თამარ კვიკვინია

უწყვეტი კოდები ალფაბეტურ-სიმბოლური სიჭარბით. აგება და მახასიათებლების კვლევა

წარდგენილია დოქტორის აკადემიური ხარისხის მოსაპოვებლად

სადოქტორო პროგრამა "ტელეკომუნიკაცია" შიფრი 0402

საქართველოს ტექნიკური უნივერსიტეტი
თბილისი, 0175, საქართველო
დეკემბერი, 2014 წ.

საქართველოს ტექნიკური უნივერსიტეტი

ენერგეტიკისა და ტელეკომუნიკაციის ფაკულტეტი

ჩვენ, ქვემორე ხელისმომწერნი ვადასტურებთ, რომ გავეცანით თამარ კვიკვინიას მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: „უწყვეტი კოდები ალფაბეტურ-სიმბოლური სიჭარბით. აგება და მახასიათებლების კვლევა“ და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის ენერგეტიკისა და ტელეკომუნიკაციის ფაკულტეტის სადისერატაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

თარიღი_____

ხელმძღვანელი: ———————— ნოდარ უღრელიძე
ტ.მ.დ., პროფ.

რეცენზენტი: ———————— გულაბერ ანანიაშვილი
ტ.მ.დ

რეცენზენტი: ———————— თამაზ კუპატაძე
ტ.მ.დ., პროფ.

საქართველოს ტექნიკური უნივერსიტეტი

ავტორი: თამარ კვიკვინია

დასახელება: უწყვეტი კოდები ალფაბეტურ-სიმბოლური სიჭარბით. აგება და მახასიათებლების კვლევა

ფაკულტეტი: ენერგეტიკა და ტელეკომუნიკაცია

ხარისხი: დოქტორი

სხდომა ჩატარებულია:: ____ _____ 2015 წ.

ინდივიდუალური პიროვნებების ან ინსტიტუტების მიერ, ზემოყვანილი დასახელების დისერტაციის გაცნობის მიზნით მოთხოვნის შემთხვევაში, მისი არაკომერციული მიზნებით კოპირებისა და გავრცელების უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსტეტს.

_____

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა და სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო უფლებებით დაცული მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა ის მცირე ზომის ციტატებისა, რომლებიც მოითხოვენ მხოლოდ სპეციფიურ მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს პასუხისგებლობას.

რეზიუმე

თანამედროვე რადიო და სატელეკომუნიკაციო სისტემებში შეცდომების მაკონტროლებელი კოდების გამოყენება დღესაც რჩება ერთერთ უმთავრეს საშუალებად მათი ეფექტურობის გაზრდისათვის. აქ, კოსმოსურ და თანამგზავრულ სისტემებში, სადაც ენერგეტიკული რესურსები შეზღუდულია, შეცდომების მაკონტროლებელი კოდების გამოყენება ნიშვნელოვანი წინაპირობაა მაღალეფექტური სისტემების ასაგებად. ნახსენები კოდებიდან განსაკუთრებით გამოირჩევიან ხვეულა კოდები, რომელთა ახალი ქვეკლასის შესწავლას ეძღვნება წინამდებარე სადისერტაციო ნაშრომი. კერძოდ, გამოკვლეულია ე.წ. სიმბოლურ-ალფაბეტური სიჭარბის მქონე ხვეულა კოდები, მათ საფუძველზე აგებული სიგნალ-კოდური სისტემები კოსმოსურ და თანამგზავრული არხებისათვის, სადაც არხთა ყველაზე ადეკვატურ მოდელად გაუსის მოდელი ითვლება. უნდა ითქვას, რომ სიმბოლურ-ალფაბეტური სიჭარბის მქონე ხვეულა კოდები დღეისათვის პრაქტიკულად შეუსწავლელია თან, უმეტესწილად სიგნალებთან ერთობლიობაში და გაუსის არხებში.

სადისერტაციო ნაშრომის შესავალ ნაწილში ხაზგასმულია დასმული საკითხის აქტუალობა, ჩამოყალიბებულია გადასაჭრელი ამოცანები, შესაბამისად მათი გადაწყვეტის გზები, წარმოდგენილია ის ძირითადი დებულებები, რომლებიც ავტორის მიერ გამოტანილია დასაცავად.

პირველ თავში განხილულია კლასიკური ხვეულა კოდები, რომლის წარმოდგენისათვის გამოყენებულია სასრული ავტომატის მოდელი, სადაც საინფორმაციო და კოდური მიმდევრობები წარმოდგებიან დაყოვნების ოპერატორებით. ჩვენ, სპეციალურად დაწვრილებით განვიხილეთ დეკოდირების მაქსიმალური დამაჯერებლობის პრინციპი (რომელიც რეალიზებულია ვიტერბის ალგორითმით), რადგანაც იგივე პრინციპით იქნება დეკოდირებული აგებული ახალი კოდები, რომლებიც ეფუძნებიან ხვეულა კოდებს ალფაბეტური სიჭარბით.

მეორე თავში განხილულია კოდები სიმბოლურ-ალფაბეტური სიჭარბით. მოყვანილია მათი აღწერა და დადებითი მხარეები. გადაწყვეტილია, რომ საუკეთესო კოდები უნდა შეირჩეს კომპიუტერული ძიების მეთოდით და ამ დროს თავისუფალი მანძილის განსაზღვრისათვის გამოყენებული იქნას დეიქსტრის ალგორითმი. კონკრეტული კოდისათვის განხილულია ალგორითმის მუშაობის პროცედურა. განსაზღვრულია, რომ ერთნაირი თავისუფალი მანძილის მქონე ხვეულა კოდებიდან საუკეთესოს შერჩევა განხორციელდეს მისი მანძილთა სპექტრის მიხედვით. აღწერილია სპექტრის განსაზღვრის ალგორითმი.

მესამე თავის დასაწყისში ჩამოყალიბებულია მეთოდი, რომელიც საკმარისია მანძილის მიმართ ინვარიანტული კოდებისა და სიგნალ კოდური სისტემების ასაგებად. დეიქსტრის ალგორითმის გამოყენებით

დამუშავებული პროგრამის საშუალებით განხორციელებულია კომპიუტერული ძებნა ახალი, ალფაბეტური სიჭარბის მქონე მანძილის მიმართ ინვარიანტული, კოდების და ნაპოვნი კოდები ტაბულირებულია. ნაჩვენებია, რომ მოყვანილი ოთხობითი კოდები შეიძლება გამოყენებული იქნან როგორც ორობით სიმეტრიული არხებისთვის, ასევე გაუსის არხებისათვის ორობითი და ოთხობითი ფაზამოდულირებული სიგნალებით. აგებულია სამობითი სიგნალ-კოდური სისტემები სიმპლექსური სიგნალების გამოყენებით.

ნაშრომი შეიცავს ორ დანართს, სადაც წარმოდგენილია პროგრამული პაკეტები, რომელთა საშუალებით განხორციელებულია საუკეთესო, ალფაბეტური სიჭარბის მქონე კოდების ძებნა, პოვნა და მათი მახასიათებლების ანგარიში.

სადისერტაციო ნაშრომში მიღებულია შემდეგი მნიშვნელოვანი შედეგები:

1. ნაჩვენებია, რომ ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდები წარმოადგენენ უწყვეტი კოდების უფრო მაღალ საფეხურს, ვიდრე ცნობილი კლასიკური ხვეულა კოდები.

2. მოყვანილია გალუას ველთა იმ არითმეტიკის ნაწილი, რომელიც საჭიროა ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ასაგებად. წარმოდგენილია შესაბამისი ცხრილები.

3. წარმოდგენილია ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ძებნის და მისი პარამეტრების განსაზღვრის ალგორითმები შესაბამისი პროგრამული რეალიზაციებით.

4. წარმოდგენილია მანძილის მიმართ ინვარიანტული ხვეულა კოდების აგების მეთოდი.

5. ნაპოვნია მაღალეფექტური ახალი ხვეულა კოდები და მოყვანილია მათი მახასიათებლები.

Abstract

Usage of error controlling codes remains one of the main part in modern Radio and Telecommunication Systems for raising its efficiency. In space and satellite systems, where energetic resources are strictly limited, using such codes is significant prerequisite for designing high effective telecommunication systems. Especially convolutional codes are distinguished, which subclass is observed in this research. Specifically, codes with symbolic-alphabetic redundancy and signal-code systems depended on it, for space and satellite channels are observed and reviewed, where most appropriate model for channels is Gaussian model. Hereby should be mentioned that codes with symbolic-alphabetic redundancy is actually not studied for nowadays, especially in connection with signals and Gaussian channels.

In the introduction part actuality of the technical issue is eliminated, task to be resolved are described, hence ways to solve it is observed and that main states are presented that author is going to prove.

In first Chapter classic convolutional codes are described, with the finite-state machine model, where information and code sequence is presented with the delay operator. We observed maximum-likelihood decoding method in details (realized with Viterbi Algorithm), because with the same principle the new designed codes will be decoded, that is based to convolutional codes with alphabetic redundancy.

In the second chapter codes with the symbolic-alphabetic redundancy are observed. Its description, with advantages and disadvantages is given. It is decided that best codes should be chosen with computer searching method and for defining free distance Dijkstra's algorithm should be used. For the specific code the procedure how algorithm works is shown. It is defined that from the convolutional codes that have the same free distance, the best choice should be provided by its distance spectrum. Also description of spectrum of algorithm is presented.

In the beginning of the third chapter, the method is described, which is enough for designing distance-invariant codes and signal-code systems. Software is processing the computer searching for new distance-invariant codes with symbolic–alphabetic redundancy, using Dijkstra's algorithm, and that codes are tabulated. Here is shown that given 4-ary codes could be used as for the binary symmetric channels, also for Gaussian Channels with binary and quadrature phase modulated signals. Also triple signal-code systems are designed, that uses simplex signals.

Dissertation Contains two appendices, where software packets are presented, that helps to provide searching of the best codes with alphabetic redundancy, finding and reporting its characteristics.

Research gets the following important results:

1. Alphabetic-symbolic redundancy convolutional codes represents even higher level of continues codes then well-known classic convolutional codes.

2. The arithmetic part of the Galois field is presented, that is needed for designing Alphabetic-symbolic redundancy convolutional codes, and the suitable tables are given.

3. Algorithm of searching the Alphabetic-symbolic redundancy convolutional codes and describing its parameters are presented with the appropriate software realization.

4. Designing method of convolutional codes, invariant toward the distance is presented.

5. High effective new convolutional codes are found and is shown its characteristics.

სარჩევი

ცხრილების ნუსხა

ნახაზების ნუსხა

შესავალი

ნაშრომის აქტუალურობა. თანამედროვე რადიო და სატელეკომუნიკაციო სისტემები წარმოადგენენ გლობალური ქსელის ისეთ შემადგენელ ნაწილს, რომლებიც მნიშვნელოვნად განსაზღვრავენ მის ეფექტურობას. ამ დროს ორი ასპექტია ძირითადი: ენერგეტიკული რესურსი ანუ გადამცემთა სიმძლავრე და ინფორმაციის გადაცემის სიჩქარე. აქ, მინიმაქსის კრიტერიუმის შესრულებას განსაზღვრავს სიგნალებისა და კოდების ოპტიმალური შერჩევა, რასაც ეძღვნება წინამდებარე სადისერტაციო ნაშრომი. კერძოდ განხილულია ახალი კლასის ხვეულა კოდების აგების საკითხი ალფაბეტური სიჭარბის მქონე კოდების ბაზაზე. დღეისათვის ამ მიმართულებით მხოლოდ რამდენიმე ნაშრომია გამოქვეყნებული [25, 26], რაც ცხადია არ არის საკმარისი.

სადისერტაციო ნაშრომის ძირითად მიზანია დამუშავდეს შესაბამისი მეთოდები და აიგოს კონკრეტული ახალი კოდები და გამოთვლილი იქნას მათი მახასიათებლები. ამ მიზნით შესრულდა გარკვეული სამუშაოები, რომლებიც შეიცავენ შემდეგი ***ამოცანების გადაწყვეტას:***

1. ჩატარდა ლიტერატურის ანალიზი მოცემული მიმართულებით არსებული შედეგების გამოვლენის მიზნით.
2. შეირჩა ახალი ხვეულა კოდების დეკოდირების ალგორითმი.
3. შეირჩა ახალი ხვეულა კოდების ძებნის და მისი პარამეტრების ანგარიშის მეთოდები. დამუშავდა შესაბამისი პროგრამული უზრუნველყოფა.
4. დამუშავდა მანძილის მიმართ ინვარიანტული სისტემების აგების მეთოდი.
5. ნაპოვნი იქნა ახალი ხვეულა კოდები და აგებული იქნა ახალი სიგნალ-კოდური სისტემები.

კვლევის მეთოდები. დასმული ამოცანების გადასაწყვეტად გამოყენებული იქნა სასრული ავტომატების თეორია, სასრული ველების

(გალუას) თეორია, კოდირების თეორია, სიგნალების თეორია, ალბათობათა თეორია.

სამეცნიერო სიახლე. ნაშრომში მიღებულია შემდეგი ახალი შედეგები:

ნაჩვენებია, რომ ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდები წარმოადგენენ უწყვეტი კოდების უფრო მაღალ საფეხურს, ვიდრე ცნობილი კლასიკური ხვეულა კოდები.

მოყვანილია გალუას ველთა იმ არითმეტიკის ნაწილი, რომელიც საჭიროა ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ასაგებად. წარმოდგენილია შესაბამისი ცხრილები.

წარმოდგენილია ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ძებნის და მისი პარამეტრების განსაზღვრის ალგორითმები შესაბამისი პროგრამული რეალიზაციებით.

წარმოდგენილია მანძილის მიმართ ინვარიანტული ხვეულა კოდების აგების მეთოდი.

ნაპოვნია მაღალეფექტური ახალი ხვეულა კოდები და მოყვანილია მათი მახასიათებლები.

პრაქტიკული ღირებულება და შედეგების რეალიზაცია. ვინაიდან ახალი კოდები და სიგნალ-კოდური სისტემები არ ხასიათდებიან სიჩქარის მაღალი პარამეტრებით (ნაიკვისტის საზღვარი), მაგრამ აქვთ მაღალი ენერგეტიკული ეფექტურობა (5-6 დბ), ჩვენი რეკომენდაცია იქნება გამოყენებული იქნან ისინი თანამგზავრულ და განსაკუთრებით შორეული კოსმოსური კავშირის სისტემებში.

ნაშრომის აპრობაცია. ნაშრომის ძირითადი შედეგები განხილული და მოხსენებულ იქნა:

1. პირველ და მეორე თემატურ სემინარებზე (თბილისი, სტუ, 2011-2014 წ.)

2. საერთაშორისო სამეცნიერო სიმპოზიუმზე - IEEE 11-th International Sympozium on Electronics and Telecommunications (ISETC ‘14). Timisoara, Romania, November 14-15, 2014.

პუბლიკაციები. დისერტაციის ძირითადი შედეგები ასახულია რეცენზირებად ჟურნალებში 5 ნაშრომის სახით.

დაცვაზე გამოტანილი ***ძირითადი დებულებებია:***

1. სიმბოლურ-ალფაბეტური სიჭარბის მქონე ხვეულა კოდები წარმოადგენენ ხვეულა კოდების უფრო მაღალ კლასს და მოიცავენ ყველა ცნობილ კლასიკურ ხვეულა კოდს.
2. არსებობს პირობა, რომელიც განსაზღვრავს სიმბოლურ-ალფაბეტური სიჭარბის მქონე ხვეულა კოდების ან მათ საფუძველზე აგებული სიგნალ-კოდური სისტემების ინვარიანტობას შესაბამის მანძილთა მიმართ.

სადისერტაციო ნაშრომის მოცულობა და სტრუქტურა. ნაშრომი შედგება შესავალის, სამი თავის, დასკვნის, გამოყენებული ლიტერატურის და ორი დანართისაგან.

პირველ თავში განხილულია კლასიკური ხვეულა კოდები, რომლის წარმოდგენისათვის გამოყენებულია სასრული ავტომატის მოდელი, სადაც საინფორმაციო და კოდური მიმდევრობები წარმოდგებიან დაყოვნების ოპერატორებით.

ჩვენ, სპეციალურად დაწვრილებით განვიხილეთ დეკოდირების მაქსიმალური დამაჯერებლობის პრინციპი (რომელიც რეალიზებულია ვიტერბის ალგორითმით), რადგანაც იგივე პრინციპით იქნება დეკოდირებული აგებული ახალი კოდები, რომლებიც ეფუძნებიან ხვეულა კოდებს ალფაბეტურივ სიჭარბით.

მეორე თავში განხილულია კოდები სიმბოლურ-ალფაბეტური სიჭარბით. მოყვანილია მათი აღწერა და დადებითი მხარეები. გადაწყვეტილია, რომ საუკეთესო კოდები უნდა შეირჩეს კომპიუტერული

ძიების მეთოდით და ამ დროს თავისუფალი მანძილის განსაზღვრისათვის გამოყენებული იქნას დეიქსტრის ალგორითმი. კონკრეტული კოდისათვის განხილულია ალგორითმის მუშაობის პროცედურა. განსაზღვრულია, რომ ერთნაირი თავისუფალი მანძილის მქონე ხვეულა კოდებიდან საუკეთესოს შერჩევა განხორციელდეს მისი მანძილთა სპექტრის მიხედვით. აღწერილია სპექტრის განსაზღვრის ალგორითმი.

მესამე თავში ჩამოყალიბებულია მეთოდი, რომელიც საკმარისია მანძილის მიმართ ინვარიანტული კოდებისა და სიგნალ-კოდური სისტემების ასაგებად. დეიქსტრის ალგორითმის გამოყენებით დამუშავებული პროგრამის საშუალებით განხორციელებულია კომპიუტერული ძებნა ახალი, ალფაბეტური სიჭარბის მქონე მანძილის მიმართ ინვარიანტული, კოდების და ნაპოვნი კოდები ტაბულირებულია. ნაჩვენებია, რომ მოყვანილი ოთხობითი კოდები შეიძლება გამოყენებული იქნან როგორც ორობით სიმეტრიული არხებისთვის, ასევე გაუსის არხებისათვის ორობითი და ოთხობითი ფაზამოდულირებული სიგნალებით. აგებულია სამობითი სიგნალ-კოდური სისტემები სიმპლექსური სიგნალების გამოყენებით.

სადისერტაციო ნაშრომის ***საბოლოო დასკვნაში*** მოყვანილია მიღებული შედეგები და რეკომენდაციები.

ნაშრომი შეიცავს ***ორ დანართს***, სადაც წარმოდგენილია პროგრამული პაკეტები, რომელთა საშუალებით განხორციელებულია საუკეთესო, ალფაბეტური სიჭარბის მქონე კოდების ძებნა, პოვნა და მათი მახასიათებლების ანგარიში.

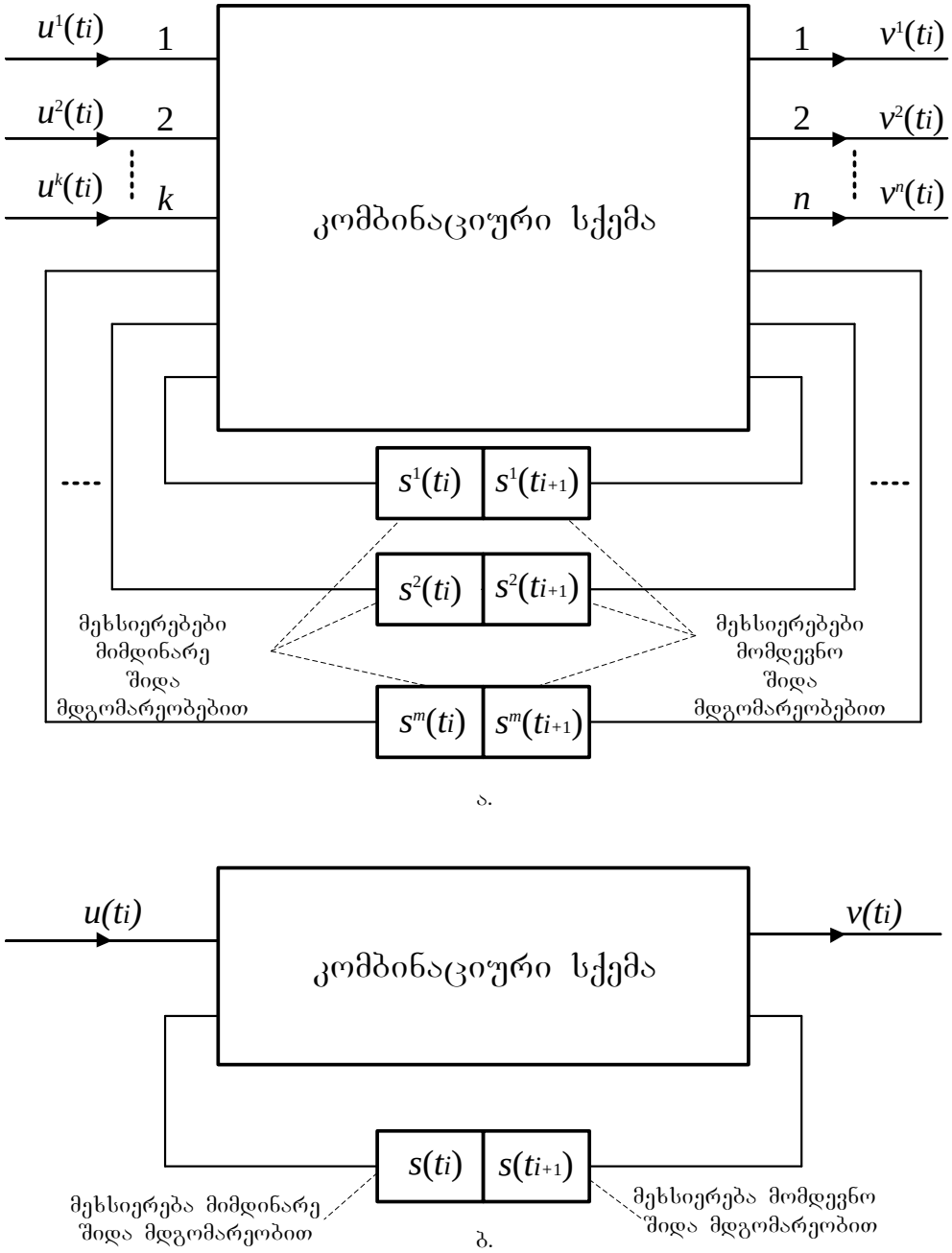
თავი I

კლასიკური ხვეულა კოდები

1.1 სასრული ავტომატები და ხვეულა კოდები

აბსტრაქტული სასრული ავტომატი არის რეალური ობიექტის ან მოვლენის იდეალიზირებული მათემატიკური მოდელი, რომელსაც აქვს: შესასვლელი, სიმბოლოთა მიმდევრობების ერთობლიობით $u(t_i) = u^1(t_i), u^2(t_i), \ldots, u^k(t_i)$; შიდა მეხსიერება სასრული მდგომარეობებით, რომელიც შეიცავს სიმბოლოთა მიმდევრობების ერთობლიობას $s(t_i)=s^1(t_i), s^2(t_i), \ldots, s^m(t_i)$ და გამოსასვლელი, სიმბოლოთა მიმდევრობების ერთობლიობით $v(t_i) = v^1(t_i), v^2(t_i), \ldots, v^n(t_i)$ [1, 2]. დროს, რომლის განმავლობაშიც ფუნქციონირებს ავტომატი არის დისკრეტული (მას ტაქტური ინტერვალები ეწოდება) და ამასთან დაკავშირებით სიმბოლოები ავტომატის შესასვლელზე, შიდა მეხსიერებაში და გამოსასვლელზე წარმოადგენენ დისკრეტულ სიმბოლოებს, კონკრეტულ შემთხვევაში კი შეიძლება იყვნენ ციფრულები. სასრული ავტომატი, წარმოდგენილი სქემის სახით, k შესასვლელით, m მეხსიერებითა და n გამოსასვლელით მოყვანილია ნახ. 1 ა-ზე, ხოლო მისი განზოგადოებული სახე მოყვანილია ნახ. 1 ბ-ზე. ავტომატი შეიცავს კომბინაციურ სქემასა და m რაოდენობის მეხსიერების წყვილ უჯრედებს, რომელთა მარცხენა ნაწილში ჩაწერილია (შენახულია) მიმდინარე დროში არსებული სიმბოლოები, ხოლო მარჯვენა ნაწილში დროის მომდევნო ინტერვალში არსებული სიმბოლოები. მოცემულ შემთხვევაში სასრული ავტომატის მეხსიერება განხილულა როგორც წყვილ უჯრედიანი ძვრის რეგისტრის ელემენტი. კომბინაციური სქემის მეშვეობით ხორციელდება შესასვლელი და მეხსიერებაში არსებული სიმბოლოების გარკვეული წესით ურთიერთქმედება (შეკრება, გამრავლება და ა.შ.), რის შედეგადაც ვღებულობთ გამოსასვლელ სიმბოლოებს. ავტომატის შესასვლელზე, შიდა მეხსიერებაში და გამოსასვლელზე არსებული შესაძლო სიმბოლოები მიეკუთვნებიან შემდეგ შესაბამის სიმრავლეებს (ალფაბეტებს): $U^1, U^2, \ldots,$

U^k; $S^1, S^2, \ldots, S^m$; $V^1, V^2, \ldots, V^n$; ხოლო ზოგადად ავტომატის შესასვლელი, შიდა და გამოსასვლელი ალფაბეტები განისაზღვრებიან, როგორც დეკარტეს ნამრავლები შესაბამისი ალფაბეტებისა: $\boldsymbol{U} = U^1 \text{x} U^2 \text{x} \ldots \text{x} U^k$; $\boldsymbol{S} = S^1 \text{x} S^2 \text{x} \ldots \text{x} S^k$; $\boldsymbol{V} = V^1 \text{x} V^2 \text{x} \ldots \text{x} V^k$.



ნახ. 1. სასრული ავტომატი (ა) და მისი ზოგადი სახე (ბ).

სასრული A ავტომატი ხასიათდება ხუთეულით A = (***U, S, V***, φ, Ψ), სადაც φ-ს უწოდებენ ავტომატის გადასვლების ფუნქციას, რომლის შესაბამისადაც $s(t_{i+1}) = \varphi(u(t_i),s(t_i))$; ეს იმას ნიშნავს, რომ ავტომატის შიდა მდგომარეობა t_{i+1} დროის მომენტში განისაზღვრება დროის t_i მომენტში შესასვლელზე არსებული და იმავე დროის მომენტში მეხსიერებაში არსებული სიმბოლოებით. Ψ-ს უწოდებენ გამოსასვლელების ფუნქციას და $v(t_i)= \Psi(u(t_i),s(t_i))$, შესაბამისად ავტომატის გამოსასვლელზე, t_i მომენტში არსებული სიმბოლოები განისაზღვრება დროის იმავე მომენტში ავტომატის მეხსიერებაში და შესასვლელზე არსებული სიმბოლოებით; ასეთ ავტომატს მილის ავტომატს უწოდებენ. იმ შემთხვევაში, თუ ავტომატის გამოსასვლელზე არსებული სიმბოლოების განსაზღვრისათვის საკმარისია მხოლოდ მის მეხსიერებაში არსებული სიმბოლოები ე.ი. $v(t_i) = \Psi(s(t_i))$, ავტომატს მურის ავტომატი ეწოდება.

ავტომატის მდგომარეობებს ეწოდებათ ექვივალენტური (განუსხვავებადი), თუ მათთვის ნებისმიერი ერთნაირი შესასვლელი სიმბოლოების დროს გვაქვს ერთნაირი გამოსასვლელი სიმბოლოები. წინააღმდეგ შემთხვევაში ავტომატის მდგომარეობები განსხვავებადია.

ხუთეული A = (***U, S, V***, φ, Ψ), ყოველგვარი დამატებითი საწყისი პირობების გარეშე განსაზღვრავს არაინიცირებულ სასრულ ავტომატს. ასეთ ავტომატს ეწოდება დაყვანილი, თუ მისი შიდა მდგომარეობები წყვილ-წყვილად განსხვავებადია.

იმ შემთხვევაში, თუ განვიხილავთ ორ ავტომატს, ანალოგიურ შემთხვევაში გვექნება ექვივალენტური ავტომატები ან განსხვავებადი ავტომატები.

ნებისმიერი ავტომატისთვის შეიძლება აიგოს შესაბამისი განსხვავებადი ავტომატი. ექვივალენტური ავტომატებიდან მინიმალური მეხსიერების მქონე ავტომატს მინიმალური ფორმის მქონე ავტომატი ეწოდება.

ზოგადად ავტომატის სირთულეს განსაზღვრავს კომბინაციური სქემის სირთულე და შიდა მეხსიერების ზომა, ამ დროს სასურველი შედეგი მიიღწევა ურთიერთკომპრომისის ხარჯზე.

