

KAUNO TECHNOLOGIJOS UNIVERSITETAS
Informatikos fakultetas



Duomenų perdavimo sistema, naudojant chaoso modelį

Atliko: Indrė Lešinskytė-Pakštienė, IFM-9/5
Data: 2010-01-22
Vadovas: E. Kazanavičius

KAUNAS, 2011

Turinys

Santrauka	3
Summary	4
Įvadas	5
Specifikacija	7
Analogiškų sistemų apžvalga	8
Kas ta chaoso teorija	15
Chaotinės sistemos savybės.....	18
Pagrindinės chaotinės sistemos problemos	22
Chaotinių sistemų matematinių lygčių pavyzdžiai	24
Aštuonių kanalų duomenų perdavimo sistema	27
Išvados ir pasiūlymai	34
Naudota literatūra:	35
PRIEDAS Nr. 1	37
PRIEDAS Nr. 2	38
PRIEDAS Nr. 3. Raktų generavimas	40
PRIEDAS Nr. 4. Programos kodas.....	41

Santrauka

Darbo metu analizuojamos esamos chaotinės sistemos. Pati chaoso teorija nėra ypatingai nauja, tačiau dėl pratiškai neįmanomo tikslaus šios teorijos aparatūrinio įgyvendinimo atsiranda didelė erdvė ieškoti tinkamiausio šios teorijos pritaikymo aspekto kiekvienu konkrečiu atveju. Chaotiškai perduodant signalą susiduriama su daugybe problemų. Šiame darbe, kuriant aštuonių kanalų duomenų perdavimo sistemą siekiama rasti ir patikrinti geriausią duomenų perdavimo būdą naudojant minėtąją teoriją.

Darbe buvo remtasi keletos skirtingų autorių parašytais matematinio modeliavimo paketu MatLab skriptais.

Summary

Job analysis of the existing systems Chaotic. Chaos theory itself is not particularly new, but it is impossible to become in practice the theory of the hardware generates a large search space of the appropriate application of the theory aspect of each case. Chaotic transmission signal faces many problems. In this work, creating an eight-channel data transmission system to locate and verify the best data transfer method using the above-mentioned theory.

The work was based on several different authors, written in a mathematical simulation package MATLAB scripts.

Išvadas

Netvarkingumą, arba kitaip vadinamą chaosą galima stebėti daugelyje gamtinių sistemų, pavyzdžiui, oro. Tokias sistemas tampa įmanoma paaiškinti taikant ir analizuojant chaoso teorijos matematinį modelį.

Chaoso teorija taikoma matematikos, fizikos srityse, taip pat apima filosofijos studijas bei dinamines sistemas, kurios yra labai jautrios pradinėms sąlygoms. Toks jautrumas vadinamas drugelio efektu. Maži pradinių sąlygų skirtumai (pavyzdžiui, dėl apvalinimo klaidų skaičiaus apskaičiavimas) gali lemti labai skirtingus Chaotinių sistemų rezultatus. Ilgalaikio prognozavimo taikyti apskritai neįmanoma, nes rezultatai kiekvieną kartą gali kardinaliai skirtis nuo prognozuotų, o bandymas prognozuoti tam tikrus rezultatus greičiau yra bandymas juos atspėti, nes mėginimas apskaičiuoti ir pagrįsti pagal seniau turėtus duomenis kokie tie rezultatai turėtų būti gauti, dažniausiai yra gautas bevertis ir neturintis reikšmės rezultatas.

Bendrai žodis „chaosas“ reiškia betvarkės būseną, bet būdvardis chaotiškas yra tiksliau apibrėžtas chaoso teorijoje, nors nėra vienintelio matematinio apibrėžimo. Bendrai vartojamas apibrėžimas teigia, kad dinaminė sistema, klasifikuojama kaip chaotinė, turi turėti tokias savybes:

- Būti priklausoma nuo pradinių sąlygų;
- Turi būti skirtingos topologijos;
- Ir iširtos periodinės orbitos.

Kaip matome, iš mano trumpo pateikto chaoso teorijos aprašymo, chaoso teorija labai plačiai taikoma ir neturi konkrečios lygties.

