

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ STUDIJŲ PROGRAMA

Golay kodo veikimo modeliavimas

Laboratorinis darbas

Atliko: 3 kurso IV grupės studentas

Arminas Bražėnas

Vilnius – 2024

Turinys

ĮVADAS	3
1. REALIZUOTOS UŽDUOTIES DALYS	3
2. TREČIŲJŲ ŠALIŲ FUNKCIJŲ BIBLIOTEKOS	3
3. UŽDUOTIES ATLIKIMO LAIKAS	3
4. PROGRAMOS PALEIDIMAS	4
5. PROGRAMOS TEKSTŲ FAILAI	4
6. VARTOTOJO SĄSAJA	5
7. PROGRAMINIAI SPRENDIMAI	7
8. EKSPERIMENTAI	8
ŠALTINIAI	9

Įvadas

Šiame dokumente pateikiamas Golay kodo veikimo modeliavimo laboratorinio darbo aprašymas. Informacijos kodavimas ir dekodavimas modeliuojamas pasitelkiant literatūroje [Hil91] aprašytus algoritmus.

1. Realizuotos užduoties dalys

Laboratoriniame darbe realizuotos visos užduoties dalys:

1. Vektoriaus kodavimas, siuntimas kanalu ir dekodavimas. Vartotojas turi galimybę prieš dekodavimą redaguoti iš kanalo išėjusį vektorius, nurodydamas klaidas ten, kur reikia, ir tiek, kiek reikia.
2. Vartotojo užrašyto teksto siuntimas kanalu:
 - nenaudojant klaidas taisančio kodo;
 - naudojant Golay kodą teksto kodavimui ir dekodavimui.
3. Vartotojo nurodyto BMP formato paveiksliuko siuntimas kanalu:
 - nenaudojant klaidas taisančio kodo;
 - naudojant Golay kodą paveiksliuko kodavimui ir dekodavimui.

2. Trečiųjų šalių funkcijų bibliotekos

Laboratorinio darbo programinis kodas parašytas C# programavimo kalba, pasitelkiant .NET karkasą. Darbe naudojamos trečiųjų šalių funkcijų bibliotekos:

- Avalonia – vartotojo sąsajos kūrimo biblioteka; pasirinkta dėl suderinamumo tarp kelių operacinių sistemų [Ava].
- xUnit – vienetų testų rašymo įrankis [xUn].

3. Užduoties atlikimo laikas

Atlikti užduotį užtruko 45 valandos:

- Literatūros skaitymas ir kodo veikimo aiškinimasis – 5 valandos.
- Projektavimas – 2 valandos.
- Programavimas – 25 valandos.
- Klaidų ieškojimas ir taisymas – 6 valandos.
- Ataskaitos ruošimas – 7 valandos.

4. Programos paleidimas

Paleisti programą galima atidarius *publish* aplanke esantį failą *GolayCodeSimulator.exe*.

5. Programos tekstų failai

Programos tekstų failai yra atskirti į 3 modulius:

1. *Core* – Golay kodo veikimo modeliavimo pagrindas. Sudarytas iš:
 - *BinarySymmetricChannel.cs* – realizuotas duomenų srauto siuntimas dvejetainiu simetriniu kanalu.
 - *Constants.cs* – Golay kodo konstantos.
 - *GolayDecoder.cs* – realizuotas duomenų srauto dekodavimas.
 - *GolayEncoder.cs* – realizuotas duomenų srauto kodavimas.
 - *GolayInformationParser.cs* – realizuotas informacijos išgavimas iš dekodauto duomenų srauto.
 - *Helpers/BitReader.cs* – realizuotas bitų skaitymas blokais iš baitų sekos.
 - *Helpers/BitWriter.cs* – realizuotas bitų rašymas blokais į baitų seką.
 - *Helpers/ExtensionMethods.cs* – pagalbiniai išplėsties metodai.
 - *Helpers/Matrix.cs* – realizuoti matematiniai veiksmai su matricomis.
 - *Helpers/Utilities.cs* – pagalbiniai metodai.
2. *Core.Tests* – vienetų testai bei eksperimentai Golay kodo modeliavimui.
3. *Presentation* – vartotojo sąsajos tekstų failai. Esminiai failai:
 - *Program.cs* – programos įeities klasė.
 - *Helpers/BinaryStringConverter.cs* – realizuotas dvejetainės tekstinės eilutės vertimas į baitų seką bei baitų sekos vertimas atgal į tekstinę eilutę.
 - *Helpers/ExtensionMethods.cs* – pagalbiniai išplėsties metodai.
 - *Helpers/SimulationUtility.cs* – realizuotas duomenų srauto kodavimas, siuntimas kanalu ir dekodavimas.
 - *Validation/** – šiuose failuose realizuotos vartotojo įvedamos informacijos validavimas.
 - *Views/** – šiuose failuose realizuoti vartotojo sąsajos elementai bei jų išdėstymas.
 - *ViewModels/MainViewModel.cs* – realizuota vartotojo lango pakeitimo logika.
 - *ViewModels/MessageSimulationViewModel.cs* – realizuotas vektoriaus kodavimas, siuntimas kanalu ir dekodavimas.
 - *ViewModels/TextSimulationViewModel.cs* – realizuotas teksto kodavimas, siuntimas kanalu ir dekodavimas.
 - *ViewModels/ImageSimulationViewModel.cs* – realizuotas paveiksluko kodavimas, siuntimas kanalu ir dekodavimas.

6. Vartotojo sąsaja

Vartotojo sąsają sudaro trys langai. Vartotojas gali pakeisti langą atitinkamai pasirinkdamas iš lango viršuje esančio meniu.

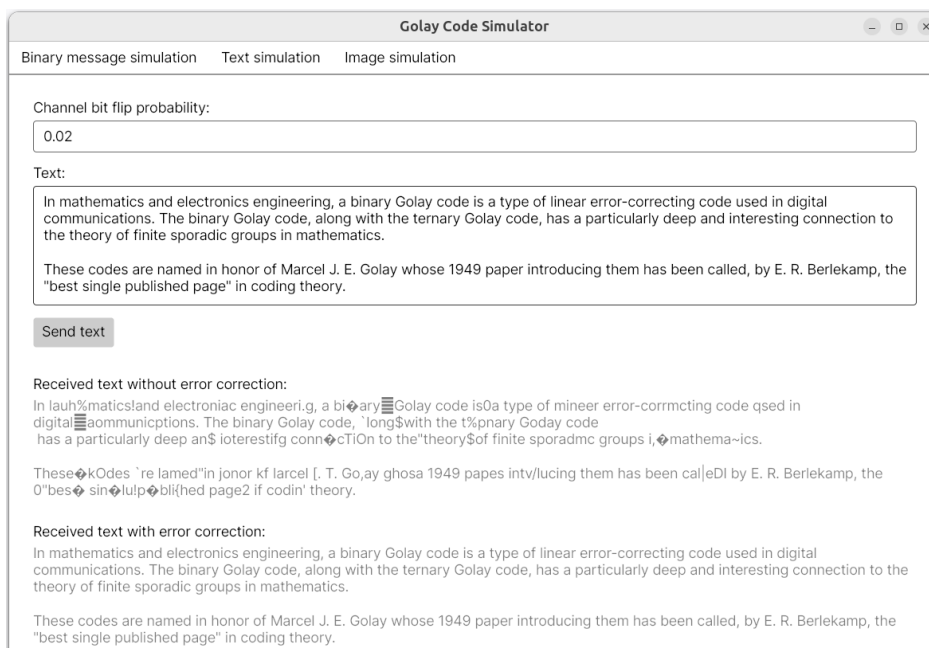
1. Šiame lange modeliuojamas vektoriaus kodavimas, siuntimas kanalu ir dekodavimas. Vartotojas turi įvesti kanalo klaidos tikimybę bei pranešimą, kurį nori siųsti kanalu. Pranešimas turi būti sudarytas iš vieno ar daugiau dvejetainių vektorių, kurių kiekvieno ilgis yra 12. Paspaudus mygtuką *Send message*, sistema pranešimą koduoja ir siunčia kanalu. Pranešimui išėjus iš kanalo, sistema vartotojui praneša kiek ir kuriose pozicijose įvyko klaidų. Vartotojas prieš dekodavimą iš kanalo išėjusį pranešimą gali jį redaguoti. Redaguotas pranešimas privalo būti sudarytas iš vieno ar daugiau dvejetainių vektorių, kurių kiekvieno ilgis yra 23. Paspaudus mygtuką *Decode message*, sistema iš kanalo gautą pranešimą dekoduoja ir parodo vartotojui.