ნახ. 1-ზე მოყვანილ სქემებში სიმბოლოები წარმოდგენილნი იყვნენ თავიანთი მნიშვნელობებით დროის t_i და t_{i+1} ტაქტური მომენტებისათვის. ცხადია, ისინი ავტომატის შესასვლელზე, შიდა მეხსიერებაში და გამოსასვლელზე არსებობენ ტაქტური მომენტების გარკვეული მიმდევრობისათვის და შეიძლება ვიმსჯელოთ შესაბამის სიმბოლოთა მიმდევრობებზეც. მაგალითად შეგვიძლია განვიხილოთ სიმბოლოთა მიმდევრობები ავტომატის რომელიმე შესასვლელისათვის

$$\underline{u} = u_0(t'_0), u_1(t'_1), u_2(t'_2), \ldots,$$

და რომელიმე გამოსასვლელისათვის

$$\underline{v} = v_0(t''_0), v_1(t''_1), v_2(t''_2), \ldots;$$

ან უფრო მარტივად თუ ჩავწერთ

$$\underline{u} = u_0, u_1, u_2, \ldots,$$

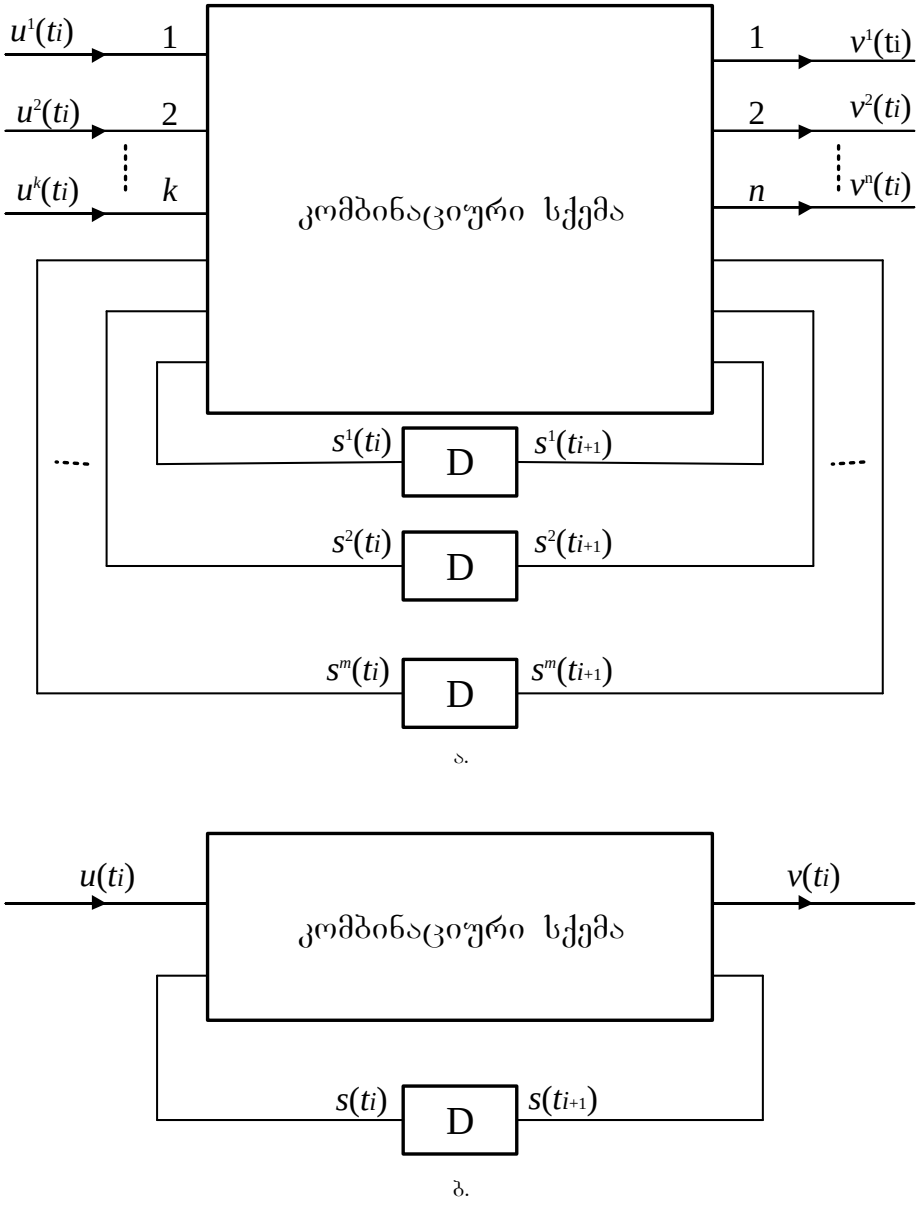
$$\underline{v} = v_0, v_1, v_2, \ldots;$$

მოცემულ შემთხვევაში, თუ გამოვიყენებთ დაყოვნების ალგებრულ D ოპერატორს, შესაბამისად გვექნება [3]:

$$\begin{aligned} u(D) &= u_0D^0 + u_1D^1 + u_2D^2 + \ldots, \\ v(D) &= v_0D^0 + v_1D^1 + v_2D^2 + \ldots. \end{aligned} \tag{1}$$

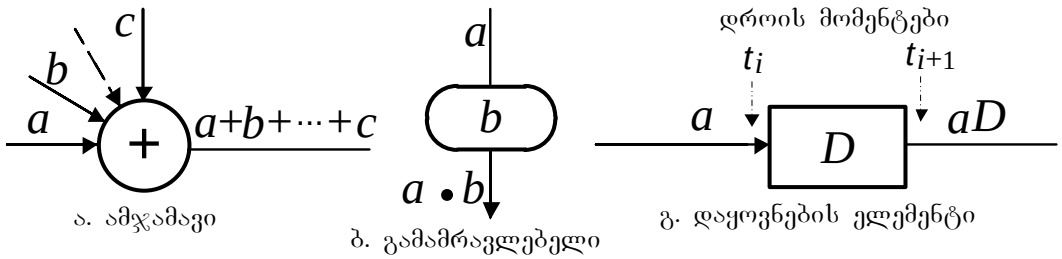
სიმბოლოთა მიმდევრობების ამგვარი წარმოდგენა შეიძლება გამოყენებული იქნას სასრული ავტომატის ნებისმიერი შესასვლელისათვის, ნებისმიერი გამოსასვლელისათვის და ასევე შიდა მეხსიერებისათვისაც. მოცემულ შემთხვევაში D-ს ხარისხი მიუთითებს სიმბოლოს დაყოვნებაზე შესაბამისი ტაქტური ინტერვალით. ცხადია

დაყოვნების ოპერატორი შეიძლება გამოყენებული იქნას დაყოვნების ელემენტების წარმოდგენის დროსაც და მაშინ ნახ. 1 მიიღებს, ქვემოთ, ნახ. 2–ზე მოყვანილ სახეს. მოცემულ შემთხვევაში წყვილ უკრედიანი ძვრის რეგისტრები შეცვლილია შესაბამისი დაყოვნების ელემენტებით.



ნახ. 2. სასრული ავტომატი (ა) და მისი ზოგადი სახე (ბ) დაყოვნების ელემენტებით.

ნახ. 3 ა-ზე მოყვანილია ამჯამავი, ხოლო ნახ. 3 ბ-ზე მოყვანილია გამამრავლებელი, ორივე შემთხვევაში შეკრებისა და გამრავლების ოპერაციები ხორციელდება შესაბამის ალგებრულ სტრუქტურაში (ჯგუფი, რგოლი, ველი, ...). ნახ. 3 გ-ზე ნაჩვენებია დაყოვნების ელემენტი, რომელსაც აქვს მეხსიერება როგორც შესასვლელის, ასევე გამოსასვლელის მხრიდან (ფაქტობრივად ის წარმოადგენს ძვრის რეგისტრის ორ უჯრედს) და ახორციელებს სიმბოლოს დაყოვნებას ერთი ტაქტური ინტერვალით.



ნახ. 3. სასრული ავტომატის შესაძლო შემადგენელი ელემენტები.

სასრული ავტომატის იმპულსური რეაქცია ($I(t)$) არის რეაქცია მის გამოსასვლელებზე, როცა შესასვლელებზე მოქმედებენ დისკრეტული დელტა იმპულსები:

$$\sigma(t_n) = \begin{cases} 1, \text{როცა } n = 0; \\ 0, \text{როცა } n \neq 0; \end{cases}$$

თუ ავტომატი ციფრულია, მაშინ 1 და 0 ციფრული სიმბოლოებია.

იმ შემთხვევაში, თუ ვიცით $u(t)$ სიმბოლოები სასრული ავტომატის შესასვლელზე და იმპულსური რეაქცია მის გამოსასვლელზე, $v(t)$ გამოსასვლელი სიმბოლოები განისაზღვრებიან როგორც დისკრეტული ნახვევი შესასვლელი სიმბოლოებისა და შესაბამისი იმპულსური რეაქციისა:

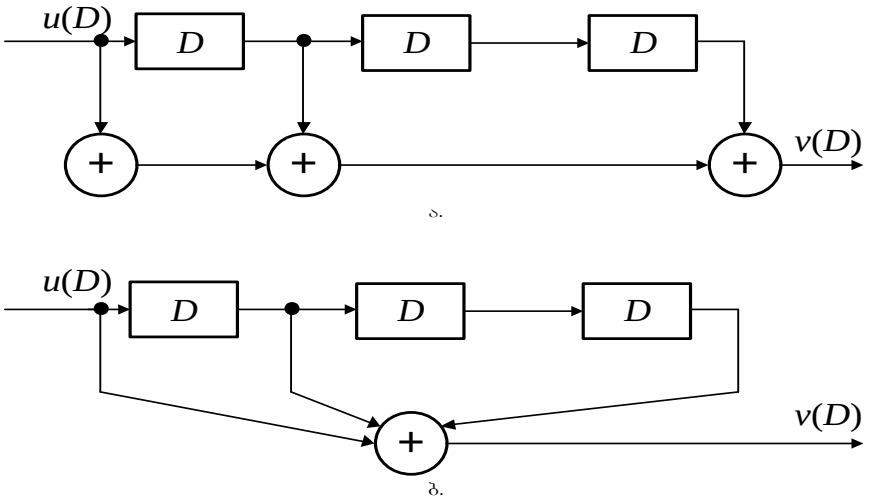
$$v(t_j) = \sum_{i=0}^{j} u(t_i) \cdot I(t_{j-1})$$

რომელშიც, max(*j*)=*L*+1 (სადაც *L* იმპულსური რეაქციის სიგრძეა ანუ სიმბოლოთა რაოდენობაა *I*(*t*)-ში, რის შემდეგადაც ავტომატის შიდა მდგომარეობა აღდგენილია საწყის მნიშვნელობამდე ანუ რეაქცია გამოსასვლელზე დასრულებულია). მოყვანილ გამოსახულებასთან დაკავშირებით, უნდა შევნიშნოთ, რომ ზოგადად *u*(*t*) და *I*(*t*) ფუნქციების ნახვევი ეწოდება ქვემოთ მოყვანილ ინტეგრალს [4] :

$$v(t) = \int_{-\infty}^{\infty} u(\tau) \cdot I(t-\tau) d\tau,$$

რომელსაც, დიუჰამელის ინტეგრალსაც უწოდებენ; ის განსაზღვრავს *v*(*t*) ფუნქციას სისტემის გამოსასვლელზე, თუ შესასვლელზე გვაქვს ფუნქცია *u*(*t*) და სისტემის იმპულსური რეაქციაა *I*(*t*).

ნახ. 4-ზე, მოყვანილია მაგალითი ციფრული სასრული ავტომატისა იმპულსური რეაქციით *I*=1,1,0,1.



ნახ. 4. სასრული ავტომატის მაგალითი (ა) და მისი გამარტივებული სახე (ბ).

აქ ავტომატის გამარტივებული სახით წარმოდგენა ეყრდნობა იმ ფაქტს, რომ ავტომატის შესასვლელზე, მეხსიერებაში და გამოსასვლელზე არსებული სიმბოლოები ეკუთვნიან ისეთ ალგებრულ სტრუქტურას (ჯგუფს, რგოლს და ა.შ.), სადაც ადგილი აქვს შეკრების ასოციატიურობას. სიმარტივის მიზნით, მიმდევრობები შეგვიძლია გამოვსახოთ მძიმეების გარეშე და გვექნება I = 1101; თუ ვისარგებლებთ დაყოვნების ოპერატორით, მაშინ $I(D)=1+D+D^3$. დაყოვნების ოპერატორის გამოყენების გვექნება:

$$v_j D^i = \sum_{i=0}^{j} u_i D^i \cdot I_{j-i} D^{j-i} = \sum_{i=0}^{j} u_i \cdot I_{j-i} D^j ;$$

(1)-ის გათვალისწინებით

$$\sum_{j=0}^{L+1} v_j D^j = \sum_{i=0}^{L+1} \sum_{i=0}^{j} u_i \cdot I_{j-i} D^j ,$$

ანუ შეგვიძლია დავწეროთ

$$v(D) = u(D) \cdot I(D); \qquad (3)$$

აქედან ჩანს, რომ ნახ. 4-ზე მოყვანილი სქემა უზრუნველყოფს ორი $u(D)$ და $I(D)$ მრავალწევრის ერთმანეთზე გამრავლებას. აქვე შეგვიძლია ჩავთვალოთ, რომ $I(D)$ არის ავტომატის გადაცემის ფუნქციაც. ე.ი. საბოლოოდ ვასკვნით, რომ თუ მოცემულია ავტომატის გადაცემის ფუნქცია მრავალწევრის სახით, მაშინ მის მიხედვით შეგვიძლია მოვახდინოთ მრავალწევრთა გამამრავლებელი შესაბამისი სქემის შედგენა ანუ სინთეზი.

როგორც მოყვანილი მაგალითებიდან ჩანს, ჩვენს მიერ განხილული სასრული ავტომატები მათ შესასვლელზე არსებულ სიმბოლოთა მიმდევრობას, გარკვეული წესით, გარდაქმნის გამოსასვლელზე არსებულ სიმბოლოთა მიმდევრობაში ანუ კოდურ მიმდევრობაში. აღნიშნულ პროცესს კოდირების პროცესი ანუ მოკლედ კოდირება ეწოდება, ხოლო

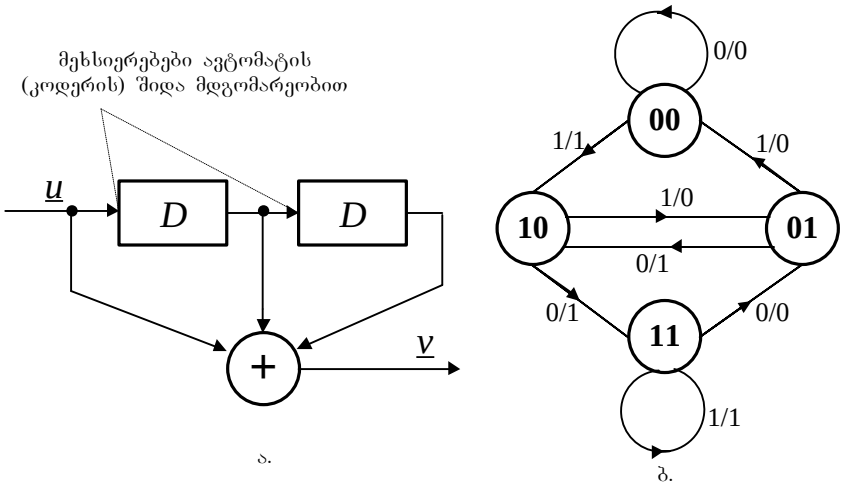
შესაბამის სასრულ ავტომატს კოდერი. კოდური მიმდევრობების სიმრავლეს კოდი ეწოდება. იმ შემთხვევაში, თუ კოდი მიიღება სასრული ავტომატის შესასვლელზე არსებული სიმბოლოებისა და ავტომატის იმპულსური რეაქციის ხვევის (რასაც ჩვენს მიერ განხილულ შემთხვევაში ჰქონდა ადგილი) შედეგად, მას ხვეულა კოდი ეწოდება.

დღეისათვის არსებული უმრავლესი კოდები აგებული არიან ისეთ მათემატიკურ სტრუქტურაზე, როგორებიცაა რგოლი და ველი, რაც იმას ნიშნავს, რომ კოდის ყველა სიმბოლო ეკუთვნის მოცემულ რგოლს ან ველს, ხოლო კოდირებისას არითმეტიკული ოპერაციები ხორციელდება შესაბამის სტრუქტურაში. ამ შემთხვევაში, თუ ველი ან რგოლი q-ობითია შესაბამისი კოდიც *q*-ობითია და *q*-ს კოდის ძირს უწოდებენ.

სასრული ავტომატები შეიძლება აღწერილი იქნან აწონილი ორიენტირებული გრაფების საშუალებით, რომლის დროსაც გრაფის წვეროებს მიეწერებათ ავტომატის შიდა მდგომარეობების მნიშვნელობები (ანუ დროის მოცემულ მომენტში ავტომატის მეხსიერებაში არსებულ სიმბოლოთა შესაძლო მიმდევრობები ანუ კომბინაციები). ცხადია, ამ დროს, გრაფის რიგი (წვეროების რაოდენობა) ტოლია ავტომატის შიდა მდგომარეობების რიცხვისა. ავტომატის ერთი მდგომარეობიდან მეორეში გადასვლას შეესაბამება მოცემულ წვეროთა შემაერთებელი წიბო, რომელსაც მიწერილი აქვს წილადი, სადაც მრიცხველი არის ავტომატის გამოსასვლელზე არსებული სიმბოლოები, ხოლო მნიშვნელი არის ამ დროს ავტომატის შესასვლელზე არსებული სიმბოლოები. მოვიყვანოთ ერთი მაგალითი.

დავუშვათ გვაქვს ნახ. 5-ზე ნაჩვენები სასრული ავტომატი, რომლის შესასვლელი, შიდა და გამოსასვლელი სიმბოლოები ეკუთვნიან ორობით გალუას, GF(2), ველს. ცხადია ავტომატის (კოდერის) გამოსასვლელზე არსებული კოდური მიმდევრობები ქმნიან ორობით კოდს. ამ შემთხვევისათვის $v(t_i)=\varphi(u(t_i), s^1(t_i), s^2(t_i), s^3(t_i)) = \varphi(s^1(t_i), s^2(t_i), s^2(t_{i-1}))$, ე. ი. კოდერის გამოსასვლელზე არსებული სიმბოლო დროის ნებისმიერ

მომენტში განისაზღვრება დაყოვნების ელემენტების შესასვლელზე არსებული სიმბოლოებით. ამასთან დაკავშირებით საკმარისია კოდერის მდგომარეობა დროის ნებისმიერმომენტში წარმოდგენილი იქნას მისი დაყოვნების ელემენტების შესასვლელზე არსებული სიმბოლოებით.



ნახ. 5. ორობითი ხვეულა კოდის მაგალითი. ა-კოდერი, ბ-მისი აღმწერი გრაფი.

ამიტომ ნახ. 5-ზე მოყვანილი კოდერი შეგვიძლია წარმოვადგინოთ მდგომარეობით – 00, 10, 11, 01. ავტომატის აღმწერიგრაფი მოყვანილია ნახ. 5. ბ-ზე.

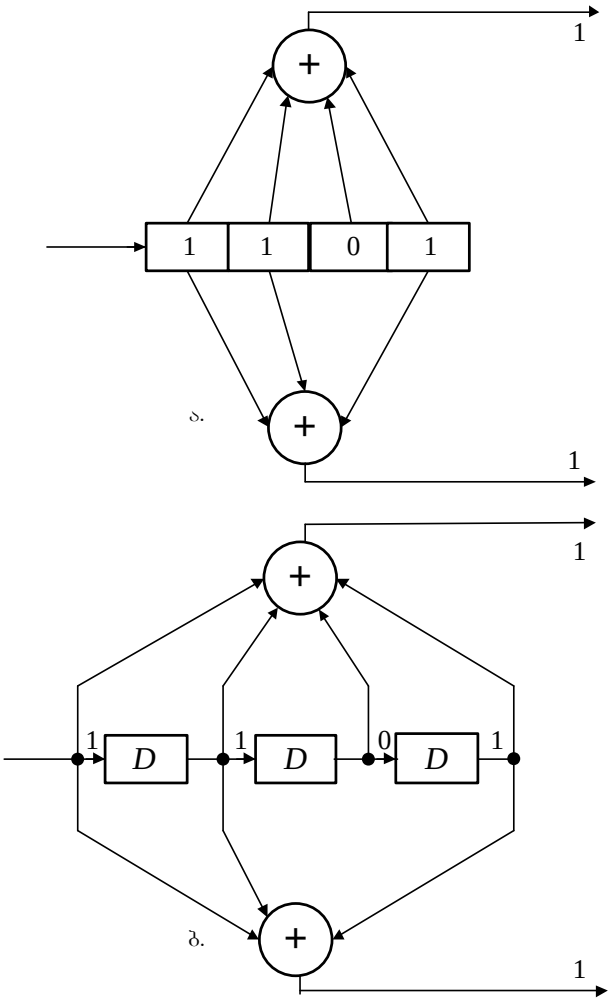
ცხრილი1. მონაცემები ავტომატის გრაფის ასაგებად

შესაძლო მიმდინარე შიდა მდგომარეობები	მომდევნო შიდა მდგომარეობები		გამოსასვლელი სიმბოლოები	
	შესასვლელზე 0 - ია	შესასვლელზე 1 - ია	შესასვლელზე 0 - ია	შესასვლელზე 1 - ია
0 0	0 0	1 0	0	1
1 0	0 1	1 1	1	0
0 1	0 0	1 0	1	0
1 1	0 1	1 1	0	1

1.2 ხვეულა კოდების აღწერა და მათი პარამეტრები

ხვეულა კოდის ფორმირების პროცესი (ხვეულა კოდის კოდერი) შეიძლება აღიწეროს გრაფის საშუალებით და მას კოდერის მდგომარეობათა დიაგრამას უწოდებენ. ამასთან დაკავშირებით განვიხილოთ ერთი მაგალითი. ვთქვათ გვაქვს ორობით გალუას ველზე (GF(2)) მოცემული ხვეულა კოდის კოდერი ერთი შესასვლელით (k=1) და ორი გამოსასვლელით (n=2). დაუშვათ, რომ მას აქვს ნახ. 6-ზე მოყვანილი სახე. აქ კოდერის შიდა მეხსიერება წარმოდგენილი არის ძვრის რეგისტრის უჯრედებისა და დაყოვნების ელემენტების საშუალებით; კოდერის მდგომარეობა წარმოვადგინოთ მის დაყოვნების ელემენტების შესასვლელებზე არსებული სიმბოლოებით და შევადგინოთ ცხრილი 2. მოცემული ცხრილის შესაბამისად აგებულ, ნახ. 6-ზე მოყვანილ კოდერის, მდგომარეობათა დიაგრამას აქვს ნახ. 7-ზე ნაჩვენები სახე.

დიაგრამის წიბოებს მიწერილი წონების, წილადი რიცხვების, მნიშვნელები შეესაბამებიან სიმბოლოებს კოდერის შესასვლელზე (საინფორმაციო სიმბოლოებს), ხოლო მრიცხველები – კოდერის გამოსასვლელებზე არსებულ შესაბამის სიმბოლოებს (კოდურ სიმბოლოებს). ცხადია მოცემული გრაფი (ხვეულა კოდის კოდერის მდგომარეობათა დიაგრამა) სრულად აღწერს კოდირების პროცესს. მაგალითად, თუ მოცემული კოდერის საწყისი მდგომარეობა იყო 000, შესასვლელზე გვქონდა საინფორმაციო სიმბოლოთა მიმდევრობა 10111, მაშინ, მისი შესაბამისი, კოდერის მდგომარეობათა მიმდევრობა იქნება 000 100 010 101 110 111 და შედეგად კოდერის გამოსასვლელზე მივიღებთ კომბინაციას 1111011110. შესაბამისი გზა ნახ. 7-ზე წარმოდგენილია შედარებით მუქი წიბოებით.

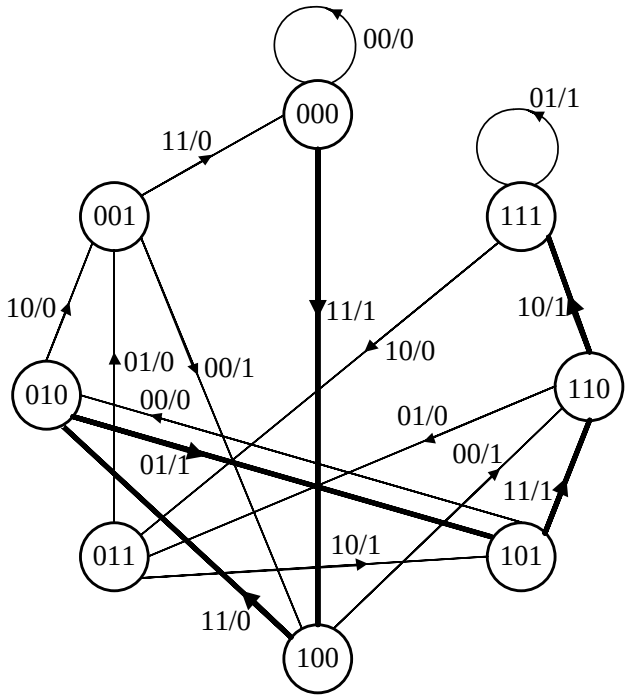


ნახ. 6. ხვეულა კოდის კოდერის მაგალითი:
ა. მეხსიერება წარმოდგედგენილია ძვრის რეგისტრის უჯრედებით
ბ. მეხსიერება წარმოდგენილია დაყოვნების ელემენტებით.

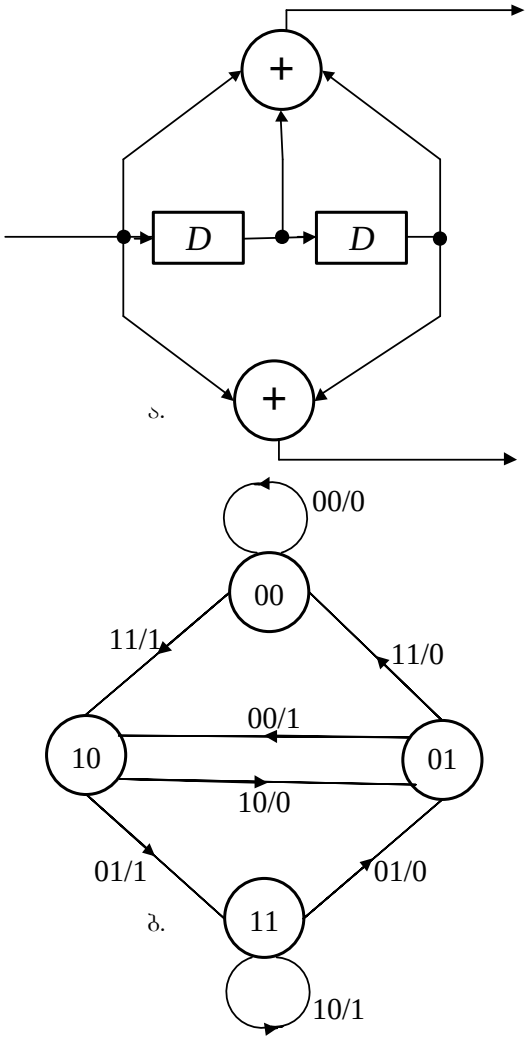
აღვწეროთ, ხვეულა კოდის კოდერი ხის საშუალებით. სიმარტივისათვის, განვიხილოთ ნახ. 8 ა-ზე მოყვანილი ორობითი კოდის კოდერი. მისი შესაბამისი კოდური ხე ნაჩვენებია ნახ. 9-ზე.

ცხრილი 2. მონაცემები კოდერის გრაფის ასაგებად

შესაძლო მიმდინარე მდგომარეობები	მომდევნო მდგომარეობები		გამოსასვლელი სიმბოლოები	
	შესასვლელზეა 0	შესასვლელზეა 1	შესასვლელზეა 0	შესასვლელზეა 1
0 0 0	0 0 0	1 0 0	0 0	1 1
0 0 1	0 0 0	1 0 0	1 1	0 0
0 1 0	0 0 1	1 0 1	1 0	0 1
0 1 1	0 0 1	1 0 1	0 1	1 0
1 0 0	0 1 0	1 1 0	1 1	0 0
1 0 1	0 1 0	1 1 0	0 0	1 1
1 1 0	0 1 1	1 1 1	0 1	1 0
1 1 1	0 1 1	1 1 1	1 0	0 1



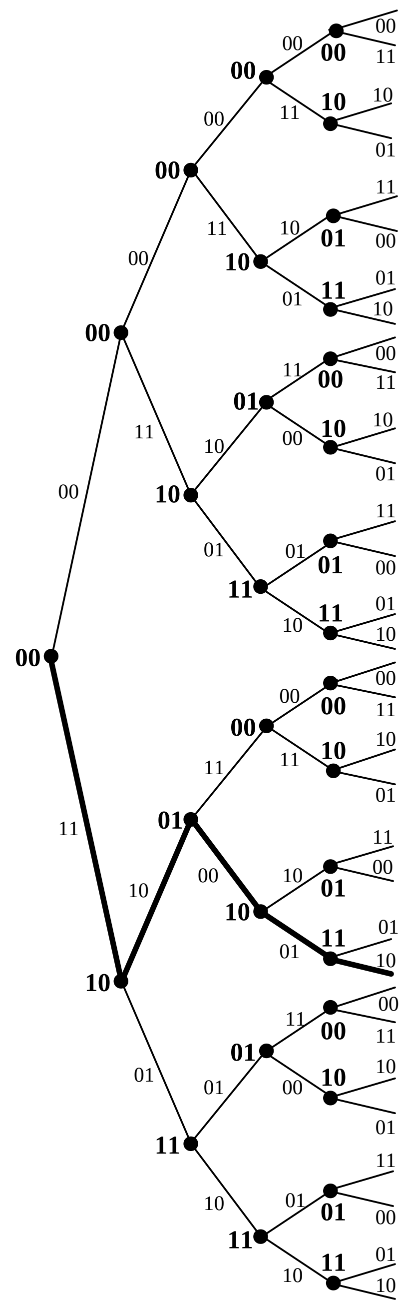
ნახ. 7. 6-ე ნახაზზე მოყვანილი ხვეულა კოდის კოდერის მდგომარეობათა დიაგრამა.