Tokias sistemas tampa įmanoma paaiškinti taikant ir analizuojant chaoso teorijos matematinį modelį, kuris sudaromas pasitelkus statistiką: tam tikrą laiką duomenys yra fiksuojami, vėliau gauti duomenys įvairiais būdais pagal tam tikras metodikas apdorojami ir gaunami rezultatai dažniausiai pateikiami vaizdžiai – diagramomis.

Mano tiriamojo darbo tikslas yra:

Sudaryti 8 radijo kanalų duomenų mainų sistemą, kur kiekvienas kanalas būtų koduojamas chaotiniu signalu.

Uždaviniai:

1. Ištyrinėti, kas yra chaoso modelis;
2. Kaip chaoso modelis yra naudojamas duomenų perdavimui;
3. Kokie chaoso modeliu pagrįsti duomenų perdavimo būdai yra sukurti;
4. Suprojektuoti savo chaoso modeliu grįstą duomenų perdavimo sistemą;

Pagrindinė darbo metodika bus konstruktyvus tyrimas. Bus nagrinėjama visa internete pateikta medžiaga ir darbo vadovo rekomenduota medžiaga. Tyrimo metu bus kaupiamos žinios, kurios vėliau bus panaudotos mano projektuojamam chaoso modeliu grįstam duomenų perdavimui. Tyrimą atliksiu literatūros analizės ir apibendrinimo metodu.

Specifikacija

Užduotis. Sukurti aštuonių kanalų duomenų perdavimo sistemą.

Matematinį modelį aprašome MatLab matematiniu paketu. Modelyje slankaus kablelio skaičius keičiame fiksuoto kablelio skaičiais, nes tokia bus aparatūrinė realizacija. Modelį testuojame su parinktais duomenimis.

Analogiškų sistemų apžvalga

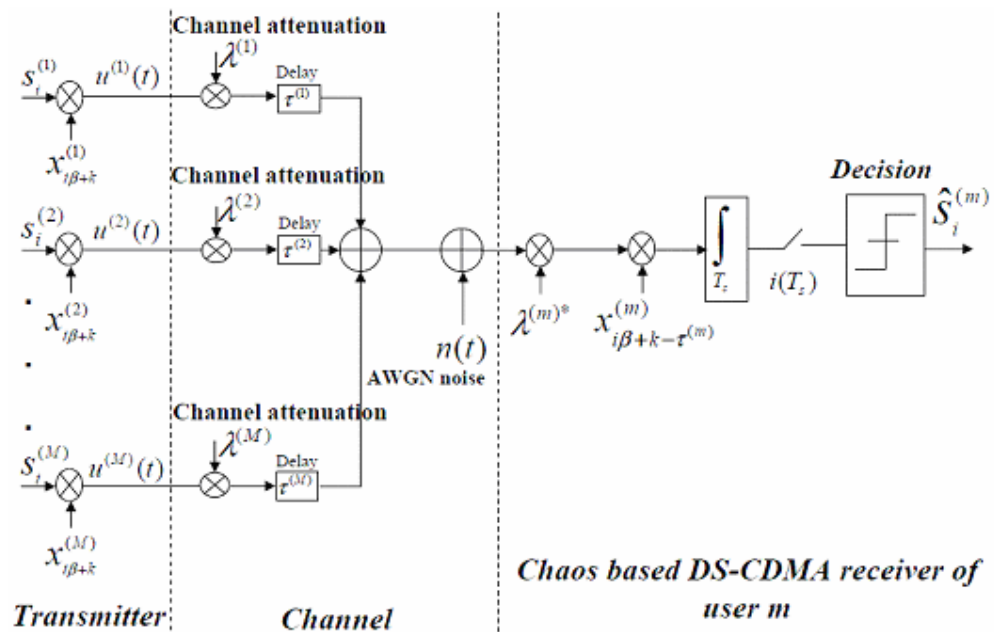
Chaos teorija buvo taikoma skirtingose srityse ir sistemose. Šiame skyriuje pristatysiu keletą jų.

Pirmasis chaoso teorijos taikymo pavyzdys apima dviejų bitų moduliacijos ir demoduliacijos schemas.

