в кожному потроєному сегменті.

```

or i = 1:5:L
    iPart = code2Text(i:i+4);
    switch iPart
        case '00000'
            deCodedText = [deCodedText, 'A'];
        case '00001'
            deCodedText = [deCodedText, 'B'];
        case '00010'
            deCodedText = [deCodedText, 'C'];
        case '00011'
            deCodedText = [deCodedText, 'D'];
        case '00100'
            deCodedText = [deCodedText, 'E'];
        case '00101'
            deCodedText = [deCodedText, 'F'];
        case '00110'
            deCodedText = [deCodedText, 'G'];
        case '00111'
            deCodedText = [deCodedText, 'H'];
        case '01000'
            deCodedText = [deCodedText, 'I'];
        case '01001'

```

```

        deCodedText = [deCodedText, 'J'];
    case '01010'
        deCodedText = [deCodedText, 'K'];
    case '01011'
        deCodedText = [deCodedText, 'L'];
    case '01100'
        deCodedText = [deCodedText, 'M'];
    case '01101'
        deCodedText = [deCodedText, 'N'];
    case '01110'
        deCodedText = [deCodedText, 'O'];
    case '01111'
        deCodedText = [deCodedText, 'P'];
    case '10000'
        deCodedText = [deCodedText, 'Q'];
    case '10001'
        deCodedText = [deCodedText, 'R'];
    case '10010'
        deCodedText = [deCodedText, 'S'];
    case '10011'
        deCodedText = [deCodedText, 'T'];
    case '10100'
        deCodedText = [deCodedText, 'U'];
    case '10101'
        deCodedText = [deCodedText, 'V'];
    case '10110'
        deCodedText = [deCodedText, 'W'];
    case '10111'
        deCodedText = [deCodedText, 'X'];
    case '11000'
        deCodedText = [deCodedText, 'Y'];
    case '11001'
        deCodedText = [deCodedText, 'Z'];
    case '11011'
        deCodedText = [deCodedText, ' '];
    otherwise
        deCodedText = [deCodedText, '*'];
        disp(['Code Part i=', num2str(i), iPart, '
ПОМИЛКОВИЙ'])
    end
end
end

```

Наприклад:

Було відправлено код англійської літери «А» → '00000'

На декодер прийшов код '000001100000010'

Наш декодер розділяє цей код на 3 частини, оскільки наша інформація потроюється і по суті 15 розрядів це одна літера і порівнює кожен відповідний елемент.

00000

11000

00010

00000

Оскільки значення моди кожного стовпчика дорівнює 0, наше повідомлення приймає наш оригінальний вид. Отже, наш код виправлено. Але такий метод не є стопроцентний, адже навіть при дуже малій ймовірності спотворення ніхто та ніщо не обіцяє нам, що спотворення 2 з 3 відповідних елементів не може статися двічі. В такому разі даний метод кодування спрацює не ефективно.

Кодуючий пристрій

```
function CodedBinary = EnCoderVVBACKUP(Text2Code)
```

Справка до цієї програми:

Ця програма виконує кодування повідомлення "Text2code" з 26 літер англійського алфавіту та клавіші "Space" за таблицею.