ნახ. 8. ხვეულა კოდის კოდერი

(ა) ოთხი მდგომარეობით და მისი მდგომარეობათა დიაგრამა (ბ).

ის აგებულია ნახ. 8 ბ-ზე მოყვანილი კოდერის მდგომარეობათა დიაგრამის (გრაფის) მიხედვით. ამ დროს ხის წვეროებს შეესაბამებათ კოდერის მდგომარეობების მნიშვნელობები, ხოლო წიბოებს – კოდერის გამოსასვლელზე არსებული (კოდური) სიმბოლოები.



ნახ. 9. მე-8 ნახ.-ზე მოყვანილი კოდერის კოდური ხის ფრაგმენტი.

წიბოს მიმართულება ზემოდან ქვემოთ შეესაბამება საინფორმაციო სიმბოლო 1-ს, ხოლო მიმართულება ქვემოდან ზემოთ – 0-ს. 10111 საინფორმაციო მიმდევრობის შესაბამისი გზა ხეზე შედარებით მუქი წიბოებითაა ნაჩვენები. აქაც, როგორც წინა მაგალითში ჩავთვალეთ, რომ

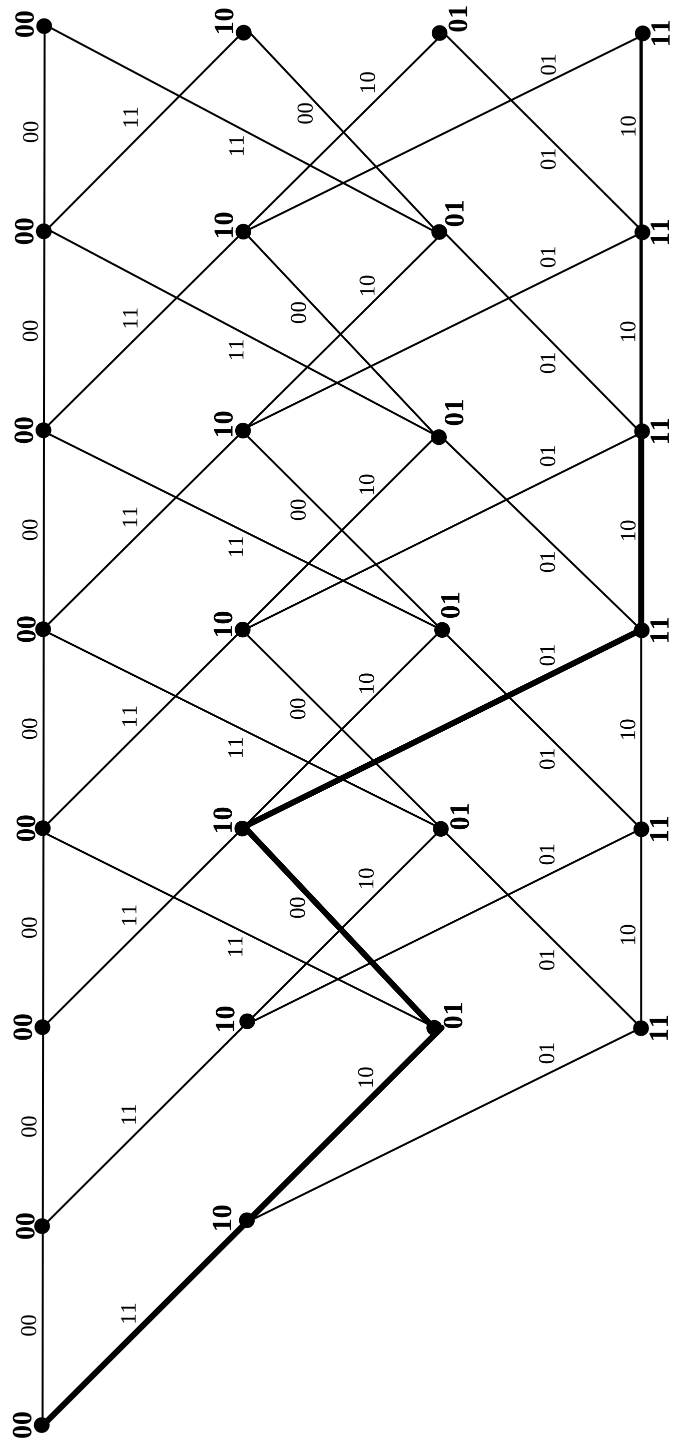
კოდერის საწყისი მდგომარეობა ნულოვანია. ჩვენს შემთხვევაში ის შეესაბამება ხის ძირს. ცხადია, მოყვანილი ნახ. 9-დან, სათანადო კოდური მიმდევრობაა 1110000110.

კოდერის მდგომარეობათა დიაგრამის დროში გაშლით (ტაქტური ინტერვალების მიხედვით) მიიღება ორიენტირებული გრაფი, რომელიც აღწერს კოდირების პროცესს და მას კოდერის გისოსისებური დიაგრამა ანუ კოდის გისოსი ეწოდება.

მაგალითისათვის, ნახ. 8 ა-ზე ნაჩვენები კოდერის შესაბამისი ხვეულა კოდის გისოსი მოყვანილია ნახ. 10-ზე. ტაქტური მომენტების შესაბამის კვეთს გისოსზე იარუსი ეწოდება [5]. აქ და მომავალში, გისოსის წვეროს, კვანძს ვუწოდებთ, ხოლო წიბოს – შტოს. ვინაიდან შტოები გისოსზე შეესაბამებიან კოდური სიმბოლოების გადაცემას დროში მარცხნიდან მარჯვნივ, ისინი გამოსახულები არიან მიმართულებების მაჩვენებლების გარეშე.

10111 საინფორმაციო მიმდევრობის შესაბამისი გზა გისოსზე, ნახ. 9 ა-ზე მოყვანილი ხვეულა კოდის კოდერისათვის, ნაჩვენებია შედარებით მუქი შტოებით. ამ დროს გვაქვს კოდერის მდგომარეობათა 00 10 01 10 11 11 მიმდევრობა და კოდური კომბინაცია 1110000110.

ნახ. 7-ზე და ნახ. 8 ბ-ზე ჩანს, რომ მოყვანილი გრაფები არასრულია; ასევე ჩანს, რომ ნახ. 10-ზე წარმოდგენილ გისოსზე ყველანაირი გზა არ არსებობს; სწორედ ეს ფაქტები მიუთითებენ იმაზე, რომ მოცემულ შემთხვევებში არსებობენ აკრძალული კოდური კომბინაციები, ე.ი. გვაქვს კოდირება სიჭარბით.



ნახ. 10. ხვეულა კოდის გისოსის ფრაგმენტი.

კოდის სიჩქარე განისაზღვრება გამოსახულებიდან *R=k/n,* სადაც *k* არის გარკვეული ალფაბეტის საინფორმაციო სიმბოლოების რაოდენობა,

რომლებიც ერთდროულად მიეწოდებიან კოდერის შესასვლელებზე, ხოლო n არის შესაბამისი, იმავე ალფაბეტის, კოდური სიმბოლოების რაოდენობა. თუ გვაქვს ისეთი კოდერი, რომლის ყველა k შესასვლელზე საინფორმაციო სიმბოლოების მიწოდება ხდება ერთდროულად და იმავე დროში კოდერის n გამოსასვლელზე ვღებულობთ n კოდურ სიმბოლოს, მაშინ კოდის სიჩქარე ტოლი იქნება კოდერის შესასვლელებისა და გამოსასვლელების რაოდენობათა თანაფარდობისა.

მეხსიერების ელემენტების რაოდენობას კოდერში ხშირად აღნიშნავენ K-თი და მას მაკოდირებელი რეგისტრის სიგრძეს უწოდებენ. მაგალითად, ნახ. 6-ზე მოყვანილი კოდისათვის K=4, ნახ. 8-ზე მოყვანილი კოდისათვის K=3.

ხვეულა კოდის კოდერის i-ური შესასვლელისათვის რეაქციის სიგრძე არის სიდიდე

$$\gamma_i = \max_j [deg\ G_{ij}(D)];$$

მოყვანილ გამოსახულებაში i არის $G(D)$ მატრიცის სტრიქონის ნომერი, ხოლო j სვეტის ნომერია.

მთლიანობაში, შესასვლელთათვის, კოდერის რეაქციის სიგრძე $\gamma = \sum_{i=1}^{k} \gamma_i$; ხშირად ამ სიდიდეს კოდური შეზღუდვის სიგრძესაც უწოდებენ [6]. კოდს, როცა ნებისმიერი i და j – თვის სრულდება პირობა,

$$\gamma_0 = \max_{i,j} [\deg G_{ij}(D)] \le 1,$$

ეწოდება ერთეულოვანი მეხსიერების მქონე კოდი [7]. კოდერისთვის, რომელსაც აქვს k შესასვლელი

$$K = \gamma + k$$

მოცემულ მეტრიკაში l რიგის სვეტური მანძილი განისაზღვრება როგორც მინიმალური მანძილი, კოდური ხის ძირიდან გამოსულ განსხვავებული პირველი წიბოების მქონე, l+1 რაოდენობის საინფორმაციო სიმბოლოს შესაბამის კოდურ მიმდევრობებს შორის.

ქვემოთ მოყვანილია (7,5) ხვეულა კოდის კოდური კომბინაციები, რომლებიც ყოველი l-ისათვის გაყოფილი არიან ორ ჯგუფად და ამ ჯგუფებს განსხვავებული აქვთ ხის ძირიდან გამომავალი პირველივე წიბოები. მანძილთა ერთობლიობას d=(d_0, d_1, d_2, d_3, d_4,. . .) ხვეულა კოდის დისტანციურ პროფილს უწოდებენ. მოყვანილი მაგალითისათვის ჰემინგის მეტრიკაში d=(2, 3, 3, 4, 4, . . .) .

ხვეულა კოდის მინიმალური მანძილი არის სვეტური მანძილი $l=\gamma$ შემთხვევისათვის ანუ γ რიგის სვეტური მანძილი. მას აღნიშნავენ d_{min}–ით და q–ობითი კოდისათვის

$$d_{\min} \le \left\lfloor ((q-1)/q)\cdot(n/(q^{R\cdot n}-1)+L) \right\rfloor$$

$\lfloor x \rfloor$ აღნიშნავს x–ზე ნაკლებ ან ტოლ უდიდეს მთელ რიცხვს, ხოლო

$$L=(\gamma_0+1)n.$$

ზოგჯერ ამ სიდიდესაც კოდური შეზღუდვის სიგრძეს უწოდებენ.

ხვეულა კოდის თავისუფალი მანძილი არის სვეტური მანძილი $l\to\infty$ შემთხვევისათვის. მას აღნიშნავენ d_{free} - ით.

თავისუფალი ჰემინგის მანძილის ზედა საზღვარი R=1/n სიჩქარიანი ორობითი ხვეულა კოდებისათვის მიღებული იქნა ჰელერის მიერ [8], რომელიც R-ის რაციონალური მნიშვნელობებისათვის მოდიფიცირებული იქნა [9]-ში შემდეგი სახით:

$$d_{free} \le \min_{I\le i} \left\lfloor (2^{i-1}/(2^i-1))\cdot(K+i-k)\cdot(n/k) \right\rfloor,$$

რომელშიც

$$I = \begin{cases} 1, \text{ თუ } K < 2k-1; \\ k, \text{ თუ } K \geq 2k-1. \end{cases}$$

l=0	l=1	l=2	l=3	l=4
00	0000	000000	00000000	0000000000
	0011	000011	00000011	0000000011
11		001110	00001110	0000001110
d_0 = 2	1110	001101	00001101	0000001101
	1101		00111011	0000111011
		111011	00111000	0000111000
	d_1 = 3	111000	00110101	0000110101
		110101	00110110	0000110110
		110110		0011101100
			11101100	0011101111
		d_2 = 3	11101111	0011100010
			11100010	0011100001
			11100001	0011010111
			11010111	0011010100
			11010100	0011011001
			11011001	0011011010
			11011010	
				1110110000
			d_3 = 4	1110110011
				1110111110
				1110111101
				1110001011
				1110001000
				1110000101
				1110000110
				1101011100
				1101011111
				1101010010
				1101010001
				1101100111
				1101100100
				1101101001
				1101101010
				d_4 = 4

ხვეულა კოდის r რიგის სტრიქონული მანძილი განისაზღვრება როგორც მინიმალური მანძილი კოდის გისოსზე იმ წყვილ გზების შესაბამის კოდურ მიმდევრობებს შორის, რომლებიც ჯერ იყოფიან, ხოლო r შტოს შემდეგ ისევ ერწყმიან ერთმანეთს. მას ჩვენ აღვნიშნავთ d_r სიმბოლოთი.

კატასტროფული კოდების გამოყენება დეკოდირებისას გვაძლევს შეცდომების უსასრულო რაოდენობას, როცა არხში შეცდომების რაოდენობა სასრულია.

დავუშვათ $R=k/n$ სიჩქარიანი კოდის წარმომქმნელი G მატრიცა შეიცავს C_n^k რაოდენობის k x n ზომის G_e ქვემატრიცებს (e=1,2,….., C_n^k). ვთქვათ, თითოეული მატრიცის დეტერმინანტია det G_e; მაშინ ხვეულა კოდი არ იქნება კატასტროფული, თუ

$$GCD[\det G_e,\ e=1,2,\ldots, C_n^k]=D^a,$$

სადაც a ნებისმიერი მთელი არაუარყოფითი რიცხვია.

თუ $R=1/n$, მაშინ ხვეულა კოდის არაკატასტროფულობის პირობა ჩაიწერება შემდეგნაირად [10]:

$$GCD[G_1(D), G_2(D),\ldots, G_n(D)]=D^a.$$

1.3 ხვეულა კოდების დეკოდირება მაქსიმალური დამაჯერებლობის პრინციპით. ვიტერბის ალგორითმი.

დეკოდირების მეთოდი მაქსიმალური დამაჯერებლობის პრინციპით წარმოადგენს დეკოდირების ტიპიურ ალგორითმს [11], რომელიც დაფუძნებულია მიღებული სიმბოლოების ალბათურ მახასიათებლების გამოყენებაზე. ალგორითმი ფართოდ გამოიყენება მოკლე ხვეულა კოდების

დეკოდირებისას. ქვემოთ, დეკოდირების ალგორითმი განხილულია ხვეულა კოდის მაგალითზე სიჩქარით $R=1/n$.

დეკოდირების პროცესში კოდის გისოსზე გზა იწყება მომენტში, როცა $t = 0$ და კოდერს შესასვლელზე მიეწოდება L სიგრძის საინფორმაციო მიმდევრობა $u_L = (u_0\ u_1\ u_2\ .\ .\ .\ u_{L-1})$. ხოლო გამოსასვლელზე იქნება სიმბოლოები $a_L = (a_0 a_1 a_2 ... a_{L-1})$. კოდერის მდგომარეობა t მომენტისათვის განისაზღვრება როგორც ნაკრები ν საინფორმაციო მიმდევრობიდან $\omega_t = (u_t\ u_{t-1}\ .\ .\ .\ u_{t-\ \nu+1})$. კოდის გისოსისებური დიაგრამა ერთმნიშვნელოვნად აკავშირებს საინფორმაციო მიმდევრობას (u_L), კოდერის მდგომარეობათა მიმდევრობას (ω_t) და კოდერის გამოსასვლელზე არსებულ მიმდევრობებს (a_L).

არხში კოდის თითოეულ სიმბოლოს შეესაბამება სიგნალი, რომელიც შეიძლება წარმოდგენილი როგორც $S_t = (S_t^{(0)}\ S_t^{(1)}\ .\ .\ .\ S_t^{(N)})$ კოორდინატთა ერთობლიობა. იქ მასზე მოქმედებს ადიტიური ხმაური და შესაბამისად დეკოდერის შესასვლელზე გვაქვს მიმდევრობა $X_L = S_L + n_L$, სადაც $S_L = (S_0\ S_1\ .\ .\ .\ S_{L-1})$, $n_L = (n_0\ n_1\ .\ .\ .\ n_L)$, $n_t = (n_t^{(0)}\ n_t^{(1)}\ .\ .\ .\ n_t^{(N)})$ არის N სიგრძის ხმაურის ვექტორი. დეკოდირების პროცესი წარმოადგენს გზის ამორჩევას კოდურ გისოსზე მაქსიმალური აპოსტერიული ალბათობის პრინციპით. დეკოდირებისას გზა შეგვიძლია ავირჩიოთ რამოდენიმე მეთოდით: კოდური გზის შემადგენელი შტოების წონათა (შეფასებათა) ერთობლიობით $\hat{a}_L = (\hat{a}_0 \hat{a}_1 ... \hat{a}_{L-1})$; კოდერის მდგომარეობათა მიმდევრობით $\hat{\omega}_L = (\hat{\omega}_0 \hat{\omega}_1 ... \hat{\omega}_{L-1})$; ან კოდერის შესასვლელზე არსებული სიმბოლოთა მიმდევრობით $\hat{u}_L = (\hat{u}_0 \hat{u}_1 ... \hat{u}_{L-1})$, რომელიც ემთხვევა კოდერის მდგომარეობის პირველ სიმბოლოს. X_L მიმდევრობა დეკოდირდება შეცდომათა მინიმალური ალბათობით, თუ ყველა შესაძლო გზიდან ავირჩევთ $\hat{a}_L$ შეფასებას, რომლისთვისაც მაქსიმალურია აპოსტერიული ალბათობა $P(\hat{a}_L / X_L)$. $\hat{a}_L$ სიმბოლოთა ერთობლიობის ყველა შესაძლო ვარიანტის გადაცემა

ითვლება თანაბარალბათურად. ასეთ შემთხვევაში დეკოდირება მაქსიმალური აპოსტერიული ალბათობის კრიტერიუმით ექვივალენტურია დეკოდირებისა მაქსიმალური დამაჯერებლობის კრიტერიუმით, როცა არჩეულია $\hat{a}_L$ შეფასება, რომელიც უზრუნველყოფს $P(X_L / \hat{a}_L) = \max$ პირობას. არხში, მეხსიერების გარეშე პირობითი ალბათობა $P(X_L / \hat{a}_L)$ პროპორციულია სიგნალისა და ხმაურის ჯამის პირობით ალბათობათა სიმკვრივის ნამრავლის.

$$p(X_L / S_L) = \prod_{t=0}^{L-1} p(X_t / S_t) = \prod_{t=0}^{L-1} p(X_t^{(0)} X_t^{(1)} ... X_t^{(N)} / S_t^{(0)} S_t^{(1)} ... S_t^{(N)}) \qquad (3)$$

გაუსის არხში, N_0 სპექტრალური სიმკვრივის მქონე თეთრი ხმაურის [12] ზემოქმედებისას (3)-ს აქვს სახე:

$$p(X_L / S_L) = (1/\sqrt{\pi N_0})^N \exp\{-[\sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2] / 2N_0\}.$$

მაქსიმუმის საპოვნელად გამოვიყენოთ გალოგარითმება:

$$\ln p(X_L / S_L) = \ln \prod_{t=0}^{L-1} (1/\sqrt{\pi N_0})^N \times \exp\{-[\sum_{l=1}^{N} (X_t^{(l)} - S_t^{(i)})^2] / 2N_0\} =$$

$$= NL \ln(1/\sqrt{\pi N_0}) - \sum_{t=0}^{L-1} \sum_{i=1}^{N} (X_t^{(l)} - S_t^{(i)})^2] / 2N_0 \sum_{i=1} (X_t^{(i)} - S_t^{(i)})^2] / 2N_0.$$

დეკოდირების პროცესში ირჩევენ სიგნალის მიმდევრობას $\hat{S}_L = (\hat{s}_0 \hat{s}_1 ... \hat{s}_{L-1})$ და მასთან ერთმნიშვნელოვნად დაკავშირებულ შტოთა მიმდევრობას $\hat{a}_L = (\hat{a}_0 \hat{a}_1 ... \hat{a}_{L-1})$, რომელიც უზრუნველყოფს ჯამის მინიმუმს

$$PM = \sum_{t=0}^{L-1} \sum_{i=1}^{N} (X_t^{(l)} - S_t^{(i)})^2 = \min; \qquad (4)$$

მას გზის მეტრიკა ეწოდება და მოიცავს შემადგენელ შტოთა მეტრიკას

$$BM_t = \sum_{i=1}^{N} (X_t^{(l)} - S_t^{(i)})^2 ,$$

გაუსის არხში შტოთა მეტრიკა პროპორციულია სიგნალისა და ხმაურის მიღებულ ჯამისა (x_t) და სიგნალის ვექტორს ($\hat{s}_t$) შორის ევკლიდეს მანძილის კვადრატის, რომელიც შეესაბამება $\hat{a}_t$ კოდის სათანადო შტოები. დისკრეტულ არხებში მანძილის შესაფასებლად იყენებენ ჰემინგის მეტრიკას.

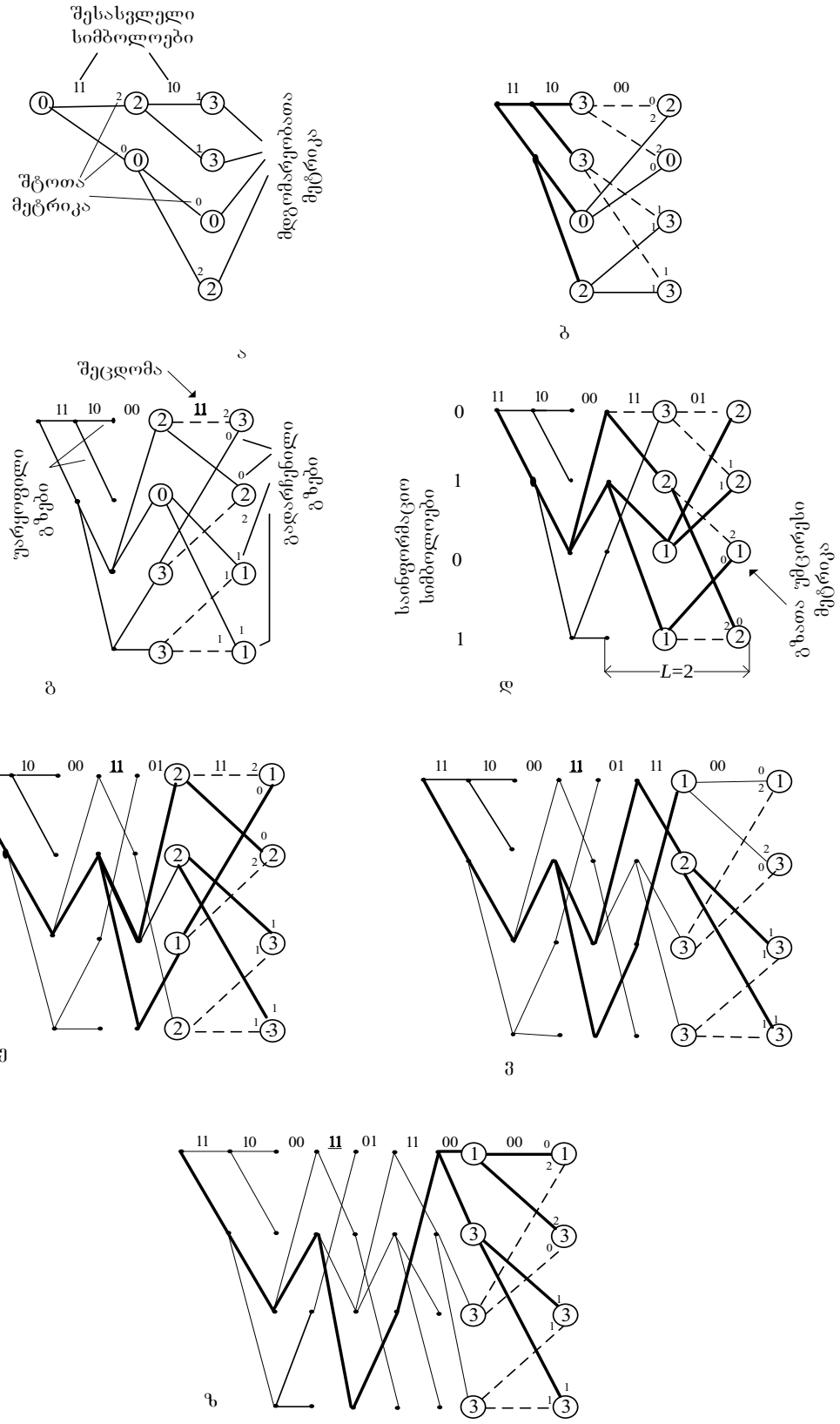
გისოსისებური დიაგრამის პერიოდული სტრუქტურა არსებითად ამარტივებს დეკოდირების პროცესში გზების შედარებას და შერჩევას (4)-ის შესაბამისად. გისოსზე მდგომარეობათა რაოდენობა შეზღუდულია, და შესაბამისად ორ შემთხვევით არჩეულ საკმარისად გრძელ გზას აქვს, როგორც წესი, საერთო მდგომარეობა. მდგომარეობათა შემადგენელი გზის სეგმენტები აუცილებელია შედარდეს ერთმანეთს და აირჩეს ის გზა რომელსაც აქვს ყველაზე ნაკლები მეტრიკა. ასეთ გზას ეწოდება გადარჩენილი. ვიტერბის ალგორითმში გზის სეგმენტების შედარებისა და უარყოფის პროცესი წარმოებს პერიოდულად, დეკოდირების ყოველ ნაბიჯზე. განვიხილოთ (7,5) კოდის დეკოდირების პროცესი, რომლის სიმბოლოებიც გადაიცემა დისკრეტული არხით.

შტოების მეტრიკა განისაზღვრება დეკოდერის შესასვლელზე არსებული $x^{(1)}$ $x^{(2)}$ სიმბოლოთა ერთობლიობასა და კოდის გისოსზე მოცემულ სიმბოლოთა $a^{(1)}$ $a^{(2)}$ ერთობლიობას შორის არსებული ხემინგის მანძილით. თუ $x^{(1)}$ $x^{(2)}$ =1, მაშინ (7,5) კოდისათვის შტოთა მეტრიკის ყველა შესაძლო მნიშვნელობები იქნება: $BM(00)$ – 1, $BM(01)$ – 0, $BM(11)$ – 1, $BM(10)$ – 2. გზის მეტრიკა კი წარმოადგენს გისოსზე არსებულ რამოდენიმე გზისგან შედგენილ შტოთა მეტრიკების ჯამს. მდგომარეობათა მეტრიკა უდრის გზათა იმ მეტრიკას, რომელიც მთავრდება მოცემულ მდგომარეობაზე. დეკოდირების ბიჯი წარმოადგენს დეკოდერის მიერ

არხიდან მიღებული მონაცემის დამუშავების იმ ტაქტების რაოდენობას, რომელიც არსებობს გისოსზე ორ მეზობელ კვანძს შორის.

ქვემოთმოყვანილ ნახ. 11-ზე ნაჩვენებია დეკოდირების პროცესის განვითარება 1/2 სიჩქარის მქონე ხვეულა კოდისათვის, რეგისტრების რაოდენობით $K = 3$ (კოდერი იხ ნახ. 8). კოდერის შესასვლელზე არსებობს სიმბოლოთა შემდეგი წყვილები: 11 10 00 11 01 . . . (განიხილება დეკოდირება ხისტი გადაწყვეტილებით).

საწყის მომენტში კოდერი იმყოფება 00 მდგომარეობაში და შესაბამისი მდგომარეობათა მეტრიკა ტოლი იქნება 0-ის (ნახ. 11 ა). თუ შესასვლელი სიმბოლოები იქნება – 11, მაშინ ამ მდგომარეობიდან გამომავალი შტოების 00 და 11 შესაბამისი მეტრიკა იქნება (00) – 2 და (11) – 0. ეს ნაჩვენებია დეკოდირების პირველ ბიჯზე. ანალოგიური სურათია შემდეგ ეტაპზეც, როდესაც არხიდან მიეწოდება კოდერს წყვილი ბიტი 10 $BM(00) = 1$, $BM(11) = 1$, $BM(10) = 0$ $BM(01) = 2$. ამ ბიჯზე, მდგომარეობათა მეტრიკა განისაზღვრება შემავალი შტოთა მეტრიკისა და წინა მდგომარეობათა მეტრიკის ჯამით $SM(00)=2+1=3$; $SM(10)=2+1=3$; $SM(01)=0+0=0$ და $SM(00)=0+2=2$.