Chaotinė seka neperiodinė. Signalai generuojami iš netiesinės dinaminės sistemos. Šie signalai yra jautrūs pradinėms sąlygoms ir sunku prognozuoti jų elgesį ateityje iš praeities signalų stebėjimo. Taip pat chaotiškai seka gali būti gaunama iš paprasto modelio. Šios savybės yra privalumai naudojant chaotines sekas komunikacijos sistemose. Taigi, chaoso rodiklių naudojimas yra svarbus kai norima pagerinti Chaotines bendravimo sistemas.

Anksčiau siųstuvai moduliavo 1-bitų informacijos chaotinę perkėlimą automatinį perjungimą. Siūloma schema moduliuoja 2-bitų informacijos iki Chaotinės sekos perduodančios pusės. Kadangi kito chaotinio signalo mėginiai gaunami pagal chaotinę dinamiką, imtuvai gali demoduliuoti duomenis naudodamas chaotinę dinamiką, kuri turi gaunamą signalą. Taigi, matome, kad tobulinant signalų siuntimo-priėmimo schemas taikoma chaoso teorija.

Kitas pavyzdys yra Chaoso pagrindu veikiančių asinchroninių daugiavartotojiškų bendravimo sistemų veikimas per M transliavimo kanalą. Pirmame paveiksle matome šio įrenginio supaprastintos pagrindinės juostos ekvivalentą, kuris pagrįstas chaoso DSCDMA su M-paskirstytu kanalu.



1. Pav. Supaprastintos pagrindinės juostos ekvivalentas, kuris

Kadangi mano tikslas ne konkreti schema, o kaip ir kur tarnauja chaoso teorija ir kaip ji taikoma, tai aukščiau pateiktos schemos veikimo nedetalizuosiu.

Kryžminės koreliacijos kodai yra didesni negu chaotiškos sekos mažo išplitimo faktorius, chaosas pagrįstas DS-CDMA pralenkia DS-SS-SSMA sistemą naudojant specifinius kodus. Kai plitimo veiksnys yra aukštas, dviejų sistemų veikimas yra labai panašus. Motyvacija naudoti chaotines sekas naudojant ne specifinį kodą, o sekas kaip susilpnėjusius perdavimą yra pateisinamas taikant aukštus apsaugos perdavimo mažo efektyvumo praradimo.

Kitas straipsnis tyrė našumą taikant chaoso teorijos pagrindą daugiavartotojinėje komunikavimo sistemoje.

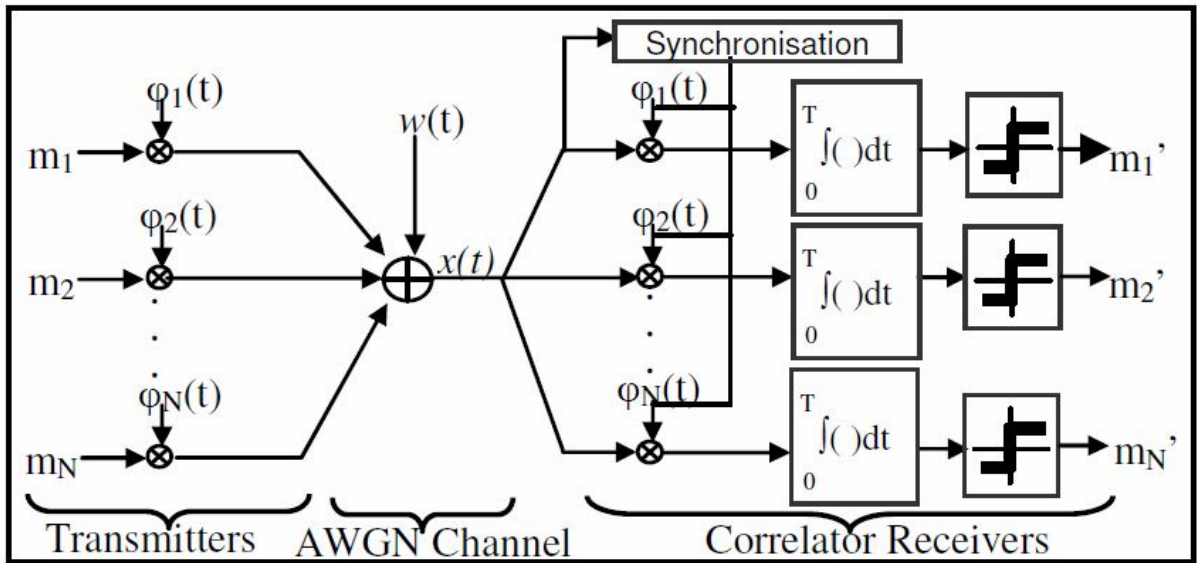
Ribotas tikslumas būdingas aparatinei platformai, pvz DSP reiškia, kad chaotiniai mėginiai bus apvalinami arba trupinami, nes jie yra generuojami. Toks begalinio tikslumo praradimas reiškia, kad bet kokia generuota chaotinė seka ilgainiui tampa periodinė ir taip praranda savo nepastovumą. Be to, realaus laiko įranga aplinkai nustato bitų spartos ir pajėgumo apribojimus daugiavartotojiškai CPSK sistemai, nes tik ribotas skaičius operacijų gali būti atliekamas per tam tikrą laikotarpį.