The screenshot shows the 'Golay Code Simulator' window with the 'Binary message simulation' tab selected. The interface includes the following elements:

- Channel bit flip probability:** A text input field containing '0.1'.
- Message:** A text input field containing '011110111011'.
- Send message:** A button to initiate the simulation.
- Encoded message:** A text area displaying '01111011101100011111110'.
- Message from channel:** A text input field containing '011010111101000010111110'.
- Error message:** A blue text line stating '3 errors occurred while sending through channel at positions 4, 12, 17.'.
- Decode message:** A button to decode the received message.
- Decoded message:** A text area displaying '01111011101100011111110'.
- Decoded message information:** A text area displaying '011110111011'.

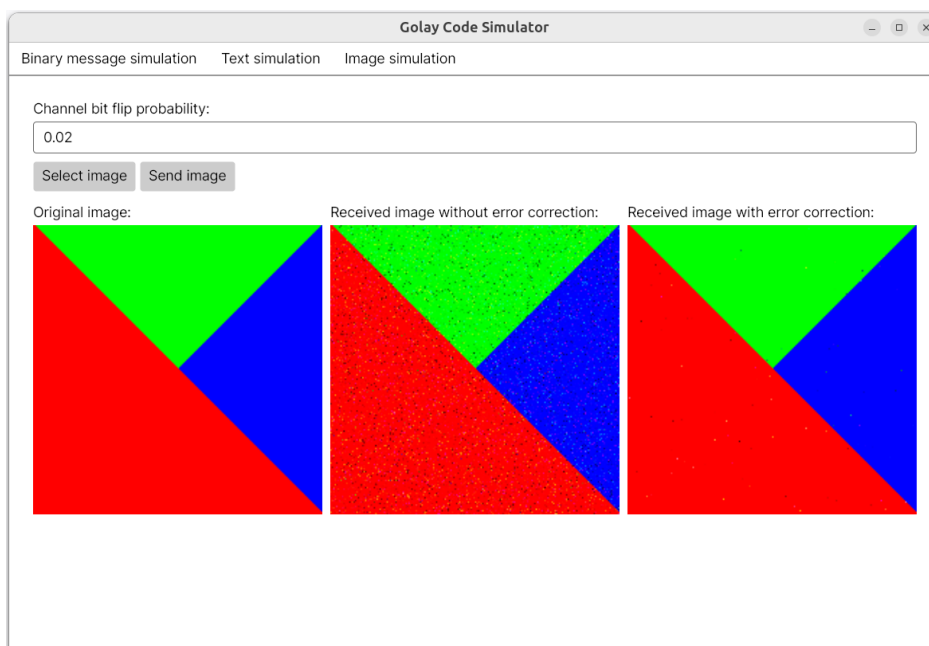
1 pav. Vektoriaus siuntimo kanalu modeliavimo langas.

2. Šiame lange modeliuojamas teksto kodavimas, siuntimas kanalu ir dekodavimas. Vartotojas turi įvesti kanalo klaidos tikimybę bei tekstą, kurį nori siųsti kanalu. Vartotojui paspaudus mygtuką *Send text*, sistema vartotojui parodo iš kanalo išėjusį tekstą nenaudojant klaidas taisančio kodo bei naudojant Golay kodą teksto kodavimui ir dekodavimui.



2 pav. Teksto siuntimo kanalu modeliavimo langas.