ნახ. 11. ვიტერბის პროცედურა

გისოსისებური დიაგრამის განვითარება ამ პროცესით მთავრდება. შემდეგი ალგორითმია ერთი ძირითადი ნაბიჯის განმეორება. ეს პროცესები დაწვრილებით არის ნაჩვენები დიაგრამაზე (ნახ. 11 ბ-ზ). *i*-ური ბიჯის დასაწყისში დეკოდერის მეხსიერებაში ჩაწერილია წინა ეტაპზე მიღებული მდგომარეობათა მეტრიკა $SM^{i-1}(00)$, $SM^{i-1}(01)$, $SM^{i-1}(10)$, $SM^{i-1}(11)$. არხიდან მიღებული სიმბოლოების მიხედვით ხდება შტოთა მეტრიკის გამოთვლა $BM^{i}(00)$, $BM^{i}(11)$, $BM^{i}(10)$, $BM^{i}(01)$ და ასევე, ოთხი მდგომარეობის $SM^{i}(00)$, $SM^{i}(10)$, $SM^{i}(01)$, $SM^{i}(11)$ ფორმირება შემდეგი წესით: ყოველ შემდეგ მდგომარეობამდე მიდის ორი გზა (მაგალითად 00 მდგომარეობამდე გზა მიდის წინა 00 და 01 მდგომარეობებიდან), დეკოდირების *i*-ურ ნაბიჯზე გზის მეტრიკას დეკოდერი გამოთვლის როგრც წინა მდგომარეობათა მეტრიკისა და შემავალი შტოთა მეტრიკის ჯამს:

$$SM^{i}(00)\begin{cases} PM^{i}(00) = SM^{i-1}(00) + BM^{i}(00) \\ PM^{i}(00) = SM^{i-1}(01) + BM^{i}(11), \end{cases}$$

$$SM^{i}(01)\begin{cases} PM^{i}(01) = SM^{i-1}(10) + BM^{i}(10) \\ PM^{i}(01) = SM^{i-1}(11) + BM^{i}(01), \end{cases}$$

$$SM^{i}(10)\begin{cases} PM^{i}(10) = SM^{i-1}(00) + BM^{i}(11) \\ PM^{i}(10) = SM^{i-1}(01) + BM^{i}(00), \end{cases}$$

$$SM^{i}(11)\begin{cases} PM^{i}(11) = SM^{i-1}(10) + BM^{i}(01) \\ PM^{i}(11) = SM^{i-1}(11) + BM^{i}(10), \end{cases}$$

შემდეგ ხდება თითოეულ მდგომარეობაში შემავალი გზათა მეტრიკის წყვილ-წყვილად შედარება. ხდება უმცირესი მეტრიკის არჩევა, რომელიც ჩაითვლება შესაბამისი მდგომარეობის მეტრიკად და გამოიყენება დეკოდირების შემდეგი ბიჯისათვის. ამ უმცირეს გზას – გადარჩენილ გზას უწოდებენ. ნახ. 11 –ზე გადარჩენილი გზები ნაჩვენებია მუქი ხაზებით. გზები, რომელსაც აქვს უფრო მეტი მეტრიკა ითვლება უარყოფილად და ნახაზზე ნაჩვენებია წყვეტილი ხაზებით.

ვიტერბის ალგორითმში, ასეთი სახით ხდება დეკოდირების ყველა ეტაპზე 3 პროცედურა: 1) წინა მდგომარეობათა მეტრიკისა და შესაბამის შტოთა მეტრიკის შეკრება, 2) შემავალ გზათა მეტრიკის შედარება და 3) უფრო ნაკლები მერიკის მქონე გზის არჩევა, რომლის სიდიდეც შემდეგ გამოიყენება როგორც, მდგომარეობათა მეტრიკა დეკოდირების შემდეგი ეტაპისათვის. თუკი მოხდება ისე რომ, ერთ მდგომარეობაში შემავალ ორ გზას შორის გზათა მეტრიკა იქნება თანაბარი, მაშინ მათ შორის არჩევა ხდება შემთხვევითი გზით.

დეკოდირების თითოეულ ეტაპზე გზის გაგრძელების შესაძლო ვარიანტებიდან ნახევარი უარყოფილია. გზათა დანარჩენი ნახევარი გრძელდება შემდეგ ბიჯამდე, სადაც ჩნდება გზის გაგრძელების შემდეგი 2 ვარიანტი. ასეთნაირად, ყოველ ეტაპზე, დეკოდერში გამოსათვლელი გზების რაოდენობა თანაბარია. დეკოდერში ინახება L-სიმბოლოთა სიგრძის გზები. უფროსი ბიტი, რომელიც არსებობს უფრო ნაკლები მეტრიკის მქონე გზაზე, წარმოადგენს დეკოდერის გამოსასვლელ ბიტს. ზემოთ მოყვანილ ნახაზზე ნაჩვენებია დეკოდირების შემდეგი ეტაპები (ნახ. 11 ბ-ზ). განვიხილოთ ეს ეტაპები დეტალურად. მესამე ეტაპზე წყვეტილი ხაზით ნაჩვენებია უარყოფილი გზები, რომლებიც შემდეგ ეტაპზე აღარ გრძელდება. უარყოფილი გზების რაოდენობა გისოსსზე იზრდება, ხოლო გადარჩენილი გზების რაოდენობა მუდმივია (მოცემულ შემთხვევაში 4). დავუშვათ გადაცემის არხში მოხდა შეცდომა ერთ სიმბოლოში (ნახ. 11 გ). ეს შეცდომა გასწორებული იქნება თუ დეკოდირების შემდეგ ეტაპზე არჩეული იქნება უმცირესი მეტრიკის მქონე გზა. მეხუთე ეტაპზე (ნახ. 11 დ) უმცირესი მეტრიკის მქონე გზა გადის მდგომარებას $S = 11$, რაც შეესაბამება გადასაცემ საინფორმაციო მიმდევრობას $u = 10110\ldots$. კოდერის მდგომარეობები, რომლებზედაც გადის შერჩეული გზები, განსაზღვრავს სწორედ საინფორმაციო მიმდევრობის სიმბოლოებს. გისოსზე ჩანს, რომ გარკვეულ მანძილზე გადარჩენილი გზები ბოლოში ერთიანდება (ნახ. 11 ზ).

1.4 დასკვნები

მოცემულ თავში განხილული იყო კლასიკიური ხვეულა კოდები, რომლის წარმოდგენისათვის გამოყენებულია სასრული ავტომატის მოდელი, რომელშიც საინფორმაციო და კოდური მიმდევრობები წარმოდგებიან დაყოვნების ოპერატორებით. იგივე იქნება გამოყენებული ჩვენს მიერ შემდგომში ახალი კოდების აგებისას. ამასთან გამოყენებული იქნება იგივე აღწერის მეთოდები და კოდთა პარამეტრები.

ჩვენ, სპეციალურად დაწვრილებით განვიხილეთ დეკოდირების მაქსიმალური დამაჯერებლობის პრინციპი (რომელიც რეალიზებულია ვიტერბის ალგორითმით), რადგანაც იგივე პრინციპით იქნება დეკოდირებული ჩვენს მიერ აგებული ახალი კოდები, რომლებიც ეფუძნებიან ხვეულა კოდებს ალფაბეტური სიჭარბით.

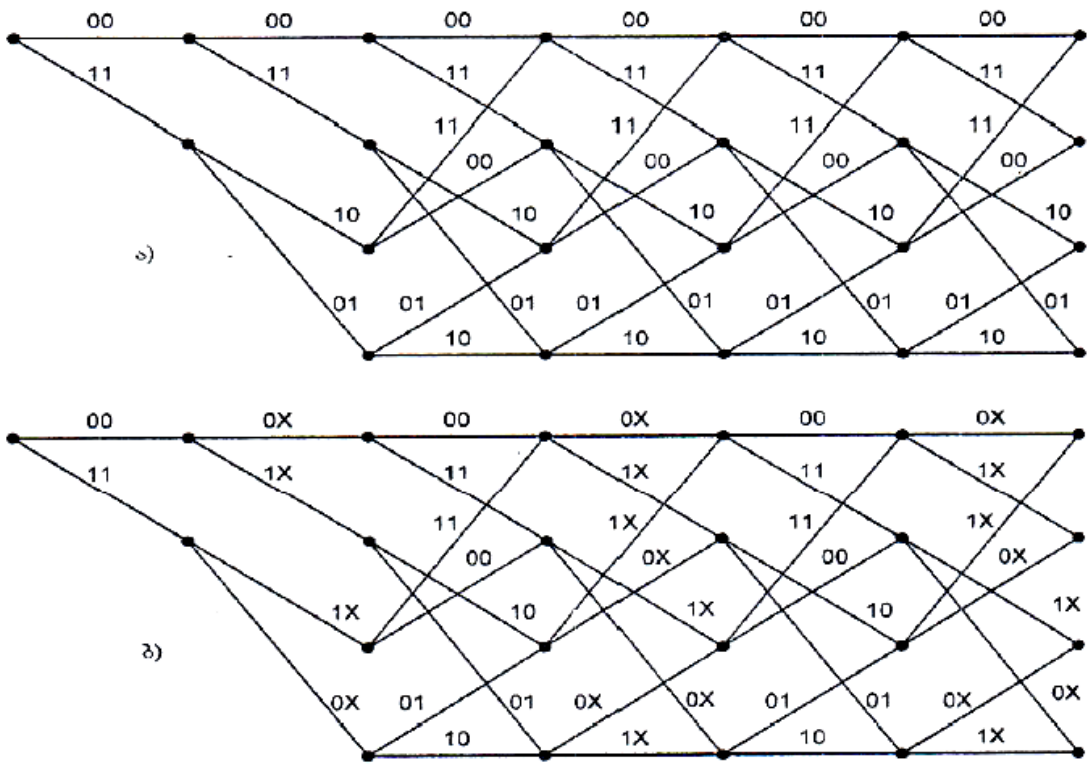
თავი II

ხვეულა კოდები ალფაბეტური სიჭარბით

2.1 კოდები სიმბოლურ-ალფაბეტური სიჭარბით

კოდთა თანამედროვე კლასიფიკაციით არსებობს მრავალი ჯგუფი, რომლებშიც რაღაც ნიშნით გაერთიანებულია გარკვეული რაოდენობის სიჭარბის კოდი და ყოველი ცალკეული ჯგუფისათვის გამოყენებულია კვლევის სპეციფიკური და კონკრეტული მიდგომა. ვფიქრობთ, შეიძლება შეიქმნას ისეთი სისტემა, რომელიც განიხილავს ერთ ზოგად კოდს, ხოლო ყველა დანარჩენი მისი კონკრეტული რეალიზაციები იქნება. ასეთ კოდს ვუწოდებთ ჩვენ კოდს სიმბოლურ-ალფაბეტური სიჭარბით და ვვარაუდობთ რომ კვლევები აღნიშნული კუთხით მნიშვნელოვან შედეგებამდე მიგვიყვანს. ამის დადასტურების ერთ კერძო შემთხვევას ეძღვნება წინამდებარე პარაგრაფი, სადაც ნაჩვენებია, რომ ზოგადად პერფორირებული ხვეულა კოდი [13] არის სიმბოლურ-ალფაბეტური სიჭარბის მქონე კოდის კონკრეტული რეალიზაცია.

ხვეულა კოდების სიჩქარის გაზრდა შეიძლება კოდური სიმბოლოების ამოგდებით წინასწარ განსაზღვრული წესით. მიღებულ კოდებს, პერფორირებულ კოდებს უწოდებენ [13] მაგალითად (იხ. ნახ. 12), თუ ავიღებთ R=1/2 სიჩქარიან ხვეულა კოდს და კოდური მიმდევრობიდან ყოველ მეოთხე სიმბოლოს ამოვაგდებთ, მივიღებთ პეფორირებულ კოდს, რომლის სიჩქარეა R=2/3. ამ შემთხვევაში კოდს ექნება იგივე სირთულე, რაც 1/2 სიჩქარიან კოდს, მაშინ როცა ჩვეულებრივი 2/3 სიჩქარიანი კოდი [14] გაცილებით რთულია. ამ შემთხვევაში ვგულისხმობთ კოდის შესაბამისი გისოსისსირთულეს, რომელზეც ხორციელდება ოპტიმალური დეკოდირების პროცედურა ვიტერბის ალგორითმის გამოყენებით (იხ. პარაგრაფი 1.3).

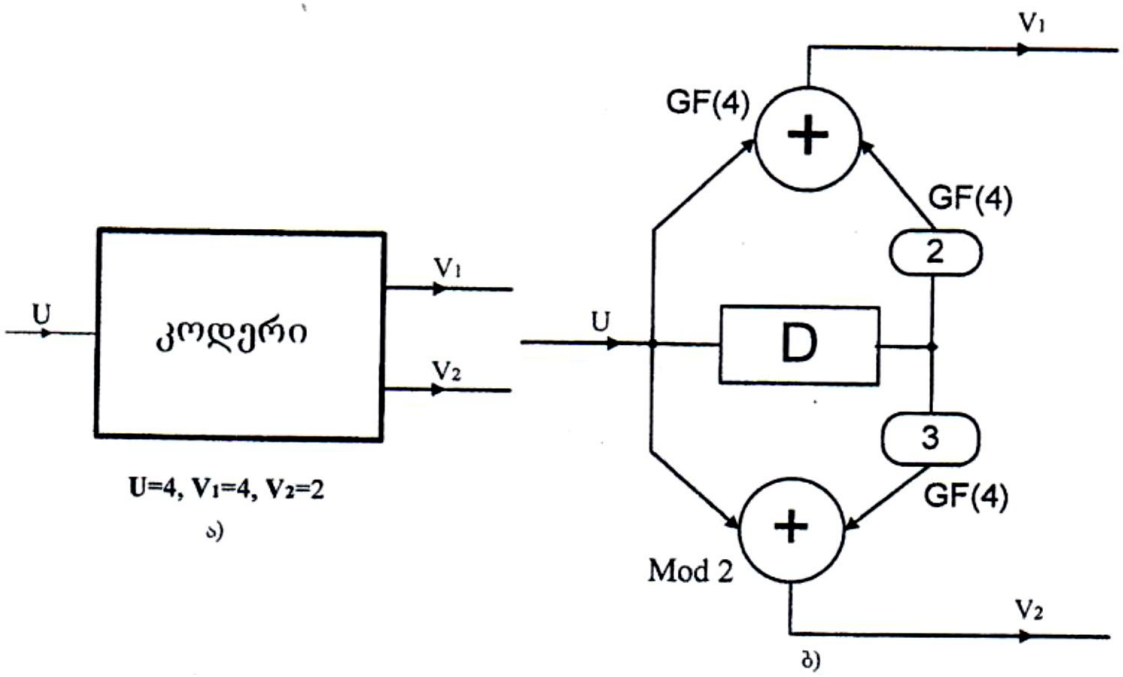


ნახ. 12. ხვეულა (7,5) კოდის გისოსის ფრაგმენტი 1/2 სიჩქარისათვის (ა) და იგივე, სიმბოლოთა ამოგდების გზით მიღებული, 2/3 სიჩქარიანი კოდისათვის (ბ)

M-ობითი ხვეულა კოდის სიჩქარე ზოგადად $R=k/n$. შესაბამისი გისოსის კვანძების რიცხვი $S=M^{v}$, სადაც ν არის კოდერში მეხსიერების ელემენტების რაოდენობა, ხოლო თითოეულ კვანძში შემავალი შტოების რიცხვი $B=M^{k}$. დეკოდირების სირთულე ფაქტიურად განისაზღვრება იმ არითმეტიკული ოპერაციების რაოდენობით რომლებიც გამოყენებულია დეკოდირების პროცესში. ცხადია, ამ შემთხვევაში, ერთი დეკოდირებული ბიტისათვის $N=(\frac{M^{v}}{k})$. $(M^{k+1}-1)$ ვინაიდან პერფორირებული კოდების შემთხვევაში $K=1$ სიმარტივის თვალსაზრისით, ჩვეულებრივ მაღალსიჩქარიან ხვეულა კოდებთან შედარებით, მათი უპირატესობა აშკარაა.

სიმბოლურ-ალფაბეტური სიჭარბის კოდის კოდერებს აქვს *k* შესასვლელი და *n* გამოსასვლელი. ამ დროს, ტრადიციული შემთხვევისაგან განსხვავებით, არაა აუცილებელი $n>k$; ამასთან,

შესასვლელი და გამოსასვლელი სიმბოლოების ალფაბეტის ზომები შეიძლება იყოს სხვადასხვა და უფრო მეტიც, სხვადასხვა შეიძლება იყოს როგორც ცალკეულ შესასვლელზე, ისე ცალკეულ გამოსასვლელებზე არსებული სიმბოლოების ალფაბეტის ზომები.

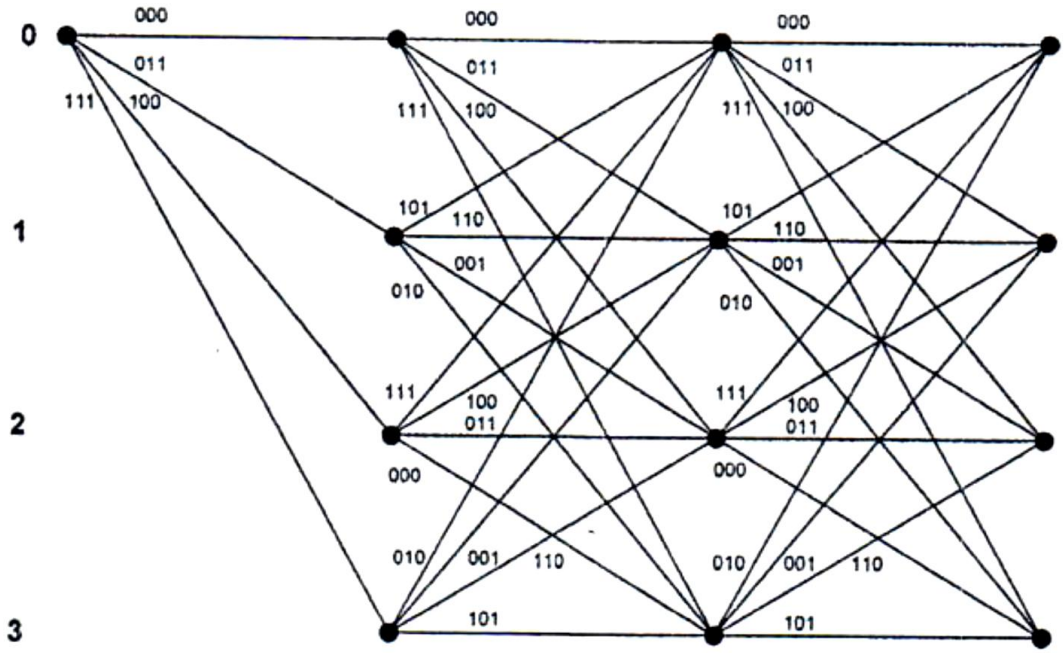


ნახ. 13. პერფორირებული კოდის ანალოგი, კოდი სიმბოლურ-ალფაბეტური სიჭარბით; ა) კოდერის ზოგადი სქემა, ბ) კონკრეტული კოდერი

ნახ. 13 ა-ზე მოყვანილია სიმბოლურ-ალფაბეტური სიჭარბის კოდის ზოგადი მაგალითი. აქ კოდერის შესასვლელზე არსებული სიმბოლოების ალფაბეტის ზომა U=4, ე.ი. კოდერის შესასვლელზე მიეწოდება ოთხობითი სიმბოლოები. ასევე, კოდერის პირველ გამოსასვლელზე არსებული სიმბოლოების ალფაბეტის ზომა V_1=4, ე.ი კოდერის პირველ გამოსასვლელზეც გვაქვს ოთხობითი სიმბოლოები. კოდერის მეორე გამოსასვლელზე არსებული სიმბოლოებისალფაბეტის ზომა V_2=2, ე.ი კოდერის მეორე გამოსასვლელზე გვაქვს ორობითი სიმბოლოები.

თუ გამოსასვლელ ოთხობით სიმბოლოებს წარმოვადგენთ ორობითების მეშვეობით ($0\rightarrow 00,\ 1\rightarrow 01,\ 2\rightarrow 10,\ 3\rightarrow 11$) და დავუშვათ რომ არსებობდა ჰიპოთეზური კოდი, რომლისთვისაც V_2=4, და მისი კოდერის

მეორე გამოსასვლელზე არსებული ორობითი კოდური მიმდევრობიდან ამოვაგდებთ ყოველ მეოთხე სიმბოლოს (ანალოგიურად 1-ელ ნახ-ზე მოყვანილი შემთხვევისა), მივიღებთ პერფორირებულ კოდს, რომლის შესაბამისი კოდერის რეალიზაციები ნაჩვენებია მე-2 ა, ბ ნახ-ზე, ხოლო მისი გისოსის ფრაგმენტი მე-14 ნახ-ზე. ცხადია, მოცემული კოდის სიჩქარე R=2/3, კოდის თავისუფალი მანძილი ჰემინგის მეტრიკაში d_f=4 ხოლო ბიჯზე დეკოდირების ორობითი სიმბოლოების რაოდენობა ჩვენი შემთხვევისათვის S=4, B=4, n=3, k=1, M=4 და მაშინ $2N$=76; იგივეა d_f-ის მქონე პერფორირებული კოდისთვის $2N$=96.



ნახ. 14. სიმბოლურ-ალფაბეტური სიჭარბის კონკრეტული კოდის გისოსი

მოყვანილი კონკრეტული მაგალითიდან გამომდინარე, სიმბოლურ-ალფაბეტური სიჭარბის კოდები იმსახურებს ინტერესს და აუცილებელია გაგრძელდეს კვლევები აღნიშნული მიმართულებით როგორც თეორიული, ისე პრაქტიკული კუთხით. ამ მიმართულებით ჩვენ გამოვიკვლევთ აღნიშნულ კოდებს შემთხვევებისათვის K=1, n=1, q>2, Q>2, Q>2. ე.ი. განხილულ კოდებს ექნებათ ერთი q-ობითი შესასვლელი და ერთი Q-ობითი გამოსასვლელი.

2.2 კოდების პოვნის მეთოდები. დეიქსტრის ალგორითმი

ვინაიდან არ არსებობს ხვეულა კოდების აგების რაიმე რეგულარული ალგორითმი, ამიტომ ჩვენ კოდებს მოვიპოვებთ კომპიუტერული ძიების გზით და ამ დროს მნიშვნელოვანია შევარჩიოთ მისი თავისუფალი მანძილის განსაზღვრის ალგორითმი. გავაკეთოთ არჩევანი დეიქსტრის ალგორითმზე [15, 16] მისი სიმარტივის გამო. ამ დროს აუცილებელია გვქონდეს კოდის აღმწერი აწონილი ორიენტირებული გრაფი, რისი აგების მაგალითები მოცემული იყო თავი I -ში. დეიქსტრის ალგორითმი მუშაობს შემდეგნაირად:

ვთქვათ გვინდა განსვსაზღვროთ უმოკლესი მანძილი გრაფის s წვეროდან f წვერომდე.

ნაბიჯი 1. პროცედურის დაწყებამდე გრაფის არცერთი წვერო და რკალი არ არის შეღებილი. ყველა x წვეროს მიენიჭება წონა რომელიც ტოლია მანძილისა sდან x-მდე.

ვუშვებთ, რომ $w(s)=0$ და $w(x)=\infty$.

ნაბიჯი 2. ყველა x შეუღებავი წვეროსთვის განვსაზღვროთ წონა $w(x)$:

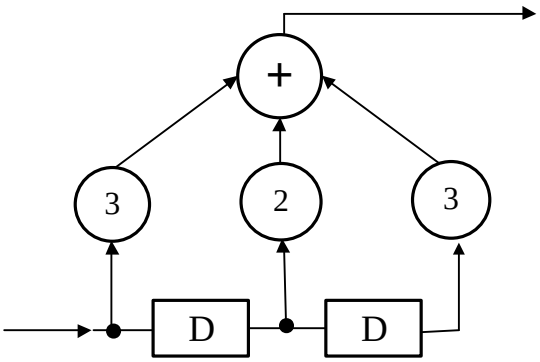
$$w(x)=\min\{w(x), w(y)+d(y,x)\},$$

სადაც $d(y,x)$ ტოლია მანძილისა yდაxწვეროს შორის.

თუ ყველა შეუღებავი x-ვის $d(x)=\infty$, ალორითმი მთავრდება, წინააღმდეგ შემთხვევაში შევღებოთ ის x წვერო, რომლისთვისაც $d(x)$ მინიმალურია. ამავე დროს იღებება შესაბამისი რკალიც და ვუშვებთ, რომ $y=x$.

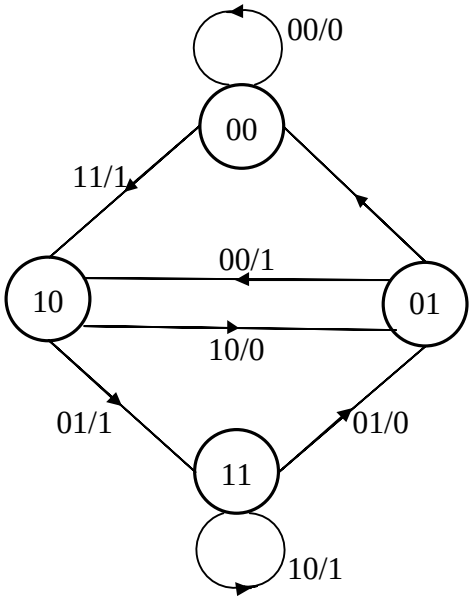
ნაბიჯი 3. თუ $y=f$ ალგორითმი მთავრდება და ნაპოვნია უმოკლესი მანძილი s-დან f-მდე. წინაღმდეგ შემთხვევაში დავუბრუნდეთ ნაბიჯ 2-ს.

განვიხილოთ კონკრეტული მაგალითი. ვთქვათ გვაქვს ალფაბეტური სიჭარბის მქონე კოდი (323). ვუშვებთ, რომ კოდი აგებულია GF(4)-ზე და გვაქვს ასახვა 00→0; 01→1; 10→ 2; 11→3. კოდერის სქემა მოცემულია ქვემოთ ნახ. 15-ზე.



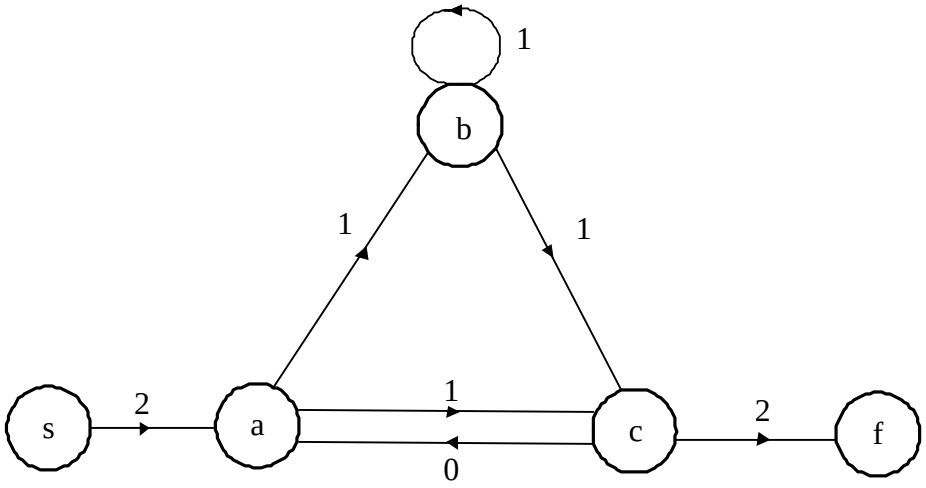
ნახ. 15. (323) ხვეულა კოდის კოდერი.

აქ შესასვლელი ალფაბეტის ზომაა q=2, ხოლო გამოსასვლელისა Q=4; მოცმული კოდერის გრაფი მოყვანილია ნახ. 16-ზე. თუ ცნობილი წესებით მოვახდენთ მის მოდიფიცირებას [17,18],



ნახ. 16. (323) ხვეულა კოდის აღმწერი გრაფი

მივიღებთ გრაფს, რომელიც მოყვანილია ნახ. 17-ზე.



ნახ. 17. (323) ხვეულა კოდის აღმწერი მოდიფიცირებული გრაფი

გამოვიყენოთ ზემოთ მოყვანილი დეიქსტრის ალგორითმი და ვიპოვოთ უმოკლესი მანძილი s წვეროდან f წვერომდე, ანუ (323) კოდის თავისუფალი მანძილი ჰემინგის მიხედვით.

$w(s)=0;\ w(a)=\infty;\ w(b)=\infty;\ w(c)=\infty;\ w(f)=\infty$

მანძილი გადასვლაზე :

$d(s,a)=2;\ d(a,b)=1;\ d(a,c)=1;\ d(b,c)=1;\ d(c,a)=0;\ d(c,f)=2$. ყველა სხვა $d(i,j)=\infty$.