Приклад використання:

```
>> CodedBinary = EnCoderVVBACKUP('ABCD EAA DD')
```

In result you will get: CodedBinary =

```
'000001010011101100000000101011011'
```

```
BodoCodes =
```

```
{'00000';'00001';'00010';'00011';'00100';'00101';'00110';'00111';'01000';'01001';'01010';'01011';'01100';'01101';'01110';'01111';'10000';'10001';'10010';'10011';'10100';'10101';'10110';'10111';'11000';'11001'; '11011'};
```

```
TripledText2Code = '';
```

```
CodedBinary = [];
```

```
L = length(Text2Code);
```

```
for i = 1:L
```

```
    Class = strcat(Text2Code(i),Text2Code(i),Text2Code(i));
```

```
    TripledText2Code = strcat(TripledText2Code, Class);
```

```
end
```

```
for i = 1:length(TripledText2Code)
```

```
    iChar = TripledText2Code(i);
```

```
    switch iChar
```

```
        case 'A'
```

```
            CodedBinary = [CodedBinary, '00000'];
```

```
        case 'B'
```

```
            CodedBinary = [CodedBinary, '00001'];
```

```
        case 'C'
```

```
            CodedBinary = [CodedBinary, '00010'];
```

```
        case 'D'
```

```
            CodedBinary = [CodedBinary, '00011'];
```

```
        case 'E'
```

```
            CodedBinary = [CodedBinary, '00100'];
```

```
        case 'F'
```

```
            CodedBinary = [CodedBinary, '00101'];
```

```
        case 'G'
```

```
            CodedBinary = [CodedBinary, '00110'];
```

```
        case 'H'
```

```
            CodedBinary = [CodedBinary, '00111'];
```

```
        case 'I'
```

```

        CodedBinary = [CodedBinary, '01000'];
    case 'J'
        CodedBinary = [CodedBinary, '01001'];
    case 'K'
        CodedBinary = [CodedBinary, '01010'];
    case 'L'
        CodedBinary = [CodedBinary, '01011'];
    case 'M'
        CodedBinary = [CodedBinary, '01100'];
    case 'N'
        CodedBinary = [CodedBinary, '01101'];
    case 'O'
        CodedBinary = [CodedBinary, '01110'];
    case 'P'
        CodedBinary = [CodedBinary, '01111'];
    case 'Q'
        CodedBinary = [CodedBinary, '10000'];
    case 'R'
        CodedBinary = [CodedBinary, '10001'];
    case 'S'
        CodedBinary = [CodedBinary, '10010'];
    case 'T'
        CodedBinary = [CodedBinary, '10011'];
    case 'U'
        CodedBinary = [CodedBinary, '10100'];
    case 'V'
        CodedBinary = [CodedBinary, '10101'];
    case 'W'
        CodedBinary = [CodedBinary, '10110'];
    case 'X'
        CodedBinary = [CodedBinary, '10111'];
    case 'Y'
        CodedBinary = [CodedBinary, '11000'];
    case 'Z'
        CodedBinary = [CodedBinary, '11001'];
    case ' '
        CodedBinary = [CodedBinary, '11011'];
    otherwise
        CodedBinary = [CodedBinary, '*****'];
        disp(['Символ i=', ' помилковий!'])
end
end

```

Декодуєчий пристрій

```
function deCodedText = DeCoderVV(code2Text)
%% Справка для цієї програми:
% Ця програма виконує декодування бінарного коду в літери
англійського
% алфавіту за таблицею
%% Приклад використання:
% >> deCodedText =
DeCoderVV('000001010011101100000000101011011')
% In result you will get: deCodedText = 'ABCD EAA DD'
% Copyright IvanVoronkov, Spring 2021.
ivanvoronkov8@gmail.com
%%
BodoCodes =
{'00000';'00001';'00010';'00011';'00100';'00101';'00110';'0
0111';'01000';'01001';'01010';'01011';'01100';'01101';'0111
0';'01111';'10000';'10001';'10010';'10011';'10100';'10101';
'10110';'10111';'11000';'11001'; '11011'};
deCodedText = [];
L = length(code2Text);

for i = 1:5:L
    iPart = code2Text(i:i+4);
    switch iPart
        case '00000'
            deCodedText = [deCodedText, 'A'];
        case '00001'
            deCodedText = [deCodedText, 'B'];
        case '00010'
            deCodedText = [deCodedText, 'C'];
        case '00011'
            deCodedText = [deCodedText, 'D'];
        case '00100'
            deCodedText = [deCodedText, 'E'];
        case '00101'
            deCodedText = [deCodedText, 'F'];
        case '00110'
            deCodedText = [deCodedText, 'G'];
        case '00111'
            deCodedText = [deCodedText, 'H'];
        case '01000'
            deCodedText = [deCodedText, 'I'];
        case '01001'
```

```

        deCodedText = [deCodedText, 'J'];
    case '01010'
        deCodedText = [deCodedText, 'K'];
    case '01011'
        deCodedText = [deCodedText, 'L'];
    case '01100'
        deCodedText = [deCodedText, 'M'];
    case '01101'
        deCodedText = [deCodedText, 'N'];
    case '01110'
        deCodedText = [deCodedText, 'O'];
    case '01111'
        deCodedText = [deCodedText, 'P'];
    case '10000'
        deCodedText = [deCodedText, 'Q'];
    case '10001'
        deCodedText = [deCodedText, 'R'];
    case '10010'
        deCodedText = [deCodedText, 'S'];
    case '10011'
        deCodedText = [deCodedText, 'T'];
    case '10100'
        deCodedText = [deCodedText, 'U'];
    case '10101'
        deCodedText = [deCodedText, 'V'];
    case '10110'
        deCodedText = [deCodedText, 'W'];
    case '10111'
        deCodedText = [deCodedText, 'X'];
    case '11000'
        deCodedText = [deCodedText, 'Y'];
    case '11001'
        deCodedText = [deCodedText, 'Z'];
    case '11011'
        deCodedText = [deCodedText, ' '];
    otherwise
        deCodedText = [deCodedText, '*'];
        disp(['Code Part i=', num2str(i), iPart, '
ПОМИЛКОВИЙ'])
    end
end
end