3. Šiame lange modeliuojamas paveiksluko kodavimas, siuntimas kanalu ir dekodavimas. Vartotojas turi įvesti kanalo klaidos tikimybę bei pasirinkti BMP formato paveiksluką, kurį nori siųsti kanalu. Vartotojui paspaudus mygtuką *Send image*, sistema vartotojui parodo iš kanalo išėjusį paveiksluką nenaudojant klaidas taisančio kodo bei naudojant Golay kodą paveiksluko kodavimui ir dekodavimui.



3 pav. Paveiksluko siuntimo kanalu modeliavimo langas.

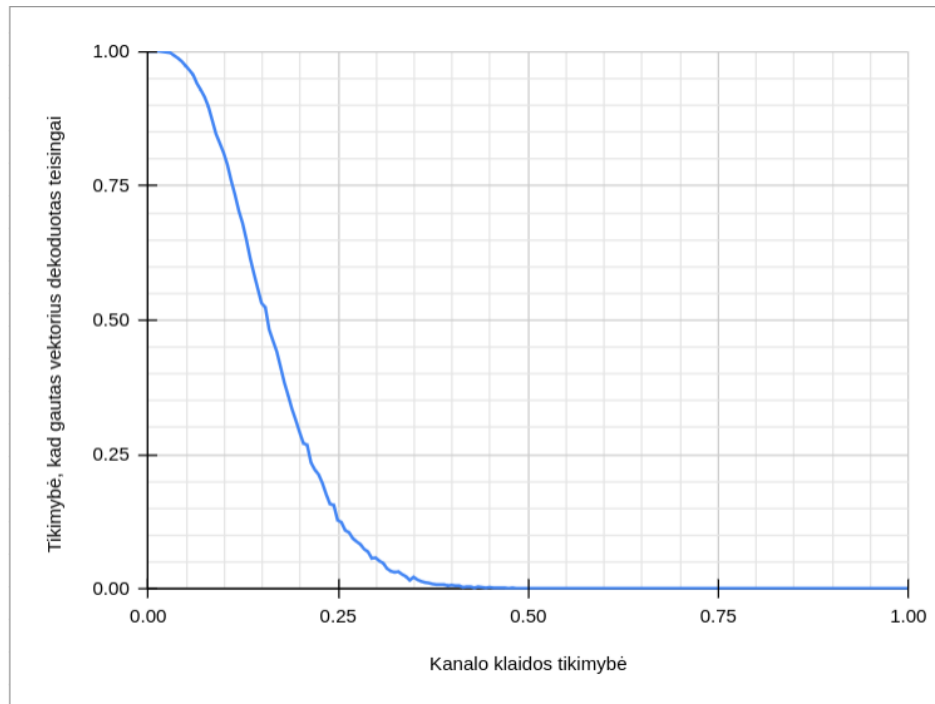
7. Programiniai sprendimai

Atliekant laboratorinį darbą buvo padaryti šie programiniai sprendimai:

- Duomenų srauto siuntimo kanalu simuliacijai naudojamas dvejetainis simetrinis kanalas [Wika].
- Dvejetainiai vektoriai yra dauginami iš transponuotos matricos. Tai leidžia efektyviai įgyvendinti daugybos operaciją: naudoti bitinę operaciją *AND*, apskaičiuoti gauto vektoriaus svorį ir imti rezultato *mod 2*.
- Vektoriaus svoriui apskaičiuoti naudojamas *Brian Kernighan* išrastas algoritmas [Edu].
- Teksto bei paveiksliuko siuntimo kanalu modeliavimo metu, duomenų srautas gali būti papildytas nuliais iki reikiamo ilgio, norint jį suskaidyti į 12 bitų vektorius. Tai laikoma tarnybine informacija, t.y. ji nėra iškraipoma. Kitu atveju, dekodavus gali tapti neaišku ar duomenų srautas buvo papildytas nuliais.
- Paveiksliuko antraštė taip pat yra laikoma tarnybine informacija. Jeigu antraštė būtų iškraipyta, sistema negalėtų vartotojui parodyti iš kanalo gauto paveiksliuko [Wikb].
- Kanalu siunčiamas tekstas yra paverčiamas duomenų srautu naudojant UTF-8 koduotę.