ნაბიჯი 1.

$$w(a)=\min\left[w(a),w(s)+d(s,a)\right]=\min\left[\infty,0+2\right]=2;$$

$$w(b)=\min\left[w(b),w(s)+d(s,b)\right]=\min\left[\infty,0+\infty\right]=\infty;$$

$$w(c)=\min\left[w(c),w(s)+d(s,c)\right]=\min\left[\infty,0+\infty\right]=\infty;$$

$$w(f)=\min\left[w(f),w(s)+d(s,f)\right]=\min\left[\infty,0+\infty\right]=\infty;$$

$\min=2;$

$w(f)=\min$? არა, შეიღებოს $w(a)$, და

ნაბიჯი 2.

$$w(b) = \min\left[w(b), w(a) + d(a,b)\right] = \min\left[\infty, 2+1\right] = 3;$$

$$w(c) = \min\left[w(c), w(a) + d(a,c)\right] = \min\left[\infty, 2+1\right] = 3;$$

$$w(f) = \min\left[w(f), w(a) + d(a,f)\right] = \min\left[\infty, 2+\infty\right] = \infty;$$

$\min = 3$;

$w(f) = \min$? არა, შეიღებოს, მაგალითად, $w(c)$, და

ნაბიჯი 3.

$$w(b) = \min\left[w(b), w(c) + d(c,b)\right] = \min\left[3, 3+\infty\right] = 3;$$

$$w(f) = \min\left[w(f), w(c) + d(c,f)\right] = \min\left[\infty, 3+2\right] = 5;$$

$\min = 3$;

$w(f) = \min$? არა, შეიღებოს $w(b)$ და

ნაბიჯი 4.

$$w(f) = \min\left[w(f), w(b) + d(b,f)\right] = \min\left[5, 3+\infty\right] = 5.$$

$\min = 5$;

$w(f) = \min$? კი (ის ერთია მხოლოდ), პროცედურა დამთავრდეს

$$\min\left[d(s,f)\right] = w(f) = 5.$$

ე.ი. კოდის თავისუფალი მანძილია $dx = 5$.

2.3 კოდთა მანძილთა სპექტრი

რამდენიმე ერთნაირი თავისუფალი მანძილის მქონე კოდიდან საუკეთესოს ამორჩევა ხორციელდება კოდის მანძილთა სპექტრის მიხედვით. განვიხილოთ მისი განსაზღვრის ალგორითმი. ის დაიყვანება კოდის წარმომქმნელი ფუნქციის განსაზღვრის ალგორითმზე და ეფუძნება n x n ზომის მომიჯნავეობის M_N მატრიცის გამოყენებას, რომლის

ელემენტები წარმოდგენილნი არიან ფორმალური $D^{\omega}N^{u}$ ცვლადების სახით, სადაც ω არის გრაფის რკალის რიცხვითი წონა, ხოლო u შესაბამისი საინფორმაციო არანულოვანი სიმბოლოთა რაოდენობა.

განვიხილოთ სტრიქონული მატრიცა $M_i = (m_{i,1},m_{i,2},\ldots,m_{i,n})$. ყოველი შემდეგი M_{i+1} მატრიცა განისაზღვება შემდეგნაირად: $M_{i+1} = M_iM_N$. პირველ ნაბიჯზე M_1 მატრიცის სახით აიღება M_N–ის პირველი სტრიქონი. კოდის წარმომქმნელი ფუნქცია $T(D,N)$ შედგება $\{M\}$ ვექტორთა პირველი $m_{i,1}$ კომპონენტების ერთობლიობისგან, ე.ი. იმ ელემენტებისგან, რომლებიც მიიღებიან ყველა მოცემული M_i ვექტორის M_N მატრიცის პირველ სვეტზე გამრავლებით.

იმისათვის, რომ გამოვრიცხოთ განხილვიდან საწყის კვანძში დამთავრებული გზები, საკმარისია მატრიცების ყოველი გამრავლების ოპერაციის წინ M_N-ის პირველი სტრიქონი გავანულოთ. თუ გვაინტერესებს მანძილთა სპექტრი რომელიღაც $D_{max}(,)$ მნიშვნელობამდე, ყოველი გამრავლების წინ M_i სტრიქონული მატრიციდან უნდა გამოირიცხოს ის წევრები, რომელთა შესაბამისი D ცვლადის ხარისხი მეტია $D_{max}(,)$-ზე. პროცედურა მთავრდება მაშინ, როცა M_i-ში არ რჩება არცერთი არანულოვანი ელემენტი.

წარმომქნელი ფუნქციიდან კოდის მანძილთა სპექტრის განსაზღვრის წესი მოყვანილია [19] – ში. განვიხილოთ ერთი მაგალითი.

წინა პარაგრაფში განხილული (323) ხვეულა კოდის მომიჯნავეობის მატრიცას აქვს სახე

$$M_N = \begin{bmatrix} 0 & D^2N & 0 & 0 \\ 0 & 0 & D & DN \\ D^2 & N & 0 & 0 \\ 0 & 0 & D & DN \end{bmatrix}$$

ავიღოთ $D_{max}(,) = 11$

ნაბიჯი 1.

$M_{1,1}=0;$

$M_{1,2}=D^2N;$

$M_{1,3}=0;$

$M_{1,4}=0.$

ნაბიჯი 2.

$M_{2,1}=0;$

$M_{2,2}=0;$

$M_{2,3}=D^3N;$

$M_{2,4}=D^3N^2.$

ნაბიჯი 3.

$M_{3,1}= D^5N;$

$M_{3,2}= D^3N^2;$

$M_{3,3}= D^4N^2;$

$M_{3,4}= D^4N^3.$

ნაბიჯი 4.

$M_{4,1}= D^6N^2;$

$M_{4,2}= D^4N^3;$

$M_{4,3}= D^4N^2+ D^5N^3;$

$M_{4,4}= D^4N^3+ D^5N^4.$

ნაბიჯი 5.

$M_{5,1}= D^6N^2+ D^7N^9;$

$M_{5,2}= D^4N^3+ D^5N^4;$

$M_{5,3}= 2D^5N^3+ D^6N^4;$

$M_{5,4}= 2D^5N^4+ D^6N^5.$

ნაბიჯი 6.

$M_{6,1}= 2D^7N^3+ D^8N^4;$

$M_{6,2}= 2D^5N^4+ D^6N^5;$

$M_{6,3}= D^5N^3+ D^6N^4+ D^7N^5;$

$M_{6,4}= D^5N^4+ D^6N^5+ D^7N^6 .$

ნაბიჯი 7.

$M_{7,1}= D^7N^3+ D^8N^4+ D^9N^5;$

$M_{7,2}= D^5N^4+ D^6N^5+ D^7N^6;$

$M_{7,3}= 3D^6N^4+ D^7N^5+ D^8N^6;$

$M_{7,4}= 3D^6N^5+ D^7N^6+ D^8N^7.$

ნაბიჯი 8.

$M_{8,1}= 3D^8N^4+ D^9N^5+ D^{10}N^6;$

$M_{8,2}= 3D^6N^5+ D^7N^6+ D^8N^7;$

$M_{8,3}= D^6N^4+ D^7N^5+ D^8N^6+ D^9N^7;$

$M_{8,4}= D^6N^5+ D^7N^6+ D^8N^7+ D^9N^8.$

ნაბიჯი 9.

$M_{9,1}= D^8N^4+ D^9N^5+ D^{10}N^6 + D^{11}N^7;$

$M_{9,2}= D^6N^5+ D^7N^6+ D^8N^7 + D^9N^8;$

$M_{9,3}= 4D^7N^5+ D^8N^6+ D^9N^7+ D^{10}N^8;$

$M_{9,4}= 4D^7N^6+ D^8N^7+ D^9N^8+ D^{10}N^9.$

ნაბიჯი 10.

$M_{10,1}= 4D^9N^5+ D^{10}N^6+ D^{11}N^7;$

$M_{10,2}= 4D^7N^6+ D^8N^7+ D^9N^8 + D^{10}N^9;$

$M_{10,3}= D^7N^5+ D^8N^6+ D^9N^7+ D^{10}N^8 +D^{11}N^9;$

$M_{10,4}= D^7N^6+ D^8N^7+ D^9N^8+ D^{10}N^9 +D^{11}N^{10}.$

ნაბიჯი 11.

$M_{11,1}= D^9N^5+ 10D^{10}N^6+ 15D^{11}N^7;$

$M_{11,2}= D^7N^6+ 10D^8N^7+ 15D^9N^8 + 7D^{10}N^9 +D^{11}N^{10};$

$M_{11,3}= 5D^8N^6+ 20D^9N^7+ 21D^{10}N^8 +8D^{11}N^9;$

$M_{11,4}= 5\ D^8N^7+ 20D^9N^8+ 21D^{10}N^9 +8D^{11}N^{10}.$

ნაბიჯი 12.

$M_{12,1}=5D^{10}N^6+ 20D^{11}N^7;$

$M_{12,2}=5D^8N^7+ 20D^9N^8 + 21D^{10}N^9 +D^{11}N^{10};$

$M_{12,3}= D^8N^6+ 15D^9N^7+ 35D^{10}N^8 +28D^{11}N^9;$

$M_{12,4}=\ D^8N^7+ 15D^9N^8+ 35D^{10}N^9 +28D^{11}N^{10}.$

ნაბიჯი 13.

$M_{13,1}=5D^{10}N^6+ 15D^{11}N^7;$

$M_{13,2}=D^8N^7+ 15D^9N^8 + 35D^{10}N^9 +28D^{11}N^{10};$

$M_{13,3}= 6D^9N^7+ 35D^{10}N^8 +56D^{11}N^9;$

$M_{13,4}= 6D^9N^8+ 35D^{10}N^9 +28D^{11}N^{10}0.$

ნაბიჯი 14.

$M_{14,1}= 6D^{11}N^7;$

$M_{14,2}= 6D^9N^8 + 35D^{10}N^9 +56D^{11}N^{10};$

$M_{14,3}= D^9N^7+ 21D^{10}N^8 +70D^{11}N^9;$

$M_{14,4}= D^9N^8+ 21D^{10}N^9 +70D^{11}N^{10}.$

ნაბიჯი 15.

$M_{15,1}= D^{11}N^7;$

$M_{15,2}= D^9N^8 + 21D^{10}N^9 +70D^{11}N^{10};$

$M_{15,3}= 7D^{10}N^8 +56D^{11}N^9;$

$M_{15,4}= 7D^{10}N^9 +56D^{11}N^{10}.$

ნაბიჯი 16.

$M_{16,1}= 0;$

$M_{16,2}= 7D^{10}N^9 +56D^{11}N^{10};$

$M_{16,3}= D^{10}N^8 +28D^{11}N^9;$

$M_{16,4}= D^{10}N^9 +28D^{11}N^{10}.$

ნაბიჯი 17.

$M_{17,1}= 0;$

$M_{17,2}= D^{10}N^9 +28D^{11}N^{10};$

$M_{17,3}= 8D^{11}N^9;$

$M_{17,4}= 8D^{11}N^{10}$.

ნაბიჯი 18.

$M_{18,1}= 0$;

$M_{18,2}= 8D^{11}N^{10}$;

$M_{18,3}= D^{11}N^{9}$;

$M_{18,4}= D^{11}N^{10}$.

ნაბიჯი 19.

$M_{19,1}= 0$;

$M_{19,2}= D^{11}N^{10}$;

$M_{19,3}= 0$;

$M_{19,4}= 0$.

ნაბიჯი 20.

$M_{20,1}= 0$;

$M_{20,2}= 0$;

$M_{20,3}= 0$;

$M_{20,4}= 0$.

პროცედურა დამთავრებულია და კოდის წარმომქმნელი ფუნქცია ტოლია:

$$T(D,N)=\sum_{i=1}^{20} m_{i,1} = D^5N + 2D^6N^2 + 4D^7N^3 + 8D^8N^4 + 16D^9N^5 + 32D^{10}N^6 + 64D^{11}N^7$$

სადაც $d_H(,)$ მოცემული გზების წონაა, ხოლო a_i ამ გზების შესაბამისი გადმოცემული არანულოვან საინფორმაციო სიმბოლოთა საერთო რაოდენობა; მოყვანილი გამოსახულებიდან (323) კოდის მანძილთა სპექტრისათვის გვაქვს: 5-1, 6-4, 7-12, 8-32, 9-80 . . . , სადაც პირველი

ციფრი ჰემინგის მანძილია, ხოლო მეორე მოცემული მანძილის მქონე გზების შესაბამისი გადაცემული ბიტების რაოდენობა.

2.4 დასკვნები

განხილულია კოდები სიმბოლურ-ალფაბეტური სიჭარბით. მოყვანილია მისი აღწერა და დადებითი მხარეები. გადაწყვეტილია, რომ საუკეთესო კოდები შეირჩეს კომპიუტერული ძიების მეთოდით და ამ დროს თავისუფალი მანძილის განსაზღვრისათვის გამოყენებული იქნას დეიქსტრის ალგორითმი. კონკრეტული კოდისათვის განხილულია ალგორითმის მუშაობის პროცედურა. განსაზღვრულია, რომ ერთნაირი თავისუფალი მანძილის მქონე ხვეულა კოდებიდან საუკეთესოს შერჩევა განხორციელდეს მისი მანძილთა სპექტრის მიხედვით. აღწერილია სპექტრის განსაზღვრის ალგორითმი და მოყვანილია საილუსტრაციო მაგალითი.

თავი III

ახალი ხვეულა კოდები ალფაბეტური სიჭარბით

3.1 ლის მეტრიკის გამოყენება ინვარიანტული სისტემების ასაგებად

კოდირების თეორიაში უმეტესწილად განიხილება შემთხვევები, როცა კოდურ ვექტორთა შორის მანძილები განისაზღვრება ჰემინგის მეტრიკაში [20-22]. ამასთან ერთად იშვიათად, მაგრამ მაინც გვხვდება სისტემები, რომლებიც იყენებენ ე.წ. რგოლურ ანუ ლის მეტრიკას [20, 23, 24]; ცნობილია ისიც, რომ ორობით და სამობით შემთხვევაში ეს მეტრიკები იდენტურია ანუ ერთიმეორეს ემთხვევა [20].

ბოლო პერიოდში, ალფაბეტური სიჭარბის მქონე კოდების გამოჩენა [25, 26] აძლიერებს ინტერესს ისეთი სისტემებისადმი, რომლებიც წარმოადგენენ წრფივი არაორობითი და სხვადასხვა (მათ შორის არაწრფივი) კოდის კოდერების გაერთიანებას, კასკადს. სწორედ ასეთი სტრუქტურის ერთი თვისებრივი ასპექტი განხილულია წინამდებარე პარაგრაფში, სადაც საბაზო კოდის სახით გამოიყენება წრფივი ხვეულა კოდი ალფაბეტური სიჭარბით.

ვსაუბრობთ რა ამა თუ იმ მეტრიკის შესახებ, მხედველობაში გვაქვს ორ v_i-სა და v_j კოდურ ვექტორს (სიტყვას) შორის $d(v_i v_j)$ მანძილის გაზომვის წესი. ცხადია, ამ დროს მანძილი აკმაყოფილებს სამ, ქვემოთ მოყვანილ, პირობას [27]:

1. $d(v_i, v_j) = 0$; თუ $i=j$, (იგივეობის აქსიომა);
2. $d(v_i, v_j) = d(v_j, v_i)$, (სიმეტრიის აქსიომა);
3. $d(v_i, v_j) \le d(v_i, v_c) + d(v_c, v_j)$, (სამკუთხედის აქსიომა)

და კოდირების თეორიაში ის განისაზღვრება, ძირითადად, როგორც ადიტიური ზომა

$$d(v_i, v_j) = \sum_{k=1}^{n} f(v_{ki}, v_{kj})$$

გამოსახულების შესაბამისად; სადაც n კოდური სიტყვის სიგრძეა, ხოლო v_{ki} და v_{kj} შესაბამისად v_i და v_j ვექტორების k-ური კომპონენტები. ამ დროს ჰემინგის მეტრიკისათვის $f(v_{ki},v_{ki})$ ფუნქცია, რომელიც განსაზღვრავს კომპონენტთა განსხვავებულობას

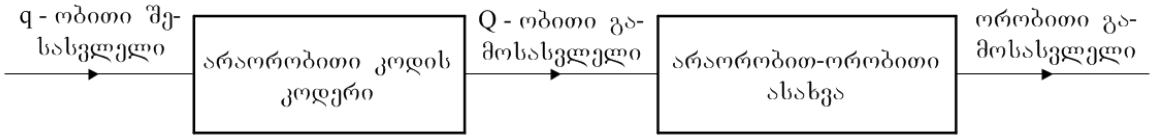
$$f_H(v_{ki}, v_{kj}) = \begin{cases} 0, & \text{თუ } v_{ki} = v_{kj}; \\ 1, & \text{თუ } v_{ki} \neq v_{kj}; \end{cases} \tag{5}$$

ხოლო Q-ობითი კოდის შემთხვევაში ლის მეტრიკისათვის გვაქვს:

$$f(v_{ki}, v_{kj}) = \min\{|v_{ki} - v_{kj}|, Q - |v_{ki} - v_{kj}|\} \tag{6}$$

და როგორც ვიცით, ეს მეტრიკა იდენტურია ჰემინგის მეტრიკისა თუ Q=2 და Q=3.

დასაწყისში, მოცემულ პარაგრაფში, განვიხილავთ ნახ. 18-ზე მოყვანილ სისტემას.

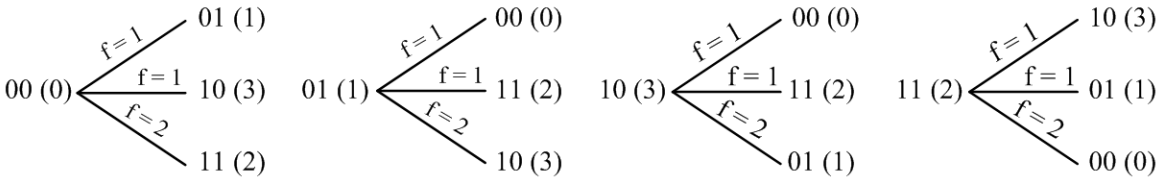


ნახ. 18. არაორობით-ორობითი სისტემა

აქ $q \geq 2$, $Q > 2$ და $Q > q$; მოცემულ შემთხვევაში Q-ობით გამოსასვლელზე არსებულ წრფივ კოდს განვიხილავთ ლის მეტრიკაში, ხოლო ორობით გამოსასვლელზე არსებულ n ბიტი სიგრძის სიტყვებისაგან შედგენილ კოდს ჰემინგის მეტრიკაში. ამასთან ვთვლით, რომ Q-ობით გამოსასვლელზე სიმბოლოთა ალფაბეტის ზომა და ორობით გამოსასვლელზე კოდური სიტყვების რაოდენობა ერთნაირია. ზოგადად ნახ. 18-ზე მოყვანილი სისტემა შეიძლება იყოს არაწრფივი, რაც ქმნის

მნიშვნელოვან სიძნელეებს მათი აგებისა და ანალიზის პროცესში. ცხადია ეს პრობლემა გარკვეულწილად გადაიჭრება, თუ სისტემა იქნება ინვარიანტული მანძილის მიმართ. განვიხილოთ ასეთი მაგალითი: ვთქვათ, ვიმყოფებით ნულოვან ვექტორთან და ვხედავთ აქედან d მანძილებით დაშორებულ v კოდურ სიტყვებს. თუ სურათი იგივე იქნება მოცემული კოდის ნებისმიერი კოდური სიტყვისთვის, მაშინ ვიტყვით, რომ კოდი ინვარიანტულია მანძილის მიმართ [21]; ე.ი. გვაქვს კოდი, რომლისთვისაც რაოდენობა იმ ვექტორებისა, რომლებიც დაშორებულია რომელიღაც v ვექტორიდან რაღაც d მანძილით არაა დამოკიდებული ამ v ვექტორის შერჩევაზე. ცხადია, ყველა წრფივი კოდი ინვარიანტულია მანძილის მიმართ [21].

ვთვლით, რომ ნახ. 18-ზე წარმოდგენილი ე.წ. არაორობით-ორობითი ტიპის კოდი (ან შემდგომში მის ადგილზე მოყვანილი სხვა კოდიც) მინიმუმ ინვარიანტულია მანძილის მიმართ; მაშინ, თუ ორობით გამოსასვლელზე არსებული კოდური სიტყვებისათვის ჰემინგის d_H მანძილები ტოლი არის Q-ობითი გამოსასვლელზე არსებულ სიმბოლოთა შესაბამისი f ფუნქციის მნიშვნელობებისა, გაერთიანებული კოდი იქნება ინვარიანტული მანძილის მიმართ, ანუ სისტემა იქნება ინვარიანტული. თვალსაჩინოებისათვის განვიხილოთ ასეთი მაგალითი: ვთქვათ არაორობითი კოდის სახით მოცემული გვაქვს წრფივი, ალფაბეტური სიჭარბის მქონე კოდი. ვთქვათ q=2 და Q=4. ცხადია, ოთხობითი კოდური სიმბოლოების ასახვა უნდა განხორციელდეს ორობითში გარკვეული წესით. ვთქვათ ასე: 0→00, 1→01, 2→11, 3→10. ზემოთ მოყვანილი მაგალითის შესაბამისად, აგრეთვე (5)-ის და (6)-ის გათვალისწინებით, გვექნება ნახ. 19-ზე ნაჩვენები სურათი, საიდანაც ჩანს, რომ თუ: f_H ფუნქციაში v_k კომპონენტები წარმოდგენილია წყვილი ბიტებით და f-ში თითო Q-ობითი სიმბოლოთი, მაშინ $d_{iH}=f_i$ ე.ი. ცხადია, რომ ლის და ჰემინგის მეტრიკები იდენტურია, ე.ი. ინვარიანტულია მთლიანი სისტემა ჰემინგის მანძილის მიმართ.



ნახ. 19. სისტემის ინვარიანტობის ინტერპრეტაცია

ზოგადად, ანალიზისას, ორივე კოდის ინვარიანტობის გამო f-ის გამოთვლის დროს ერთ-ერთ კომპონენტად შეგვიძლია ავირჩიოთ ნულოვანი სიმბოლო, ხოლო d_H-ის გამოთვლისას ნულოვანი კოდური სიტყვა.

ადვილი შესამჩნევია, რომ წრფივი Q-ობითი კოდის შემთხვევაში ნებისმიერი

$$f_i \in \begin{cases} \{\overline{0, Q/2}\}, & \text{როცა } Q \text{ ლუწია} \\ \{\overline{0, (Q-1)/2}\}, & \text{როცა } Q \text{ კენტია} \end{cases}$$

ჩაწერა $\overline{0, Q/2}$ $\overline{0, (Q-1)/2}$ გულისხმობს რიცხვთა ყველა მთელ მნიშვნელობას 0-დან Q/2-მდე (0-დან (Q-1)/2-მდე); ამასთან, თუ f-ის ერთობლიობიდან ერთნაირი f-ების რაოდენობას აღვნიშნავთ $|f_e|$-ით, მაშინ, თუ Q ლუწია

$$|f_e| = \begin{cases} 1, & \text{თუ } f_L=0 \text{ და } f_L=Q/2 \\ 2, & \text{ყველა სხვა შემთხვევაში} \end{cases}$$

ხოლო თუ Q კენტია

$$|f_e| = \begin{cases} 1, & \text{თუ } f_L=0 \\ 2, & \text{ყველა სხვა შემთხვევაში} \end{cases}$$

მაგალითად, თუ Q=6, და f-ის მნიშვნელობებს დავალაგებთ 6-ობითი კოდური სიმბოლოების ზრდის შესაბამისად, გვექნება

მიმდევრობა f=0, 1, 2, 3, 2, 1, რომელშიც 0-იანი და 3-იანი გვხვდება თითოჯერ, 1-იანი და 2-იანი ორ-ორჯერ; თუ Q=7, მაშინ f=0, 1, 2, 3, 3, 2, 1, რომელშიც 0-იანი გვხვდება ერთხელ, ხოლო ყველა დანარჩენი სიმბოლო ორ-ორჯერ. ცხადია, $|f|=Q$; შემდგომში ჰემინგის მანძილის მიმდევრობები აღნიშნული გვექნება d_H-ით; ნახ. 19-ზე მოყვანილი მაგალითისათვის $d_H = f$.

Q-ობითი სიმბოლოების ასახვისას ორობითებში, ბუნებრივია, ორობითი სიმბოლოების ბლოკის სიგრძისათვის უნდა კმაყოფილებოდეს პირობა:

$$n \geq \text{ceil}(\log_2 Q)$$

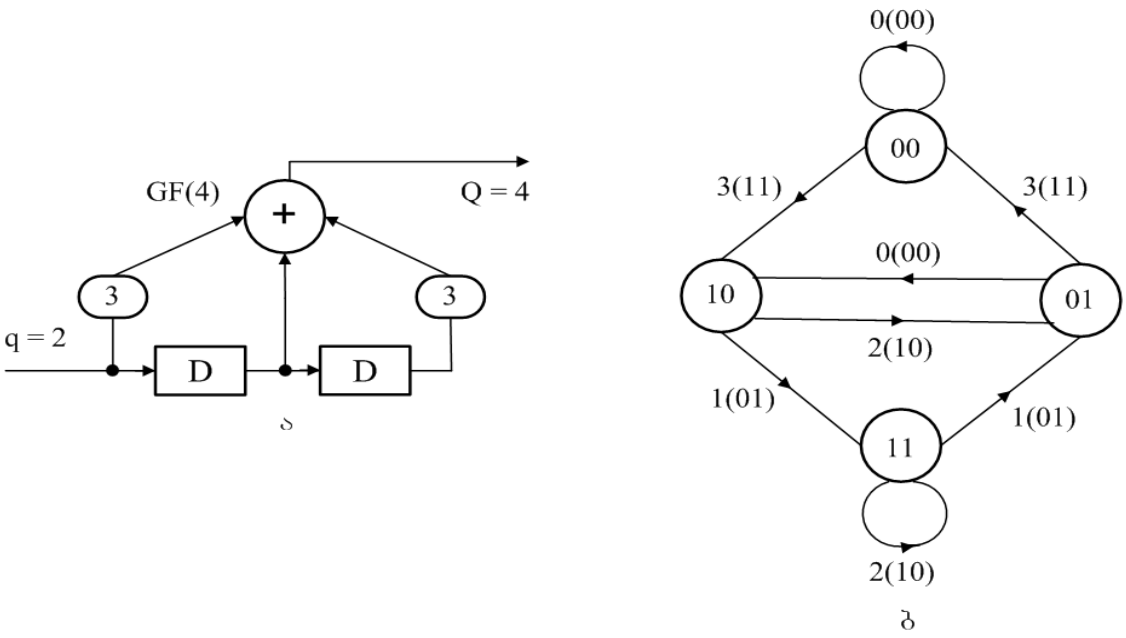
სადაც, ceil (x) აღნიშნავს x რიცხვის უდიდეს მთელამდე დამრგვალებას.

ორობითი კოდის ინვარიანტობის გამო, სისტემა ინვარიანტული იქნება მაშინაც, თუ

$$d_H \in \{\text{perms}(f)\}, \tag{7}$$

რაც ნაკლებად მკაცრი პირობაა (d_H=f)-თან შედარებით. (7)-დან ცხადია, რომ სისტემა ინვარიანტული იქნება სხვა ასახვის დროსაც (მაგალითად, 0→00, 1→01, 2→10, 3→11 და ა.შ.). (7)-ში perms(f) აღნიშნავს f-ის წევრების ყველა შესაძლო გადანაცვლებას.

ქვემოთ, ნახ. 20 ა-ზე ნაჩვენებია (313) ხვეულა კოდის კოდერი; ხოლო იქვე ნახ. 20 ბ-ზე მისი მდგომარეობათა დიაგრამა 0→00, 1→01, 2→10, 3→11 ასახვით. მოცემულ დიაგრამაზე ფრჩხილებში მოყვანილია ორობითი სიმბოლოებისაგან შედგენილი n=2 სიგრძის ბლოკები, რომლებიც გამოიყენებიან ოთხობითი სიმბოლოების ორობითში ასახვისათვის.