```


Генератор помилок

```
function r=RandVV(p)
%% Справка для цієї програми:
% Ця програма задає значення 1 заданою вірогідністю.
%% Приклад використання:
% >> r=RandVV(0.44)
% In result you will get 1 with probability of 44%.
% Copyright IvanVoronkov, Spring 2021.
ivanvoronkov8@gmail.com
%%
Occ=rand();
if Occ <= p
    r=1;
else
    r=0;
end

function ErrBinaryCode=ErrorVV(BinaryCode, p)
%% Справка для цієї програми:
%Ця програма імітує помилку в отриманому повідомленні.
Значення змінюється
%на протилежне з випадковою вірогідністю, задаваної
програмою RandVV
% Copyright IvanVoronkov, Spring 2021.
ivanvoronkov8@gmail.com
%%
L=length(BinaryCode);
ErrBinaryCode=BinaryCode;
for i=1:L
    Err=RandVV(p);
    if Err==1
        if ErrBinaryCode(i)=='1'
            ErrBinaryCode(i)=='0';
        else
            ErrBinaryCode(i)=='1';
        end
    end
end
end
```

Висновки

Завадостійкість системи грає дуже важливу роль при автоматизації технологічних процесів. Так як в наш час, повна автоматизація певних сфер людського життя вже нікого не здивує, то з цього виходить, що ми в певному сенсі стаємо «залежними» від роботів. В дипломній роботі вже було наведено як приклад, можливий шлях розвитку авіації, а саме безпілотні пасажирські перевезення. То в такому випадку життя пасажирів літака буде повністю належати роботам.

Це означає, що дуже важливо на будь-якому етапі розвитку автоматизації та передачі інформації забезпечити надійно, ефективно, та стійку систему з можливістю виправлення помилок.

Джерела інформації:

1. <https://rb.ru/story/robotics-dictionary/> - коротка історія роботизації
2. <https://www.unn.com.ua/ru/news/1930580-v-nasa-rozprovili-podrobitsi-shostogo-polotu-gelikoptera-ingenuity-na-marsi> – повідомлення про космічну роботизовану місію НАСА на Марс
3. https://stud.com.ua/107373/zhurnalistika/matematiczna_model_komunikatsiyi_shennona_uivera – повідомлення про основи передачі інформації.
4. Цымбал В.П. Теория информации и кодирование : Учебник.—4-е изд., перераб. и доп.—К :Вища шк., 1992. – 263 с.: ил. ISBN 5-11-001943-6
5. Чикрин Д.Е. Теория информации и кодирования: курс лекций / Д.Е. Чикрин. -Казань: Казанский университет, 2013. – 116 с
6. Игнатов В.А. Теория информации и передачи сигналов: Учебник для вузов.—М.: Сов.радио, 1979. – 280 с., ил.
7. Гаєв Є.О. Практикум Основи теорії інформаційних процесів: MATLAB та Java
8. Дипломна робота студента НАУ Бойко А. «Цифрова лабораторія з теорії інформаційних процесів. Java – реалізація»
9. Дипломна робота студента НАУ Бабориґа. А
10. <https://studfile.net/preview/3619115/page:26/> - схема передачі інформації
11. <https://step.org.ua/konspekt/info/tema3>
12. https://studme.org/194638/tehnika/pomehi_shumy_kanalakh_peredachi_informatsii