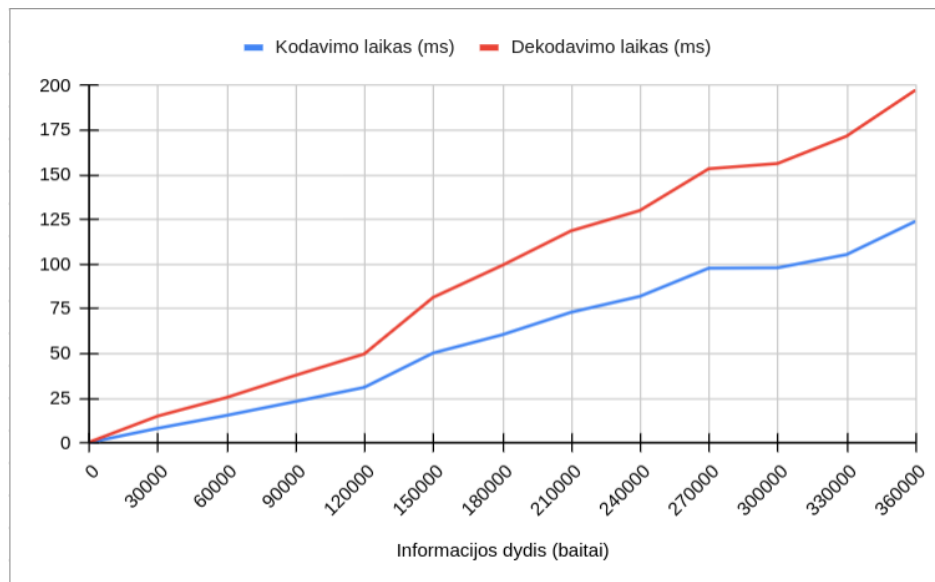
8. Eksperimentai

1. Eksperimente nagrinėjama, kaip tikimybė teisingai dekoduoti iš kanalo gautą vektorių priklauso nuo kanalo klaidos tikimybės. Eksperimento metu kanalo klaidos tikimybė buvo palaipsniui didinama nuo 0 iki 1 su žingsniu 0.005. Su kiekviena klaidos tikimybe vektorius 001111101110 buvo siunčiamas kanalu 5000 kartų. Po to, apskaičiuotas šių bandymų teisingo vektoriaus dekodavimo tikimybės vidurkis.



4 pav. Tikimybės teisingai dekoduoti iš kanalo gautą vektorių priklausomybė nuo kanalo klaidos tikimybės.

2. Eksperimente nagrinėjama, kaip kodavimo ir dekodavimo laikas priklauso nuo siunčiamos informacijos dydžio. Eksperimento metu kanalu su klaidos tikimybe 0.1 buvo siunčiamas atsitiktinis duomenų srautas. Duomenų srauto dydis augo nuo 30000 baitų iki 360000 baitų, kiekvieną iteraciją siunčiant 30000 baitų daugiau negu praeitąją. Kiekvienas pranešimas buvo 10 kartų koduojamas, siunčiamas kanalu ir dekuodojamas. Po to, apskaičiuotas šių bandymų laiko vidurkis.



5 pav. Kodavimo ir dekodavimo laiko priklausomybė nuo siunčiamos informacijos dydžio.

Šaltiniai

- [Ava] Avalonia. *Cross-platform UI framework for .NET*. Prieiga per internetą: <https://avaloniaui.net>.
- [Edu] Educative. *What is Kernighan's algorithm?* Prieiga per internetą: <https://www.educative.io/answers/what-is-kernighans-algorithm>.
- [Hil91] R. Hill. *A first course in coding theory*. New York: Oxford University Press, 1991. Prieiga per internetą: <https://klevas.mif.vu.lt/~skersys/doc/ktkt/literatura12.pdf>.
- [Wika] Wikipedia. *Binary symmetric channel*. Prieiga per internetą: https://en.wikipedia.org/wiki/Binary_symmetric_channel.
- [Wikb] Wikipedia. *BMP file format*. Prieiga per internetą: https://en.wikipedia.org/wiki/BMP_file_format.
- [xUn] xUnit. *Unit testing tool for the .NET*. Prieiga per internetą: <https://xunit.net>.