ნახ. 20. ხვეულა (313) კოდის კოდერი და მისი მდგომარეობათა დიაგრამა

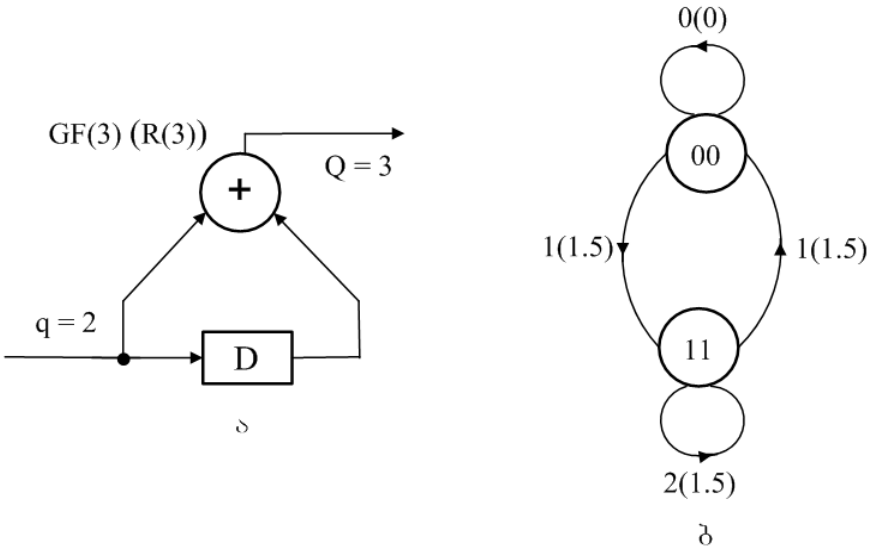
(313) კოდი აგებულია GF(4)-ზე, შესასვლელი სიმბოლოები კი ორობითია (q=2). აქ კოდირების სიჩქარე R=1/2, ხოლო თავისუფალი მანძილი ჰემინგის მიხედვით d_f=5. დიაგრამიდან ადვილი შესამჩნევია, რომ მოყვანილი სისტემა იდენტურია კარგად ცნობილი ორობითი (7,5) კოდისა [22,28]. ცხადია მოყვანილი სისტემა ინვარიანტულია ჰემინგის მანძილის მიმართ, რომელიც გამოითვლება ორბიტიანი ბლოკებისათვის.

(7)-ის განზოგადოებით, სისტემა ინვარიანტული იქნება ნებისმიერი A მეტრიკისათვის, თუ

$$d_A \in \{\text{perms}(a \cdot f)\},\ a>0; \qquad (8)$$

ცხადია, აქ d_A შეგვიძლია შევცვალოთ ევკლიდეს მეტრიკის შესაბამისი d^2_E ფუნქციით, სადაც ის წარმოადგენს ევკლიდური მანძილის კვადრატის მნიშვნელობას შესაბამის ელემენტარულ სიგნალებს შორის და (8) პირობის შესრულების შემთხვევაში გვექნება ევკლიდური მანძილის მიმართ ინვარიანტული სიგნალ-კოდური სისტემა. მაგალითად, თუ ავიღებთ GF(3)-ზე მოცემულ ხვეულა კოდს ალფაბეტური სიჭარბით და სამობით

ფაზამოდულირებულ (3PSK) სიმპლექსურ სიგნალს მათი კასკადირებით მივიღებთ ევკლიდური მანძილის მიმართ ინვარიანტულ სიგნალ-კოდურ სისტემას. უმარტივესი ხვეულა კოდისათვის ასეთი მაგალითი ნაჩვენებია ნახ. 21-ზე, სადაც მოყვანილია (11) სამობითი ხვეულა კოდი და მისი მდგომარეობათა დიაგრამა, რომელზედაც ფრჩხილებში მოცემული რიცხვები შეესაბამება ევკლიდური მანძილის კვადრატებს ელემენტარულ სიგნალებს შორის. აქ $d^2_E=1.5\cdot f$.



ნახ. 21. სამობით ველზე მოცემული (11) ხვეულა კოდის კოდერი

ნახ. 21 ბ-ზე მოყვანილი დიაგრამიდან ჩანს, რომ სიგნალ-კოდური სისტემის ნორმირებული ($2E_b$ –ით, სადაც E_b ერთ ბიტიანი სიმბოლოს შესაბამისი სიგნალის ენერგიაა) თავისუფალი ევკლიდური მანძილის კვადრატი ტოლია $d_f^2=3$, ხოლო კოდირების სიჩქარე R=0,63.

განხილულია ლის მეტრიკის გამოყენებით მანძილის მიმართ ინვარიანტული სისტემების აგება. მართალია, ნაშრომის დასაწყისში, ის განიხილება არაორობით-ორობით სისტემის მაგალითზე ან უფრო კონკრეტულად ლი-ჰემინგის მეტრიკისათვის, შემდგომში შედეგი გაზოგადოებულია ლი-სა და ევკლიდეს მეტრიკისათვის. მეთოდი

საკმარისად ზოგადია (მარტივიც) და მისი გამოყენება შეიძლება როგორც ჰემინგის მანძილის მიმართ ინვარიანტული ხვეულა კოდების ასაგებად, ასევე იმ სხვა მსგავსი კონსტრუქციებისათვისაც [29-31], სადაც გამოიყენებიან ხვეულა კოდები და სიგნალები; ბოლოს მოყვანილი მაგალითიც ამას ადასტურებს და თუ მოვახდენთ ზემოთ მოყვანილის რეზიუმირებას შეგვიძლია ზოგადად ვთქვათ: Q-ობითი ხვეულა კოდი ლის მეტრიკით, $d_A \in \{\mathrm{perms}(a \cdot f)\}$, $a>0$ შემთხვევაში წარმოქმნის ნებისმიერი A მეტრიკისათვის დისტანციურად ინვარიანტულ სისტემას.

3.2 კოდები რგოლზე და გალუას ველზე

ჩვენს მიერ, ზემოდ, ხშირად იყო ნახსენები გალუას ველი (GF) და მასთან დაკავშირებული კოდები. ვინაიდან მომავალში ჩვენ განვიხილავთ კოდებს ოთხობით ველზე (GF(4)) და ექვსობით რგოლზე (R(6)), ამიტომ აუცილებელია ამ ალგებრული სტრუქტურების განხილვა.

ვთქვათ, q მთელი დადებითი რიცხვია. $\{0, 1, 2, \ldots, q-1\}$ სიმრავლეში შემოვიღოთ შეკრების და გამრავლების ოპერაციები q-ს მოდულით, ე.ი. ამ სიმრავლის ნებისმიერი a და b ელემენტისათვის მივიღოთ

$$a+b=\mathrm{R}_q[a+b] \text{ და } a \cdot b = \mathrm{R}_q[a \cdot b],$$

სადაც, 3.1 ქვეთავის თანახმად, $\mathrm{R}_q[a+b]$ და $\mathrm{R}_q[a \cdot b]$, არის შესაბამისად $(a+b)$-ს და $(a \cdot b)$-ს q-ზე გაყოფისას მიღებული ნაშთები. ცხადია, რომ აღნიშნული სიმრავლე ასეთნაირად განსაზღვრულ ოპერაციებთან ერთად ქმნის კომუტატიურ რგოლს. ასეთ რგოლს ეწოდება მთელ რიცხვთა რგოლი (ან ნაშთთა რგოლი) q-ს მოდულით და აღინიშნება $\mathrm{R}(q)$ სიმბოლოთი.

აღვნიშნოთ, რომ სამართლიანია თანაფარდობები:

$$\mathrm{R}_q[a+b]=\mathrm{R}_q\left[\mathrm{R}_q[a]+\mathrm{R}_q[b]\right] \text{ და } \mathrm{R}_q[a \cdot b]=\mathrm{R}_q\left[\mathrm{R}_q[a] \cdot \mathrm{R}_q[b]\right].$$

***თეორემა* 1.** იმისათვის, რომ მთელ რიცხვთა R(q) რგოლი იყოს ველი (ე.ი. გალუას GF(q) ველი), აუცილებელია და საკმარისი, რომ q იყოს მარტივი რიცხვი.

ქვემოთ მოყვანილია მთელ რიცხვთა R(q) რგოლში და მარტივი q რიცხვის შემთხვევაში შესაბამის გალუას GF(q) ველში ელემენტების შეკრების და გამრავლების ცხრილები.

ცხრილი 3. R(2), GF(2)

+	0	1
0	0	1
1	1	0

.	0	1
0	0	0
1	0	1

ცხრილი 4. R(3), GF(3)

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

.	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

ცხრილი 5. R(4)

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

.	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

ცხრილი 6. R(5), GF(5)

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

.	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

ცხრილი 5. R(6)

+	0 1 2 3 4 5	.	0 1 2 3 4 5
0	0 1 2 3 4 5	0	0 0 0 0 0 0
1	1 2 3 4 5 0	1	0 1 2 3 4 5
2	2 3 4 5 0 1	2	0 2 4 0 2 4
3	3 4 5 0 1 2	3	0 3 0 3 0 3
4	4 5 0 1 2 3	4	0 4 2 0 4 2
5	5 0 1 2 3 4	5	0 5 4 3 2 1

ცხრილი 6. R(7), GF(7)

+	0 1 2 3 4 5 6	.	0 1 2 3 4 5 6
0	0 1 2 3 4 5 6	0	0 0 0 0 0 0 0
1	1 2 3 4 5 6 0	1	0 1 2 3 4 5 6
2	2 3 4 5 6 0 1	2	0 2 4 6 1 3 5
3	3 4 5 6 0 1 2	3	0 3 6 2 5 1 4
4	4 5 6 0 1 2 3	4	0 4 1 5 2 6 3
5	5 6 0 1 2 3 4	5	0 5 3 1 6 4 2
6	6 0 1 2 3 4 5	6	0 6 5 4 3 2 1

ცხრილი 7. R(8)

+	0 1 2 3 4 5 6 7	.	0 1 2 3 4 5 6 7
0	0 1 2 3 4 5 6 7	0	0 0 0 0 0 0 0 0
1	1 2 3 4 5 6 7 0	1	0 1 2 3 4 5 6 7
2	2 3 4 5 6 7 0 1	2	0 2 4 6 0 2 4 6
3	3 4 5 6 7 0 1 2	3	0 3 6 1 4 7 2 5
4	4 5 6 7 0 1 2 3	4	0 4 0 4 0 4 0 4
5	5 6 7 0 1 2 3 4	5	0 5 2 7 4 1 6 3
6	6 7 0 1 2 3 4 5	6	0 6 4 2 0 6 4 2
7	7 0 1 2 3 4 5 6	7	0 7 6 5 4 3 2 1

ცხრილი 8. R(9)

+	0 1 2 3 4 5 6 7 8
0	0 1 2 3 4 5 6 7 8
1	1 2 3 4 5 6 7 8 0
2	2 3 4 5 6 7 8 0 1
3	3 4 5 6 7 8 0 1 2
4	4 5 6 7 8 0 1 2 3
5	5 6 7 8 0 1 2 3 4
6	6 7 8 0 1 2 3 4 5
7	7 8 0 1 2 3 4 5 6
8	8 0 1 2 3 4 5 6 7

.	0 1 2 3 4 5 6 7 8
0	0 0 0 0 0 0 0 0 0
1	0 1 2 3 4 5 6 7 8
2	0 2 4 6 8 1 3 5 7
3	0 3 6 0 3 6 0 3 6
4	0 4 8 3 7 2 6 1 5
5	0 5 1 6 2 7 3 8 4
6	0 6 3 0 6 3 0 6 3
7	0 7 5 3 1 8 6 4 2
8	0 8 7 6 5 4 3 2 1

ვთქვათ, F-ნებისმიერი ველია. თუ მრავალწევრის ყველა კოეფიციენტი ეკუთვნის ამ ველს, მაშინ ასეთ მრავალწევრს ეწოდება F ველზე მოცემული მრავალწევრი. ამ პარაგრაფში ჩვენ ძირითადად განვიხილავთ გალუას GF(q) ველზე მოცემულ მრავალწევრებს. შევნიშნოთ, რომ თითქმის ყველაფერი ის, რაც წინა პარაგრაფში ითქვა მთელ რიცხვთა რგოლზე მოცემულ მრავალწევრებზე, ძალაში რჩება GF(q) ველზე მოცემულ მრავალწევრებისთვისაც. ისევე, როგორც მთელ რიცხვთა რგოლზე მოცემულ მრავალწევრების შემთხვევაში, GF(q) ველზე მოცემულ მრავალწევრთა სიმრავლეც ქმნის რგოლს. ჩვენ აქ ძირითადად განვიხილავთ შემთხვევას, როცა ამ რგოლში როგორც შეკრების, ასევე

გამრავლების ოპერაციები ხორციელდება გარკვეული $p(x)$ მრავალწევრის მოდულით.

***დებულება* 1.** ვთქვათ, $p(x)$ არის GF(q) ველზე მოცემული არანულოვანი ხარისხის ნორმირებული მრავალწევრი. მაშინ GF(q) ველზე მოცემულ მრავალწევრთა $\{R(x)\}$ რგოლი, რომელშიც განსაზღვრულია შეკრების და გამრავლების ოპერაციები $p(x)$ მრავალწევრის მოდულით, შედგება ამ ველზე მოცემულ ყველა ისეთი $T(x)$ მრავალწევრისაგან, რომელთა ხარისხი ნაკლებია $p(x)$ მრავალწევრის ხარისხზე, ე.ი. $T(x) \in \{R(x)\} \Leftrightarrow \deg(T(x) < \deg p(x)$.

***თეორემა* 2.** ნორმირებული $p(x)$ მრავალწევრით წარმოქმნილი მრავალწევრთა რგოლი იქნება ველი მაშინ და მხოლოდ მაშინ, როცა $p(x)$ მრავალწევრი არის მარტივი.

თუ ამ თეორემას შევადარებთ მსგავს თეორემას მთელ რიცხვთა რგოლის შემთხვევაში, დავინახავთ სრულ ანალოგიას მთელ რიცხვთა რგოლებსა და მრავალწევრთა რგოლებს შორის.

ზემოთ მოყვანილი დებულება 1 და თეორემა 2-ის გამოყენებით შეგვიძლია ჩამოვაყალიბოთ გალუას ველის აგების პროცედურა მრავალწევრთა რგოლზე დაფუძნებით:

ვთქვათ, q მარტივი რიცხვია და $p_n(x)$-GF(q) ველზე მოცემული n ხარისხის ($n \geq 1$) ნორმირებული მარტივი მრავალწევრი. მაშინ ამავე ველზე მოცემული ყველა ისეთი მრავალწევრი, რომლის ხარისხი ნაკლებია n-ზე, ქმნის GF(q^n) გალუას ველს q^n ელემენტით.

მაგალითი. ვისარგებლოთ ზემოთმოყვანილით და ავაგოთ GF(9) გალუას ველი. აქ $q=3$ და $n=2$ ($9=3^2$). ამის შესაბამისად შევარჩიოთ მეორე ხარისხის მარტივი ნორმირებული მრავალწევრი GF(3) ველზე. ასეთად გამოგვადგება, მაგალითად, მრავალწევრი $p_2(x)=x^2+2x+2$ ამოვწეროთ ორზე ნაკლები ხარისხის ყველა შესაძლო მრავალწევრი GF(3) ველზე:

$p_1(x) = 0$, $p_2(x) = 1$, $p_3(x) = x$, $p_4(x) = x + 1$, $p_5(x) = 2x + 1$, $p_6(x) = 2$, $p_7(x) = =2x$, $p_8(x) = 2x + 2$, $p_9(x) = x + 2$.

მოცემული მრავალწევრების შეკრება და გამრავლება განვახორციელოთ $p_2(x)=x^2+2x+2$ მრავალწევრის მოდულით. ასეთნაირად განსაზღვრულ ოპერაციებთან ერთად ამ ცხრა მრავალწევრისაგან შედგენილი სიმრავლე ქმნის GF(9) ველს (იხ. ცხრილი 9 და ცხრილი 10).

ამ ცხრილებიდან გამომდინარეობს, რომ, მაგალითად, $2x+1$ ელემენტის მოპირდაპირე ელემენტია $x+2$, ხოლო ამავე ელემენტის შებრუნებული ელემენტია $2x$.

ცხრილი 9. შეკრების ოპერაციები მრავალწევრთა GF(9)-ისათვის

+	0	1	x	x+1	2x+1	2	2x	2x+2	x+2
0	0	1	x	x+1	2x+1	2	2x	2x+2	x+2
1	1	2	x+1	x+2	2x+2	0	2x+1	2x	x
x	x	x+1	2x	2x+1	1	x+2	0	2	2x+2
x+1	x+1	x+2	2x+1	2x+2	2	x	1	0	2x
2x+1	2x+1	2x+2	1	2	x+2	2x	x+1	x	0
2	2	0	x+2	x	2x	1	2x+2	2x+1	x+1
2x	2x	2x+1	0	1	x+1	2x+2	x	x+2	2
2x+2	2x+2	2x	2	0	x	2x+1	x+2	x+1	1
x+2	x+2	x	2x+2	2x	0	x+1	2	1	2x+1

ცხრილი 10. გამრავლების ოპერაციები მრავალწევრთა GF(9)-ისთვის

.	0	1	x	x+1	2x+1	2	2x	2x+2	x+2
0	0	0	0	0	0	0	0	0	0
1	0	1	x	x+1	2x+1	2	2x	2x+2	x+2
x	0	x	x+1	2x+1	2	2x	2x+2	x+2	1
x+1	0	x+1	2x+1	2	2x	2x+2	x+2	1	x
2x+1	0	2x+1	2	2x	2x+2	x+2	1	x	x+1
2	0	2	2x	2x+2	x+2	1	x	x+1	2x+1
2x	0	2x	2x+2	x+2	1	x	x+1	2x+1	2
2x+2	0	2x+2	x+2	1	x	x+1	2x+1	2	2x
x+2	0	x+2	1	x	x+1	2x+1	2	2x	2x+2

GF(9) ველის ელემენტები ჩვენ შეგვიძლია წარმოვადგინოთ გარდა მრავალწევრებისა სხვა სახითაც. ალტერნატიული ვარიანტები ასახულია ცხრილ 11-ში.

ცხრილი 11. GF(9) -ის ელემენტების შესაძლო წარმოდგენები

წარმოდგენა ხარისხით	წარმოდგენა მრავალწევრით	3-ობითი წარმოდგენა	წარმოდგენა მთელი რიცხვებით
0	0	00	0
x^0	1	01	1
x^1	x	10	2
x^2	x+1	11	3
x^3	2x+1	21	4
x^4	2	02	5
x^5	2x	20	6
x^6	2x+2	22	7
x^7	x+2	12	8

აღნიშნული ცხრილი საჭიროებს ახსნას:

1. ცხრილი 10-ის თანახმად, ველის ყველა ელემენტი, გარდა ნულოვანისა, წარმოადგენს ამავე ველის *x* ელემენტის ხარისხს. სწორედ ეს უდევს საფუძვლად ველის ელემენტების წარმოდგენას *x* ელემენტის ხარისხების საშუალებით;

2. ყოველი მრავალწევრი ცალსახად განისაზღვრება თავისი კოეფიციენტებით, სწორედ ეს კოეფიციენტები მონაწილეობენ მრავალწევრების წარმოდგენაში სამობითი რიცხვების საშუალებით;

3. GF(9) სასრული ველის ყოველ ელემენტს ცალსახად შეგვიძლია შევუსაბამოთ გარკვეული მთელი რიცხვი {0,1,2,3,4,5,6,7,8} სიმრავლიდან და ეს ყველაფერი ასახულია ცხრილი 11-ის მეოთხე სვეტში.

GF(9) ველის ელემენტების შეკრების და გამრავლების ოპერაციები იმ შემთხვევაში, როცა ეს ელემენტები წარმოდგენილია მთელი რიცხვების საშუალებით, მოყვანილია ცხრილ 12-ში:

ცხრილი 12. ალგებრული ოპერაციები მთელ რიცხვთა GF(9)-თვის

+	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	5	3	8	7	0	4	6	2
2	2	3	6	4	1	8	0	5	7
3	3	8	4	7	5	2	1	0	6
4	4	7	1	5	8	6	3	2	0
5	5	0	8	2	6	1	7	4	3
6	6	4	0	1	3	7	2	8	5
7	7	6	5	0	2	4	8	3	1
8	8	2	7	6	0	3	5	1	4

.	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8
2	0	2	3	4	5	6	7	8	1
3	0	3	4	5	6	7	8	1	2
4	0	4	5	6	7	8	1	2	3
5	0	5	6	7	8	1	2	3	4
6	0	6	7	8	1	2	3	4	5
7	0	7	8	1	2	3	4	5	6
8	0	8	1	2	3	4	5	6	7

განსაზღვრება. GF(q) ველის ელემენტს ეწოდება პრიმიტიული ელემენტი, თუ ამ ველის ნებისმიერი არანულოვანი ელემენტი წარმოადგენს ამ ელემენტის ხარისხს.

მტკიცდება, რომ ყოველ გალუას ველში არსებობს პრიმიტიული ელემენტი.

მაგალითი. GF(5) ველის პრიმიტიული ელემენტია 2. მართლაც, $1=2^4$, $2=2^1$, $3=2^3$, $4=2^2$, ე.ი GF(5) ველის ყველა არანულოვანი ელემენტი წარმოადგენს 2-ის ხარისხს.

მაგალითი. GF(9) ველის პრიმიტიული ელემენტია 2, რადგან ამ ველში (იხ. ცხრილი 12)

$$2^1=2,\ 2^2=3,\ 2^3=4,\ 2^4=5,\ 2^5=6,\ 2^6=7,\ 2^7=8,\ 2^8=1,$$

ე.ი. 2-ის ახარისხებით მიიღება GF(9)={0,1,2,3,4,5,6,7,8} ველის ყველა არანულოვანი ელემენტი.

ცხადია, რომ ნებისმიერ სასრულ ველში არსებობს ელემენტების მინიმალური რაოდენობის მქონე ქვეველი. ამ რაოდენობას ეწოდება მოცემული ველის მახასიათებელი. ნებისმიერი სასრული ველის მახასიათებელი მარტივი რიცხვია. გალუას GF(p^n) ველის მახასიათებელი არის p.

F ველს ეწოდება F_0 ველის გაფართოება, თუ F_0 არის F ველის ქვეველი ($F_0 \subset F$). გალუას ველის აგების ზემოთმოყვანილი პროცედურის გათვალისწინებით შეიძლება ითქვას, რომ GF(q) ველზე მოცემული მარტივი $p(x)$ მრავალწევრის მოდულით წარმოქმნილი მრავალწევრთა ველი წარმოადგენს GF(q) ველის გაფართოებას. თუ ამ გაფართოებულ ველში x მრავალწევრი, განხილული როგორც ველის ელემენტი, პრიმიტიულია, მაშინ შესაბამის მარტივ $p(x)$ მრავალწევრს ეწოდება პრიმიტიული მრავალწევრი.

ნებისმიერ GF(q) ველისათვის არსებობს ერთი მაინც პრიმიტიული მრავალწევრი. მაგალითად, როგორც ჩვენ უკვე ვაჩვენეთ, GF(3) ველის შემთხვევაში $p(x)=x^2+2x+2$ მრავალწევრი პრიმიტიულია, რადგან მის

საფუძველზე აგებული ამ ველის გაფართოების ნებისმიერი მრავალწევრი წარმოადგენს x ელემენტის ხარისხს (იხ. ცხრილი 11).

პრიმიტიული მრავალწევრის გამოყენება საშუალებას გვაძლევს შედარებით მარტივად შევადგინოთ გამრავლების ცხრილები მოცემული ველისათვის.

მაგალითი. შევადგინოთ ალგებრული ოპერაციების ცხრილები GF(4) ველისათვის ორობით ველზე მოცემული $p(x)=x^2+x+1$ პრიმიტიული მრავალწევრის გამოყენებით.

ცხრილი 13. პრიმიტიული ელემენტის გამოყენებით შედგენილი ალგებრული ოპერაციების ცხრილი GF(4)-თვის

+	0	x^0	x^1	x^2
0	0	1	x	x+1
x^0	1	0	x+1	0
x^1	x	x+1	0	1
x^2	x+1	x	1	0

.	0	x^0	x^1	x^2
0	0	0	0	0
x^0	0	1	x	x+1
x^1	0	x	x+1	1
x^2	0	x+1	1	x

ცხრილი 14. GF(4)-ის ელემენტების შესაძლო წარმოდგენა

წარმოდგენა პრიმიტიული ელემენტის ხარისხით	წარმოდგენა მრავალწევრით	ორობითი წარმოდგენა	წარმოდგენა მთელი რიცხვებით
0	0	00	0
x^0	1	01	1
x^1	x	10	2
x^2	x+1	11	3

თუ ვისარგებლებთ 13 და 14 ცხრილებით, მთელი რიცხვებისათვის გვექნება:

ცხრილი 15. ალგებრული ოპერაციები მთელ რიცხვთა GF(4)-თვის

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

.	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

ქვემოთ მოყვანილია მსგავსი ცხრილები GF(8) ველისათვის, შედგენილი პრიმიტიული $p(x)=x^3+x+1$ მრავალწევრის გამოყენებით.

ცხრილი 16. GF(8)-ის ელემენტების შესაძლო წარმოდგენები

წარმოდგენა პრიმიტიული ელემენტის ხარისხით	წარმოდგენა მრავალწევრით	ორობითი წარმოდგენა	წარმოდგენა მთელი რიცხვებით
0	0	000	0
x^0	1	001	1
x^1	x	010	2
x^2	x^2	100	3
x^3	x+1	011	4
x^4	x^2+x	110	5
x^5	x^2+x+1	111	6
x^6	x^2+1	101	7

ცხრილი 17. შეკრების ოპერაციები მრავალწევრთა გამოყენებით GF(8)-თვის

+	0	1	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1
0	0	1	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1
1	1	0	x+1	x^2+1	x	x^2+x+1	x^2+x	x^2
x	x	x+1	0	x^2+x	1	x^2	x^2+1	x^2+x+1
x^2	x^2	x^2+1	x+ x^2	0	x^2+x+1	x	x+1	1
x+1	x+1	x	1	x^2+x+1	0	x^2+1	x^2	x^2+x
x^2+x	x^2+x	x^2+x+1	x^2	x	x^2+1	0	1	x+1
x^2+x+1	x^2+x+1	x^2+x	x^2+1	x+1	x^2	1	0	x
x^2+1	x^2+1	x^2	x^2+x+1	1	x^2+x	x+1	x	0

ცხრილი 18. გამრავლების ოპერაციები მრავალწევრთა გამოყენებით GF(8)-თვის

.	0	1	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1
0	0	0	0	0	0	0	0	0
1	0	1	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1
x	0	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1	1
x^2	0	x^2	x+1	x^2+x	x^2+x+1	x^2+1	1	x
x+1	0	x+1	x^2+x	x^2+x+1	x^2+1	1	x	x^2
x^2+x	0	x^2+x	x^2+x+1	x^2+1	1	x	x^2	x+1
x^2+x+1	0	x^2+x+1	x^2+1	1	x	x^2	x+1	x^{2+}x
x^2+1	0	x^2+1	1	x	x^2	x+1	x^2+x	x^2+x+1

ცხრილი 19. ალგებრული ოპერაციები მთელ რიცხვთა GF(8)-თვის

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	4	7	2	6	5	3
2	2	4	0	5	1	3	7	6
3	3	7	5	0	6	2	4	1
4	4	2	1	6	0	7	3	5
5	5	6	3	2	7	0	1	4
6	6	5	7	4	3	1	0	2
7	7	3	6	1	5	4	2	0

.	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	3	4	5	6	7	1
3	0	3	4	5	6	7	1	2
4	0	4	5	6	7	1	2	3
5	0	5	6	7	1	2	3	4
6	0	6	7	1	2	3	4	5
7	0	7	1	2	3	4	5	6

როგორც ზემოთ მოყვანილიდან გამომდინარეობს, მრავალწევრთა რგოლზე დაფუძნებული გალუას ველის ასაგებად უნდა გვქონდეს შესაბამისი მარტივი ან, რაც უკეთესია, პრიმიტიული მრავალწევრები. ზოგიერთი კონკრეტული პრიმიტიული მრავალწევრის მაგალითი მოყვანილია ცხრილ 20-ში.

ცხრილი 20. ზოგიერთი პრიმიტიული მრავალწევრი

GF(2)	GF(3)	GF(5)	GF(7)
x^2+x+1	x^2+x+2	x^2+x+2	x^2+x+3
x^3+x+1	x^3+2x+1	x^3+3x+2	x^3+3x+2
x^4+x+1	x^4+x+2	x^4+x^2+2x+2	x^4+x^2+3x+5
x^5+x^2+1	x^5+2x+1	x^5+4x+2	
x^6+x+1	x^6+x+2		
x^7+x^3+1			

ამ თავის ბოლოს მოვიყვანოთ სასრულ ველებთან დაკავშირებული რამდენიმე მნიშვნელოვანი წინადადება:

– ნებისმიერი სასრული ველის ელემენტების რაოდენობა მარტივი რიცხვის ხარისხის ტოლია;

– ნებისმიერი მარტივი p რიცხვისათვის და ნებისმიერი ნატურალური m რიცხვისათვის არსებობს ველი p^m ელემენტით;

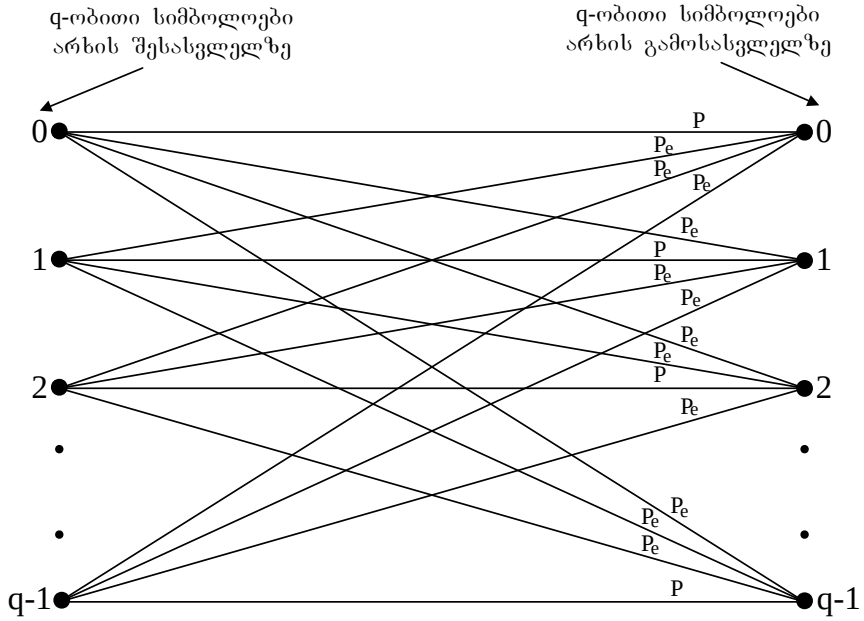
– თუ ორი სასრული ველის ელემენტების რაოდენობა ერთმანეთის ტოლია, მაშინ ეს ორი ველი იზომორფულია;

– GF(p^m) ველის ნებისმიერი ელემენტი წარმოადგენს $x^{p^m}-x$ მრავალწევრის ფესვს.

დამატებითი მასალა იხილეთ [32-38]-ში.

3.3 ახალი ალფაბეტური სიჭარბის მქონე კოდები

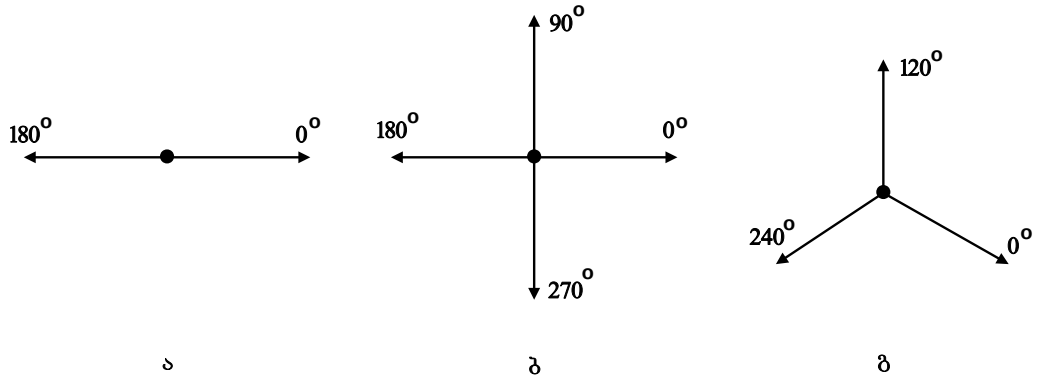
აქ და მომავალში, ჩვენს მიერ მოყვანილ გადაცემის სისტემებში გარემო, რომელშიც ვრცელდება გარკვეული ფორმით წარმოდგენილი მიმდევრობა, შეიძლება იყოს მექანიკური, ელექტრული, ელექტრომაგნიტური და ა.შ.; ხოლო ერთ-ერთი არხის სახით განიხილება მოდელი, რომელიც ნაჩვენებია ნახ. 22-ზე. აქ იგულისხმება, რომ სხვადასხვა სიმბოლოთა გადაცემისას ადგილი აქვს დამოუკიდებელ შეცდომებს. ესაა ე.წ. q-ობითი სიმეტრიული არხი მეხსიერების გარეშე, რომელშიც სიმბოლოს არასწორად მიღების ალბათობა $P_e=(1-P)/(q-1)$, სადაც P არის სიმბოლოს სწორად მიღების ალბათობა. ვინაიდან ჩვენ განვიხილავთ ორობით კოდებს, ამიტომ ახალი ხვეულა კოდები აგებული იქნებიან ორობითი სიმეტრიული არხისათვის (q=2);



ნახ. 22. q - ობითი სიმეტრიული არხის მოდელი

აქვე იქნება ნაჩვენები, რომ ეს კოდები ოპტიმალურები (მანძლითა მიხედვით) იქნებიან ორობითი და ოთხობითი ფაზამოდულირებული სიგნალებისათვის (ნახ. 23 ა,ბ); ხვეულა კოდები სამობითი

გამოსასვლელით აგებული იქნება სამობითი სიმპლექსური ფაზამოდულირებული სიგნალებისათვის (ნახ. 23 გ). აქ არხების სახით განიხილებიან არხები თეთრი გაუსის ხმაურით, რომელთაც აქვთ ნულოვანი დისპერსია და N_0 სიმძლავრის სპექტრალური სიმკვრივე. თითოეული ვექტორის სიგრძე ნახ. 23-ზე ტოლია $\sqrt{Es}$, სადაც Es სიგნალის ენერგიაა.



ნახ. 23. ორობითი (ა), ოთხობითი (ბ) და სამობითი (გ) ფაზამოდულირებული სიგნალები

3.1 პარაგრაფსა და [39]-ში მოყვანილი მასალის შესაბამისად ახალი ხვეულა კოდები იქნებიან მანძილის მიმართ ინვარიანტულები. 2.2 პარაგრაფში მოყვანილი ალგორითმისა და მის შესაბამისად შედგენილი პროგრამის (იხ. დანართი 1) გამოყენებით ნაპოვნი იქნა ახალი, მანძილის მიმართ ინვარიანტული ხვეულა კოდები, რომლებიც მოყვანილი არიან 21 და 22 ცხრილებში. აქ ისინი წარმოდგენილი არიან თავიანთი მანძილის სპექტრებით (იხ. პარაგრაფი 2.3), სადაც d ჰემინგის მანძილია (ე. ი. $d = d_H$), d^2 არის ევკლიდური მანძილის კვადრატი, a მოცემული მანძილის (d ან d^2) მქონე გზების შესაბამისი გადაცემული ბიტების რაოდენობა, L იარუსების ის რაოდენობაა კოდურ გისოსზე, სადაც მანძილთა სპექტრში გვაქვს $\max(d)$ ან $\max(d^2)$.

ცხრილი 21. ხვეულა კოდები ორობითი შესასვლელითა და ორობითი გამოსასვლელით

3 1 ***L*=21**		**3 1 3** ***L*=46**		**3 3 1 3** ***L*=49**		**3 2 1 1 3** ***L*=68**		**3 1 3 1 1 3** ***L*=80**	
d_i	a_i	d_i	a_i	d_i	a_i	d_i	a_i	d_i	a_i
3	1	5	1	6	2	7	4	8	4
4	2	6	4	7	7	8	12	9	11
5	3	7	12	8	18	9	20	10	36
6	4	8	32	9	49	10	72	11	83
7	5	9	80	10	130	11	225	12	250
8	6	10	192	11	333	12	500	13	630
9	7	11	448	12	836	13	1324	14	1776
10	8	12	1024	13	2069	14	3680	15	4531
11	9	13	2304	14	5060	15	8967	16	11982
12	10	14	5120	15	12255	16	22270	17	30474
13	11	15	11264	16	29444	17	57403	18	78492
14	12	16	24576	17	70267	18	142234	19	198907
15	13	17	53248	18	166726	19	348830	20	504730
16	14	18	114688	19	393635	20	867106	21	1270141
17	15	19	245760	20	925334	21	2134239	22	3190906
18	16	20	524288	21	2166925	22	5205290	23	7978154
19	17	21	1114112	22	5057286	23	12724352	24	19903010
20	18	22	2359296	23	11767305	24	31022962	25	49491586
21	19	23	4980736	24	27305864	25	75250693	26	122799106
22	20	24	10485760	25	63207473	26	182320864	27	303950322
23	21	25	22020096	26	145986568	27	441125164	28	750869456

3 2 0 3 3 1 3 ***L*=90**		**3 1 3 1 2 2 1 3** ***L*=93**		**3 1 3 3 2 1 0 1 3** ***L*=102**		**3 2 2 0 1 3 2 1 2 3** ***L*=110**		**3 1 3 3 0 3 2 2 0 1 3** ***L*=112**	
d_i	a_i	d_i	a_i	d_i	a_i	d_i	a_i	d_i	a_i
10	36	10	4	12	33	12	8	14	71
11	0	11	10	13	0	13	0	15	0
12	211	12	64	14	281	14	154	16	419
13	0	13	130	15	0	15	0	17	0
14	1404	14	318	16	2179	16	1064	18	3383
15	0	15	905	17	0	17	0	19	0
16	11633	16	2424	18	15035	18	7346	20	23484
17	0	17	6187	19	0	19	0	21	0
18	77433	18	17422	20	105166	20	52073	22	157716
19	0	19	44417	21	0	21	0	23	0
20	502690	20	113904	22	692330	22	347879	24	1048620
21	0	21	297764	23	0	23	0	25	0
22	3322763	22	763260	24	4580007	24	2308011	26	6914707
23	0	23	1934897	25	0	25	0	27	0
24	21292910	24	4933300	26	29692894	26	15074111	28	44725065
25	0	25	12557730	27	0	27	0	29	0
26	134365911	26	31730448	28	190453145	28	97031904	30	286678096
27	0	27	80015153	29	0	29	0	31	0
28	843425871	28	201670146	30	1208999091	30	619561390	32	1822054919
29	0	29	506533510	31	0	31	0	33	0
30	5245283348	30	1269364860	32	7622677693	32	3921984002	34	11496908044

ცხრილი 22. ხვეულა კოდები ორობითი შესასვლელითა და სამობითი გამოსასვლელით

1 1 ***L*=21**		**1 1 2** ***L*=66**		**1 1 1 2** ***L*=57**		**1 1 1 2 2** ***L*=116**		**1 1 1 1 2 2** ***L*=83**	
d_i^2	a_i	d_i^2	a_i	d_i^2	a_i	d_i^2	a_i	d_i^2	a_i
3	1	4.5	3	6	6	7.5	5	9	17
4.5	2	6	15	7.5	6	9	42	10.5	39
6	3	7.5	58	9	58	10.5	135	12	187
7.5	4	9	201	10.5	118	12	727	13.5	683
9	5	10.5	655	12	507	13.5	3018	15	2741
10.5	6	12	2052	13.5	1284	15	12783	16.5	10322
12	7	13.5	6255	15	4323	16.5	51782	18	37096
13.5	8	15	18687	16.5	11846	18	210330	19.5	136834
15	9	16.5	54974	18	36009	19.5	835046	21	488901
16.5	10	18	159765	19.5	100844	21	3294834	22.5	1749291
18	11	19.5	459743	21	292830	22.5	12870612	24	6190259
19.5	12	21	1312200	22.5	821568	24	49963630	25.5	21803038
21	13	22.5	3719643	24	2330668	25.5	192707673	27	76377332
22.5	14	24	10482351	25.5	6509702	27	739596902	28.5	266343461
24	15	25.5	29391490	27	18219432	28.5	2825474756	30	925251279
25.5	16	27	82048737	28.5	50592106	30	10752036148	31.5	3202795872
27	17	28.5	228160495	30	140349045	31.5	40771803481	33	11052511705
28.5	18	30	632293452	31.5	387559772	33	154127275727	34.5	38033968207
30	19	31.5	1746896199	33	1068269069	34.5	581007132093	36	130555763220
31.5	20	33	4813063455	34.5	2935542798	36	2184683661144	37.5	447122121024
33	21	34.5	13228122758	36	8051416787	37.5	8195990895282	39	1528107655812

ა.

1 1 1 1 2 2 2 *L*=163		**1 1 0 1 1 2 2 2** *L*=117		**1 0 1 1 1 1 2 2 2** *L*=180		**1 1 1 1 2 2 1 0 1 1** *L*=197		**1 1 0 1 1 0 1 2 2 2 2** *L*=159	
d_i^2	a_i	d_i^2	a_i	d_i^2	a_i	d_i^2	a_i	d_i^2	a_i
9	3	10.5	8	12	32	13.5	75	13.5	21
10.5	11	12	28	13.5	95	15	297	15	59
12	100	13.5	129	15	751	16.5	1306	16.5	336
13.5	295	15	634	16.5	2854	18	7290	18	1531
15	1557	16.5	3257	18	14105	19.5	33419	19.5	6661
16.5	6256	18	13549	19.5	64101	21	153429	21	30546
18	27000	19.5	57889	21	294704	22.5	708669	22.5	132964
19.5	108696	21	245965	22.5	1304178	24	3212938	24	579101
21	442192	22.5	1027205	24	5763673	25.5	14428284	25.5	2486497
22.5	1762717	24	4244525	25.5	25137830	27	64218919	27	10596909
24	7005217	25.5	17367881	27	109332756	28.5	283986883	28.5	44841232
25.5	27555165	27	70640257	28.5	471215062	30	1248820820	30	188442686
27	107912228	28.5	285414744	30	2022521534	31.5	5463283722	31.5	788745319
28.5	420128113	30	1147440092	31.5	8638086694	33	23801148162	33	3284832842
30	1628896996	31.5	4591770680	33	36748343314	34.5	103293194002	34.5	13626151524
31.5	6289690428	33	18300413810	34.5	155763088262	36	446741415859	36	56328749339
33	24204148568	34.5	72682354368	36	658167323756	37.5	1926344546847	37.5	232139114131
34.5	92850251450	36	287757299534	37.5	2773028465632	39	8283804419885	39	953988726757
36	355205342557	37.5	1136034224353	39	11653734622624	40.5	35535275442910	40.5	3910613200080
37.5	1355441391359	39	4473536062828	40.5	48860808634532	42	152097360173159	42	15993716896091
39	5160549482108	40.5	17575524972785	42	204426787916078	43.5	649681793202754	43.5	65276103262934

ვინაიდან ცხრილებში მოყვანილი ხვეულა კოდის კოდერებს ექნებათ ერთი შესასვლელი და ერთი გამოსასვლელი, ამიტომ კოდირების სიჩქარე ტოლია

$$R=\log_2(q)/\log_2(Q),$$

სადაც, q კოდერის შესასვლელი სიმბოლოების ალფაბეტის ზომაა, ხოლო Q არის კოდერის გამოსასვლელი სიმბოლოების ალფაბეტის ზომა. ცხადია, მაშინ ცხრილ 21-ში მოყვანილი კოდებისათვის, სადაც q=2, Q=4, R=1/2; აქ d მანძილია ჰემინგის მიხედვით, ე. ი. $d= d_H$. უნდა აღინიშნოს, რომ, მართალია ცხრილ 21-ში მოყვანილი კოდები ოპტიმალურები არიან ორობითი სიმეტრიული არხისათვის, ისინი ოპტიმალურები იქნებიან გაუსის არხებისათვის ორობითი (M=2) ფაზამოდულირებული სიგნალების (BPSK) გადაცემისას. ცნობილია, რომ ფაზამოდულირებულ სიგნალებს შორის ევკლიდური მანძილის კვადრატი

$$d^2=2Es(1-cos\Delta\varphi),$$

სადაც, $\Delta\varphi$ არის განსხვავებულ სიგნალებს შორის ფაზათა სხვაობა; ცხადია, BPSK-ს შემთხვევისთვის $\Delta\varphi=180^o$ და $d^2=4Es=4E_bR\log_2(M)$. აქ თუ მოვახდენთ მის ნორმირებას $2E_b$–ით (E_b ერთი ბიტის გადამტანი სიგნალის ენერგიაა) გვექნება $d^2= d_H$. ძნელი არაა ვაჩვენოთ, რომ ოთხობითი (M=4) ფაზამოდულირებული სიგნალების (QPSK) გადაცემისას, თუ კოდური სიმბოლოების წყვილების ასახვა ფაზების მნიშვნელობებში ხდება პრინციპით (ე.წ. გრეის კოდი):

$$00\rightarrow0^0 \qquad 01\rightarrow90^0 \qquad 11\rightarrow180^0 \qquad 10\rightarrow270^0$$

მაშინაც $d^2= d_H$.

ე.ი. თუ ცხრილ 21-ში მოყვანილი კოდების გამოყენებით ავაგებთ სიგნალ-კოდურ სისტემებს გაუსის არხისათვის, BPSK და QPSK სიგნალების ბაზაზე, ორივე შემთხვევაში სიგნალ-კოდური სისტემის ევკლიდური

მანძილის კვადრატის მნიშვნელობა ტოლი იქნება შესაბამისი კოდის ჰემინგის მანძილისა ($d^2 = d_H$).

ცხრილ 22-ში მოყვანილია სამობითი, ალფაბეტური სიჭარბის მქონე ხვეულა კოდებისა (q=2, Q=3), და სამობითი სიმპლექსური ფაზამოდულიბული სიგნალის (TPSK) ბაზაზე აგებული სიგნალ-კოდური სისტემის დისტანციური მახასიათებლები.

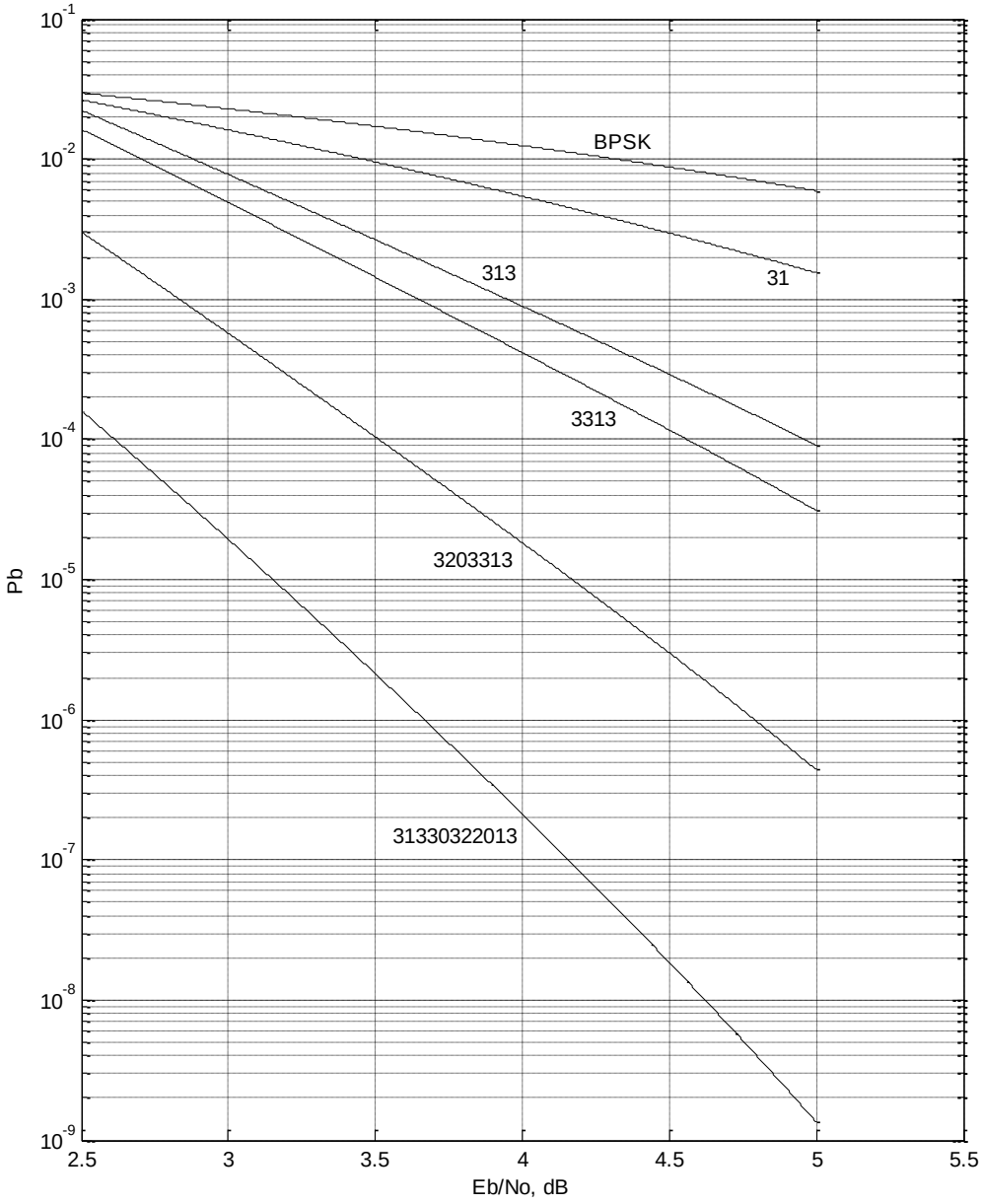
ორივე ცხრილში კოდები ჩაწერილი არიან Q-ობით ფორმაში.

3.4 ახალ კოდთა ბაზაზე აგებულ სისტემათა მახასიათებლები

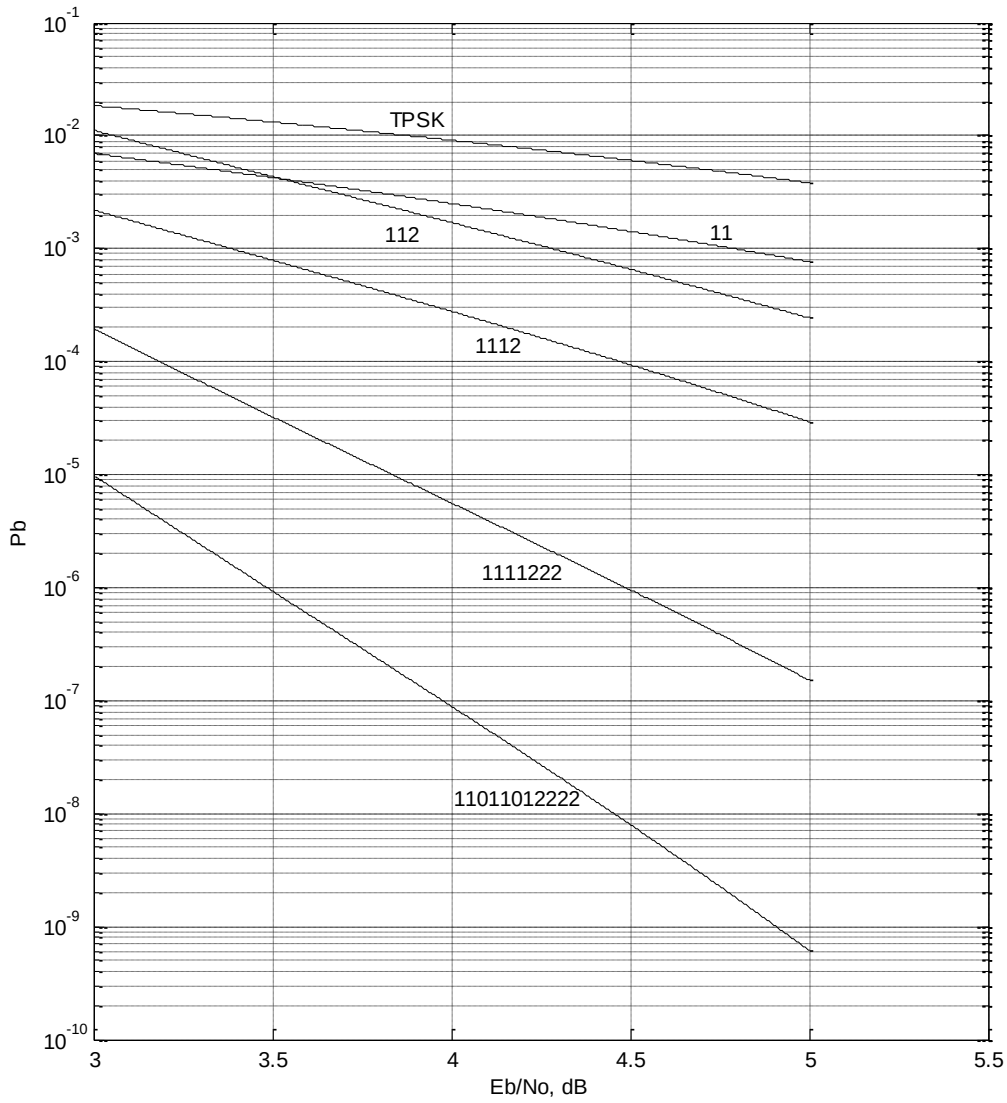
ორობით სიმეტრიული არხებისთვის ნაპოვნი ახალი ხვეულა კოდების მახასიათებლები ფასდებიან თავისუფალი ჰემინგის მანძილებით (იხ. ცხრილი 21); ხოლო ისეთი გაუსის არხებისათვის, სადაც გამოიყენებიან ორობითი (M=2) ფაზამოდულირებული სიგნალები (BPSK), სამობითი (M=3) ფაზამოდულირებული სიგნალები (TPSK) და ოთხობითი (M=4) ფაზამოდულირებული სიგნალები (QPSK) ეფექტურობის შესაფასებლად ვიყენებთ სიჩქარის γ მახასიათებელს და ბიტზე შეცდომის ალბათობათა (P_b) მახასიათებლებს. აქ $\gamma = R\log_2(M)$ ბიტი/სიმბოლო/ჰც; ხოლო $P_b = f(E_b/N_0)$, სადაც E_b/N_0 ერთი საინფორმაციო ბიტის შესაბამისი სიგნალის ენერგიისა და გაუსის ხმაურის ენერგიის თანაფარდობაა. იმის გათვალისწინებით, რომ ჩვენს შემთხვევებში $M=Q$, γ მახასიათებლისთვის გვაქვს: γ(BPSK)=0.5; γ(TPSK)=γ(QPSK)=1 (ნაიკვისტის სიჩქარე). შესაბამისად P_b-ს ზედა ადიტიური საზღვრებისათვის [40]:

$$P_b \leq (1/\log_2(M)) \sum_{i=1}^{n} (a_i) Q\{d^2/2N_0\};$$

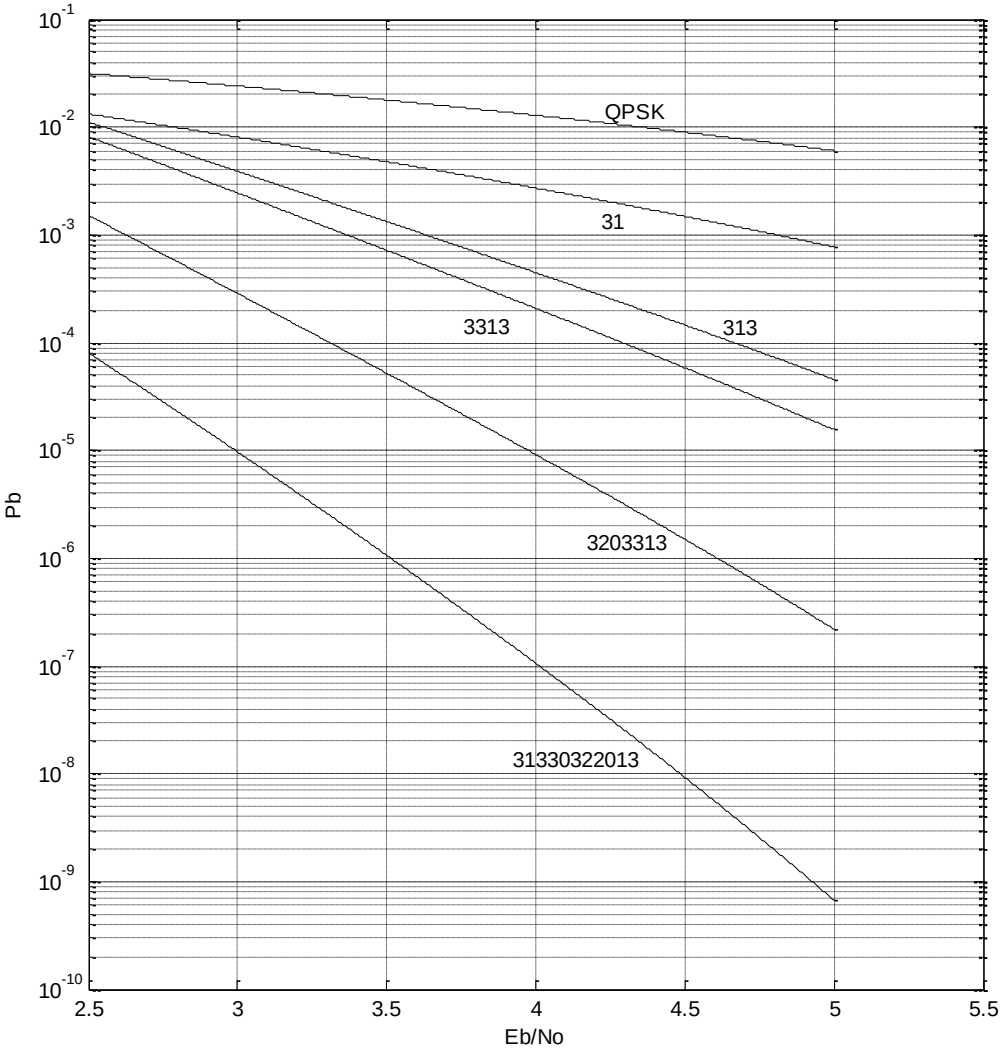
ზემოთ მოყვანილის და იმის გათვალისწინებით, რომ n=21 და $Q(x)$=0.5erfc(x/$\sqrt{2}$), გვექნება ქვემოთ, ნახ. 24-26-ზე მოყვანილი შედეგები.



ნახ. 24. ალბათური მახასიათებლები კოდირებული BPSK-თვის



ნახ. 25. ალბათური მახასიათებლები კოდირებული TPSK-თვის



ნახ. 26. ალბათური მახასიათებლები კოდირებული QPSK-თვის

3.5 დასკვნები

მოყვანილი თავის დასაწყისში ჩამოყალიბებულია მეთოდი, რომელიც საკმარისია მანძილის მიმართ ინვარიანტული კოდებისა და სიგნალ-კოდური სისტემების ასაგებად. დეიქსტრის ალგორითმის გამოყენებით დამუშავებული პროგრამის საშუალებით განხორციელებულია კომპიუტერული ძებნა ახალი, ალფაბეტური სიჭარბის მქონე მანძილის მიმართ ინვარიანტული, კოდების და ნაპოვნი კოდები ტაბულირებულია. ნაჩვენებია, რომ მოყვანილი ოთხობითი კოდები შეიძლება გამოყენებული იქნან როგორც ორობით სიმეტრიული არხებისთვის, ასევე გაუსის არხებისათვის ორობითი და ოთხობითი ფაზამოდულირებული სიგნალებით. აგებულია სამობითი სიგნალ-კოდური სისტემები სიმპლექსური სიგნალების გამოყენებით. მოყვანილია სისტემათა სიჩქარისა და ალბათური მახასიათებლები, საიდანაც ჩანს, რომ ნაიკვისტის სიჩქარეზე აგებული სისტემებისათვის ენერგეტიკული მოგება არაკოდირებულ ორობით ფაზამოდულირებულ სიგნალთან შედარებით შეადგენს: შეცდომის ალბათობაზე 10^{-5} 6 დბ-ს, ხოლო 10^{-4} -ზე 5.5 დბ-ს [40, 42].

დასასრულს შევნიშნავთ, რომ მოყვანილი მიდგომებისა და მეთოდების გამოყენებით სავსებით შესაძლებელია ანალოგიური სისტემების აგება სხვა სიგნალების (მაგ. იხ. [13, 40, 43] ბაზაზეც.

საბოლოო დასკვნა. მიღებული შედეგები და რეკომენდაციები

სადისერტაციო ნაშრომში მიღებულია შემდეგი შედეგები:

1. ნაჩვენებია, რომ ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდები წარმოადგენენ უწყვეტი კოდების უფრო მაღალ საფეხურს, ვიდრე ცნობილი კლასიკური ხვეულა კოდები.

2. მოყვანილია გალუას ველთა იმ არითმეტიკის ნაწილი, რომელიც საჭიროა ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ასაგებად. წარმოდგენილია შესაბამისი ცხრილები.

3. წარმოდგენილია ალფაბეტურ-სიმბოლური სიჭარბის მქონე ხვეულა კოდების ძებნის და მისი პარამეტრების განსაზღვრის ალგორითმები შესაბამისი პროგრამული რეალიზაციებით.

4. წარმოდგენილია მანძილის მიმართ ინვარიანტული ხვეულა კოდების აგების მეთოდი.

5. ნაპოვნია მაღალეფექტური ახალი ხვეულა კოდები და მოყვანილია მათი მახასიათებლები.

ვინაიდან ახალი კოდები და სიგნალ-კოდური სისტემები არ ხასიათდებიან სიჩქარის მაღალი პარამეტრებით (ნაიკვისტის საზღვარი), მაგრამ აქვთ მაღალი ენერგეტიკული ეფექტურობა (5-6 დბ), ჩვენი რეკომენდაცია იქნება გამოყენებული იქნან ისინი თანამგზავრულ და განსაკუთრებით შორეული კოსმოსური კავშირის სისტემებში.

ბიბლიოგრაფია

1. Трахтенброт Б. А., Барздинь Я. М. Конечные автоматы. М.: „Наука”, 1970.

2. Захаров В. Н. Автоматы с распределенной памятью. М.: „Энергия”, 1975.

3. Huffman D. A. The Synthesis of Linear Sequential Coding Networks. Symposium on „Information Theory”. London, September 12-16, 1955.
   არსებობს თარგმანი რუსულ ენაზე:
   Хаффмен Д. А. Синтез линейных многотактных кодирующих схем. В. кн.: Теория передачи сообщении. М.: ИИЛ, 1957.

4. მათემატიკა ინჟინრებისათვის. თარგმანი ინგლისურიდან (გ. ჯეიმზის რედაქცია) დ. ნატროშვილისა და ო. ზუმბურიძის რედაქციით. თბილისი, „გლობალ-პრინტი”, 2001.

5. Банкет В.Л., Голощапов В.А., Ляхов А.И. Техника декодирования сверточных кодов. Зарубежная радиоэлектроника, 1983, N 2, с. 3-27.

6. Forney G.D., Jr. Convolutional Codes I: Algebraic Structure. IEEE Trans. Inform. Theory. 1970, v. IT-16, №6, pp. 720-738.

7. Зяблов В.В., Шавгулидзе С.А. Обобщенные каскадные помехоустойчивые конструкции на базе сверточных кодов. М.: „Наука”, 1991.

8. Heller J.A. Sequential Decoding: Short constraint length Convolutional Codes. Jet Propusion Lab., California Inst. Technol., Pasadena, Space Program Summary 37-54, v.3, Dec. 1968, pp. 171-174.

9. Daut D.G., Modestino J.W., Wismer L.D. New Short Constraint Length Convolutional Code Constructions for Selected Rational Rates. IEEE Trans. Inform. Theory. 1982, v.IT-28, №5, pp. 794-800.

10. Massey J.T, Sain M.K. Inverses of Linear Sequential Circuits. IEEE Trans. Comp. 1968, v. C-17, №4, pp. 330-337.

11. Viterbi A.J. Error Bound for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Trans. Inform. Theory.1967, v. IT-13, №2, pp. 260-269.

12. ნ. უღელიძე, თ. ვიკვინია, თ. ქამხაძე, ე. ურუშაძე. დაბრკოლებები ინფორმაციის გადაცემის სისტემებში. ხმაური. საერთაშორისო სამეცნიერო ჟურნალი “ინტელექტი”. 2012, №1 (42), გვ. 129-132.

13. Sklar B. Digital Communications. Prentice Hall, N.J., 2001.
   არსებობს თარგმანი რუსულ ენაზე:
   Скляр Б. Цифровая связь. М.: „Вильямс”, 2003.

14. Paaske E. Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4 . IEEE Trans. Inform. Theory. 1974, v. IT-20, № 5, pp. 683-689.

15. Lipski W. Kombinatoryka Dla Programistow. Wydawhictwa Naukowo-Techniczne, Warszawa, 1982.
არსებობს თარგმანი რუსულ ენაზე:
Липский В. Комбинаторика для програмистов. М.: „Мир", 1988.

16. Minieka E. Optimization Algorithms for Networks and Graphs. Marcel Dekker Inc., N. Y., 1978.
არსებობს თარგმანი რუსულ ენაზე:
Майника Э. Алгоритмы оптимизации на сетях и графах. М.: „Мир", 1981.

17. Abrahams J. R., Coverly G. P. Signal Flow Analysis. Pergamon Press, Oxford, 1965.

18. Гантмахер Ф. Р. Теория матриц. М.: „Наука", 1988.

19. ნ. უღრელიძე. გისოსისებური კოდების აგება უწყვეტფაზიანი სიხშირე მოდულირებული სიგნალების ბაზაზე. სადოქტორო დისერტაცია. თბილისი, სტუ, 1994.

20. Питерсон У., Уэлдон Э. Коды исправляющие ошибки. М.: Мир, 1976.

21. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов исправляющих ошибки. М.: Связь, 1979.

22. Морелос-Сарагоса Р. Искуство помехоустойчивого кодирования. М.: Техносфера, 2006.

23. Lee C. Y. Some Properties of Nonbinary Error-Correcting Codes. IRE Trans., v. IT-4, 1958, pp.77-82.

24. Бородин Л.Ф. Введение в теорию помехоустойчивого кодирования. М.: Советское радио, 1968.

25. Угрелидзе Н. А. Новый класс сверточных кодов. Российская Академия Наук. Сборник докладов. Фев., 19-27, 2009, с. 17-19.

26. ნ. უღრელიძე. სიმბოლურ-ალფაბეტური სიჭარბის კოდის ერთი რეალიზაციის შესახებ. „ინტელექტი". თბილისი, აპრილი, 2009, გვ. 131-133.

27. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа. М.: Наука, 1989.

28. Зюко А. Г. и др. Помехоустойчивость и эффективность систем передачи информации. М.: Радио и связь , 1985.

29. Ugrelidze N.A., Shavgulidze S.A., Asanidze I.G. Convolutional Codes Over GF(4) for 4-ary Distance –Invariant CPFSK Signalling. Electronics Letters, V. 29, June, N 12, 1993, p.1104.

30. Bossert M. Channel Coding for Telecommunicatins. Jon Wiley&Sons, N.Y., 1999.

31. Shavgulidze S. A., Chachua L., Ugrelidze N.A. Concordand and Partially Concordant Ring Convolutional Codes for CPFSK. In Proceedings of 1997 International Symposium on Infor-mation Theory. Ulm, Germany, June 29 – July 4, 1997, p. 168.

32. Berlekamp E.R. Algebraic Coding Theory. McGraw-Hill, N.Y., 1968.
არსებობს თარგმანი რუსულ ენაზე:
Берлекэмп Э. Алгебрайческая теория кодирования. М.: „Мир'', 1971.

33. Blake I. F. Mullin R. C. The Mathematical Theory of Coding. Academic Press, N. Y., 1975.

34. Lidl R., Niederreiter H. Finite Fields. Addison-Wesley, Massachusetts, 1983.
არსებობს თარგმანი რუსულ ენაზე:
Лидл Р., Нидеррайтер Г. Конечные поля. Т. I-II. М.: „Мир'', 1988.

35. Lin S., Costello D.J. Jr. Error Control Coding: Fundamentals and Applications. Prentice-Hall, Englewood Cliffs, N. J., 1983.

36. Peterson W.W., Weldon E.J., Jr. Error-Correcting Codes. The MIT Press, Cambridge, 1972.
არსებობს თარგმანი რუსულ ენაზე:
Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. М.: „Мир'', 1976.

37. Фаддеев Д.К., Соминский И.С. Алгебра. М.: „Наука'', 1964.

38. Швецов К. И., Бевз Г. П. Справочник по элементорной математике. Киев, „Наукова Думка", 1965.

39. ნ. უღრელიძე, მ. სორდია, თ. კვიკვინია. ლის მეტრიკის გამოყენება ინვარიანტული სისტემების ასაგებად. საქართველოს ტექნიკური უნივერსიტეტის შრომები. №1, თბილისი, 2015 (მიღებულია დასაბეჭდად).

40. Proakis J.G. Digital Communications. Mc Graw-Hill, N.Y., 1995.
არსებობს თარგმანი რუსულ ენაზე:
Прокис Дж. Цифровая связь. М.: „Радио и связь», 2000.

41. უღრელიძე ნ. ა., კვიკვინია თ. ნ. კოდური კონსტრუქციები სიმპლექსური სიგნალების ბაზაზე. საქართველოს საინჟინრი სიახლენი. 2014, ტ. 72, №4, გვ.15-18.

42. კვიკვინია თ. ნ., ურუშაძე ე. ა. ახალი ალფაბეტური სიჭარბის მქონე კოდები. GESJ: Computer Sciences and Telecommunications // 2014 | No. 4(44), pp. 44-52 (http://gesj.internet-academy.org.ge/download.php?id=2360.pdf).

43. Ugrelidze N.A., Kvikvinia T. K., Kamkhadze T. I., Urushadze E.A. Multi-Amplitude Minimum Shift Keying Signals Designing. IEEE 11-th International Sympozium on Electronics and Telecommunications (ISETC ‘14). Proceedings. Timisoara, Romania, November 14-15, 2014, pp.123-126.

დანართი 1. საუკეთესო კოდების ძებნის პროგრამა

```
%                              1 infut 1 output
clc;
clear;
%----------------------I. SACKISI MONACEMEBI---------------------------
Dp=2;             % Uaxloesi mokle kodis tavisufali mandzili
q=2;              % Coderis shesasvleli alphabetis zoma  (2 an 4)
Q=4;              % Coderis gamosasvleli alphabetis zoma (4 an 8)
niu=2;            % Dakovnebis elementebis raodenoba codershi
Kalora=27351;     % Grafis tsveros gaferadeba

% -----------II.CODERIS SHESADZLO MDGOMAREOBEBIS GANSAZGVRA-------------
R=(log2(q))/(log2(Q));
S=q^(niu);
K=niu+1;
Vs=zeros(S+1,niu);
   for j=1:1:niu
       i=1;
       t=1;
         for i=1:1:S
               Vs(i,j)=mod((Vs(i,j)+(t-1)),q);
                     a=q^(j-1);
                    if mod(i,a)==0
                       t=t+1;
                    end;
         end;
   end;
   Vs;

 %----------------------III.ALGEBRULI OPERACIEBI GF-ze------------------
                % Namravli GF(4)-ze
M4(1,1)=0; M4(1,2)=0; M4(1,3)=0; M4(1,4)=0;
M4(2,1)=0; M4(2,2)=1; M4(2,3)=2; M4(2,4)=3;
M4(3,1)=0; M4(3,2)=2; M4(3,3)=3; M4(3,4)=1;
M4(4,1)=0; M4(4,2)=3; M4(4,3)=1; M4(4,4)=2;
                % Jami GF(4)-ze
S4(1,1)=0; S4(1,2)=1; S4(1,3)=2; S4(1,4)=3;
S4(2,1)=1; S4(2,2)=0; S4(2,3)=3; S4(2,4)=2;
S4(3,1)=2; S4(3,2)=3; S4(3,3)=0; S4(3,4)=1;
S4(4,1)=3; S4(4,2)=2; S4(4,3)=1; S4(4,4)=0;
                % Namravli GF(8)-ze
M8(1,1)=0; M8(1,2)=0; M8(1,3)=0; M8(1,4)=0; M8(1,5)=0; M8(1,6)=0;
M8(1,7)=0; M8(1,8)=0;
M8(2,1)=0; M8(2,2)=1; M8(2,3)=2; M8(2,4)=3; M8(2,5)=4; M8(2,6)=5;
M8(2,7)=6; M8(2,8)=7;
M8(3,1)=0; M8(3,2)=2; M8(3,3)=3; M8(3,4)=4; M8(3,5)=5; M8(3,6)=6;
M8(3,7)=7; M8(3,8)=1;
M8(4,1)=0; M8(4,2)=3; M8(4,3)=4; M8(4,4)=5; M8(4,5)=6; M8(4,6)=7;
M8(4,7)=1; M8(4,8)=2;
M8(5,1)=0; M8(5,2)=4; M8(5,3)=5; M8(5,4)=6; M8(5,5)=7; M8(5,6)=1;
M8(5,7)=2; M8(5,8)=3;
```

```
M8(6,1)=0; M8(6,2)=5; M8(6,3)=6; M8(6,4)=7; M8(6,5)=1; M8(6,6)=2;
M8(6,7)=3; M8(6,8)=4;
M8(7,1)=0; M8(7,2)=6; M8(7,3)=7; M8(7,4)=1; M8(7,5)=2; M8(7,6)=3;
M8(7,7)=4; M8(7,8)=5;
M8(8,1)=0; M8(8,2)=7; M8(8,3)=1; M8(8,4)=2; M8(8,5)=3; M8(8,6)=4;
M8(8,7)=5; M8(8,8)=6;
                % Jami GF(8)-ze
S8(1,1)=0; S8(1,2)=1; S8(1,3)=2; S8(1,4)=3; S8(1,5)=4; S8(1,6)=5;
S8(1,7)=6; S8(1,8)=7;
S8(2,1)=1; S8(2,2)=0; S8(2,3)=4; S8(2,4)=7; S8(2,5)=2; S8(2,6)=6;
S8(2,7)=5; S8(2,8)=3;
S8(3,1)=2; S8(3,2)=4; S8(3,3)=0; S8(3,4)=5; S8(3,5)=1; S8(3,6)=3;
S8(3,7)=7; S8(3,8)=6;
S8(4,1)=3; S8(4,2)=7; S8(4,3)=5; S8(4,4)=0; S8(4,5)=6; S8(4,6)=2;
S8(4,7)=4; S8(4,8)=1;
S8(5,1)=4; S8(5,2)=2; S8(5,3)=1; S8(5,4)=6; S8(5,5)=0; S8(5,6)=7;
S8(5,7)=3; S8(5,8)=5;
S8(6,1)=5; S8(6,2)=6; S8(6,3)=3; S8(6,4)=2; S8(6,5)=7; S8(6,6)=0;
S8(6,7)=1; S8(6,8)=4;
S8(7,1)=6; S8(7,2)=5; S8(7,3)=7; S8(7,4)=4; S8(7,5)=3; S8(7,6)=1;
S8(7,7)=0; S8(7,8)=2;
S8(8,1)=7; S8(8,2)=3; S8(8,3)=6; S8(8,4)=1; S8(8,5)=5; S8(8,6)=4;
S8(8,7)=2; S8(8,8)=0;

%------IV.CODTA GENERATOREBIS FORMIREBA  ZOGIERTI KODEBIS GAMORICXVIT---
K=niu+1;
Nx=Q^K;
Vk=zeros(Nx,K);
   for j=1:1:K
       t=1;
         for i=1:1:Nx
             Vk(i,j)=mod((Vk(i,j)+(t-1)),Q);
                    a=Q^(j-1);
                 if mod(i,a)==0
                    t=t+1;
                 end;
         end;
   end;
   Vk

   %%---------------Katastrofuli koderebis gamoricxva--------------
     x=0;
     for i=1:1:Nx
              Xs=0;
             for j=1:1:K
                if Q==4
                Xs=S4((Xs+1),(Vk(i,j))+1);
                else
                Xs=S8((Xs+1),(Vk(i,j))+1);
                end;
             end;
                       if Xs~=0
                          x=x+1;
                       end;
     end;
```

```
  Vx=zeros(x,K);
  p=0;
  for i=1:1:Nx
          Xs=0;
         for j=1:1:K
            if Q==4
            Xs=S4((Xs+1),(Vk(i,j))+1);
            else
            Xs=S8((Xs+1),(Vk(i,j))+1);
            end;
         end;

                  if Xs~=0
                     p=p+1;
                     Vx([p],:)=Vk([i],:);
                  end;
  end;
  Vx;
  Nx=p;

%%-------------Kodta gamoricxva striqonuli mandzilit------------
c=0;
for i=1:1:Nx
              if mod(i,1000)==0
                 Stepi=i;
              end;
    Ds=0;
    for j=1:1:K
                                  if Q==4
                                         if Vx(i,j)==0
                                            Vl(j)=0;
                                         end;
                                         if Vx(i,j)==1
                                            Vl(j)=1;
                                         end;
                                         if Vx(i,j)==2
                                            Vl(j)=1;
                                         end;
                                         if Vx(i,j)==3
                                            Vl(j)=2;
                                         end;
                                  else
                                         if Vx(i,j)==0
                                            Vl(j)=0;
                                         end;
                                         if Vx(i,j)==1
                                            Vl(j)=1;
                                         end;
                                         if Vx(i,j)==2
                                            Vl(j)=1;
                                         end;
                                         if Vx(i,j)==3
                                            Vl(j)=1;
                                         end;
                                         if Vx(i,j)==4
                                            Vl(j)=2;
```

```
                                        end;
                                        if Vx(i,j)==5
                                           Vl(j)=2;
                                        end;
                                        if Vx(i,j)==6
                                           Vl(j)=3;
                                        end;
                                        if Vx(i,j)==7
                                           Vl(j)=2;
                                        end;
                                end;
                    Ds=Ds+Vl(j);
        end;

                    if Ds>=Dp
                       c=c+1;
                       Vp([c],:)=Vx([i],:);
                    end;
    end;
    Vp;
    Nx=c;

%-V.GADASVLEBIS DA SHESABAMISI TSONEBIS GANSAZGVRA DINAMIKASHI DA
DIJKSTRA-s ALGORITMI-
      f=0;
      for x=1:1:Nx
                if mod(x,1000)==0
                   StepX=x;
                end;
                        X=Vp([x],:);
                        for j=1:1:S+1
                            W(j)=Inf;
                            MIN=0;
                            W1(j)=0;
                            W2(j)=Inf;
                        end;
                   N=1;
              while W(S+1)~=MIN
                   i=N;
                   W(i)=Kalora;
                   S1=Vs([i],:);
                   for j=2:1:(S+1);
                       S2=Vs([j],:);
                                        b=0;
                                        for t=1:1:(niu-1)
                                            A(t)=S1(t)-S2(t+1);
                                            b=b+abs(A(t));
                                        end;
                            if b==0
                                        V=0;



                                    if Q==4
```

```
                                        for l=1:1:niu
Vo(l)=M4((X(l)+1),(S2(l)+1));
                                            V=S4((V+1),(Vo(l)+1));
                                        end;
V=S4((V+1),((M4((X(niu+1)+1),(S1(niu)+1)))+1));
                                    else
                                        for l=1:1:niu
Vo(l)=M8((X(l)+1),(S2(l)+1));
                                            V=S8((V+1),(Vo(l)+1));
                                        end;
V=S8((V+1),((M8((X(niu+1)+1),(S1(niu)+1)))+1));
                                    end;
                                      if Q==4
                                            if V==0
                                               Vw=0;
                                            end;
                                            if V==1
                                               Vw=1;
                                            end;
                                            if V==2
                                               Vw=1;
                                            end;
                                            if V==3
                                               Vw=2;
                                            end;
                                      else
                                            if V==0
                                               Vw=0;
                                            end;
                                            if V==1
                                               Vw=1;
                                            end;
                                            if V==2
                                               Vw=1;
                                            end;
                                            if V==3
                                               Vw=1;
                                            end;
                                            if V==4
                                               Vw=2;
                                            end;
                                            if V==5
                                               Vw=2;
                                            end;
                                            if V==6
                                               Vw=3;
                                            end;
                                            if V==7
                                               Vw=2;
                                            end;
                                      end;
                                      d=Vw;
                                else
```

```
                            d=Inf;
                          end;



                                                  if i==1
                                                     if j==S+1
                                                        d=Inf;
                                                     end;
                                                  end;

    %%------------------ Dijkstra-s algoritmi----------------------
                                 if W1(j)==Kalora
                                    W(j)=W1(j);
                                 else
                                    W(j)=min(W2(j),MIN+d);
                                 end;
                       end;
                                     MIN=min(W);
                            if W(S+1)==MIN
                                    f=f+1;
                            Dcfree(f)=MIN;
                            else
                                      c=1;
                                      while W(c)~=MIN
                                            c=c+1;
                                            N=c;
                                      end;
                                            W1(1)=Kalora;
                                            for j=2:1:S+1
                                               if j==N
                                                  W(j)=Kalora;
                                               end;
                                                 W2(j)=W(j);
                                                 W1(j)=W(j);
                                            end;
                            end;
                  end;
      end;
Dfree=max(Dcfree)

%----------------------VI.NAPOVNI SAUKETESO KODEBI---------------------
j=0;
Vc=zeros(Nx,K);
for i=1:1:Nx
    if Dcfree(i)==Dfree
       j=j+1;
    end;
end;
     Nc=j;
     Vc=zeros(j,K);
     Vo=zeros(j,K);
     j=0;
     for i=1:1:Nx
        if Dcfree(i)==Dfree
```

```
            j=j+1;
            Vc([j],:)=Vp([i],:);
            Vo([j],:)=Vp([i],:);
         end;
      end;
      Dfree;

      %%-----Napovni kodebidan inversiuli koderebis gamoricxva------
      Vxo=zeros(Nc,K);
      Zx=zeros(Nc,K);
      for i=1:1:Nc
           V_c([i],:)=fliplr(Vc([i],:));
           for j=(i+1):1:Nc
               k=isequal(V_c([i],:),Vc([j],:));
                   if k==0
                      Vo([j],:)=Vo([j],:);
                   else
                      Vo([j],:)=0;
                   end;
           end;
      end;
      Vo;
      j=0;
      for i=1:1:Nc
          k=isequal(Vo([i],:),Zx([i],:));
          if k==0
             j=j+1;
             Vxo([j],:)=Vo([i],:);
          end;
      end;
           Vk=zeros(j,K);
           for i=1:1:j
               Vk([i],:)=Vxo([i],:);
           end;
       Codes=Vk
     % save q;
     % save Q;
      %save niu;
      %save Dfree;
      save Vk3;
      Nx=j;

%%-----------------------------------------------------------------
    ts=cputime;
    tm=ts/60;
    th=tm/60;
```

დანართი 2. შეცდომის ალბათობის გამოთვლის პროგრამა

```
%------------SHETSDOMIS  ALBATOBIS ANGARISHI codi+MPSK-istvis----------
clc;
clear;
%---------------------------SACKISI MONACEMEBI-------------------------
M=4;              % Signalebis raodenoba
SNRmin=2.5;       % Minimaluri tanafardoba signali/xmaurtan bitze dB-shi
SNRmax=5;         % Maximaluri tanafardoba signali/xmaurtan bitze dB-shi
delta=0.001;      % Tanafardoba signali/xmaurtan cvlilebis biji
q=2;              % Koderis shesasvleli simboloebis alfabetis zoma
Q=4;              % Koderis gamosasvleli simboloebis alfabetis zoma
%-----------------------Codis mandzilta speqtri-------------------------
d=[1 2];            % Kodis Hemingis mandzilis speqtri
a=[2 1];            % Bitebis raodenoba mocemul mandzilze
%------------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK-tvis------------
k=1/(log2(M));
R=1;
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
               for j=1:1:La
                   SNR=10^(0.1*SNRdB);
                   X=sqrt((2*d(j))*SNR);
                   Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
               end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig1(i)=P(i);
end;
%----------------------------- Code=31 ---------------------------------
% Kodis Hemingis mandzilis speqtri
d=[3 4 5 6 7 8 9 10 11 12 13];
% Bitebis raodenoba mocemul mandzilze
a=[1 2 3 4 5 6 7 8 9 10 11];
%------------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK+31 -------------
R=log2(q)/log2(Q);
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
               for j=1:1:La
                   SNR=10^(0.1*SNRdB);
                   X=sqrt((2*R*d(j))*SNR);
                   Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
               end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig2(i)=P(i);
end;
%----------------------------- Code=313 --------------------------------
% Kodis Hemingis mandzilis speqtri
```

```
d=[5 6 7 8 9 10 11 12 13 14 15];
% Bitebis raodenoba mocemul mandzilze
a=[1 4 12 32 80 192 448 1024 2304 5120 11264];
%-----------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK+31 ------------
R=log2(q)/log2(Q);
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
              for j=1:1:La
                  SNR=10^(0.1*SNRdB);
                  X=sqrt((2*R*d(j))*SNR);
                  Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
              end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig3(i)=P(i);
end;
%---------------------------- Code=3313 -------------------------------
% Kodis Hemingis mandzilis speqtri
d=[6 7 8 9 10 11 12 13 14 15 16];
% Bitebis raodenoba mocemul mandzilze
a=[2 7 18 49 130 333 836 2069 5060 12255 29444];
%-----------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK+31 -------------
R=log2(q)/log2(Q);
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
              for j=1:1:La
                  SNR=10^(0.1*SNRdB);
                  X=sqrt((2*R*d(j))*SNR);
                  Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
              end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig4(i)=P(i);
end;
%---------------------------Code=3203313-------------------------------
% Kodis Hemingis mandzilis speqtri
d=[10 11 12 13 14 15 16 17 18 19 20];
% Bitebis raodenoba mocemul mandzilze
a=[36 0 211 0 1404 0 11633 0 77433 0 502690];
%-----------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK+31 -------------
R=log2(q)/log2(Q);
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
              for j=1:1:La
                  SNR=10^(0.1*SNRdB);
```

```
                X=sqrt((2*R*d(j))*SNR);
                Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
            end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig5(i)=P(i);
end;
%----------------------------Code=31330322013--------------------------
% Kodis Hemingis mandzilis speqtri
d=[14 15 16 17 18 19 20 21 22 23 24];
% Bitebis raodenoba mocemul mandzilze
a=[71 0 419 0 3383 0 23484 0 157716 0 1048620];
%-----------------SHETSDOMIS  ALBATOBIS ANGARISHI BPSK+31 -------------
R=log2(q)/log2(Q);
La=length(a);
i=0;
SNRdB=SNRmin-delta;
while SNRdB<SNRmax
      i=i+1;
      SNRdB=SNRdB+delta;
            for j=1:1:La
                SNR=10^(0.1*SNRdB);
                X=sqrt((2*R*d(j))*SNR);
                Pi(j)=(k*a(j)*(1/2)*erfc(X/(sqrt(2))));
            end;
SN(i)=SNRdB;
P(i)=sum(Pi);
Pfig6(i)=P(i);
end;
plot(SN,Pfig1,SN,Pfig2,SN,Pfig3,SN,Pfig4,SN,Pfig5,SN,Pfig6);
```