Esamas DSP realizavimo chaotines ryšių sistemas galima rasti skirtinguose literatūros šaltiniuose su aprašytomis pagrindinėmis savybėmis, o pagrindinė problema yra ta, kad visos sistemos yra vieno vartotojo sistemos.

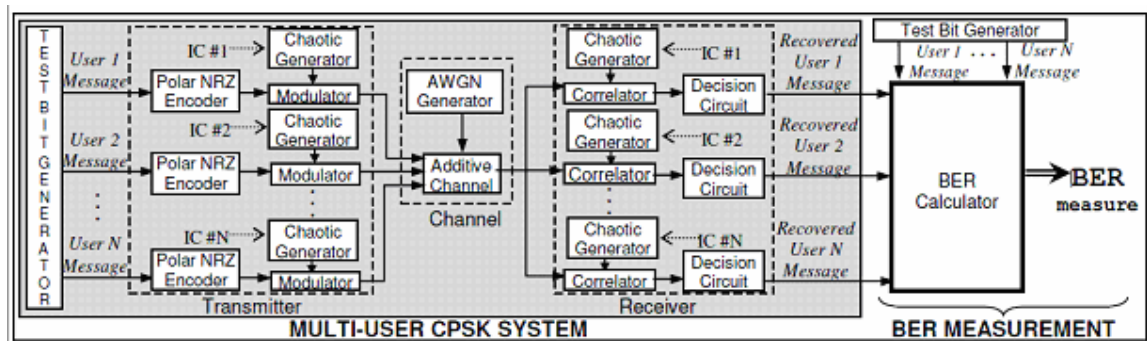
DSP procesorius buvo pasirinktas CPSK daugiavartotojiškos komunikacijos sistemos įgyvendimui ADSP-TS201S Tigeris SHARC 32 bitų plaukiančiojo taško procesorius iš analoginių įrenginių. Pasirinkto žodžio ilgis yra 32 bitai ir buvo pakankamai tikslus chaotiniam pritaikymui pritaikant keliems naudotojams. Plaukiojančio taško procesorius buvo pasirinktas, dėl to kad gali užtikrinti platų dinaminį diapazoną, kaip reikalaujama, kad būtų pasiektas tiek chaotiškumas tiek triukšmas generatoriumi DSP. Be to, nors ir aritmetinės operacijos paprastai atliekamos greičiau naudojant plaukiojančio taško pateikimą, bet ir reikalaujama papildomų operatorių valdant didėjantį greitį.

Maksimaliai išnaudojant CPSK sistemos pajėgumą, reikia kuo greitesnio procesoriaus. TigerSHARC turi didžiausią vykdymo greitį tarp konkuruojančių DSP procesorių, o procesoriaus taktinį dažnį 600MHz ir yra pajėgi atlikti 2400MIPS, 3600MFLOPS ir 1200MMACS. Šis prototipas buvo sukurtas naudojant TS201S EZKIT Lite vertinimo lenta TigerSHARC procesorius kartu su VisualDSP ++ plėtojimo aplinka. Iš CPSK sistemos funkcija yra nurodyta programinė įranga parašyta aukšto lygio C kalba ir perkeliama į perdirbėjo Asemblerio kalbą naudojant

VisualDSP ++. Kraunasi vykdomąjį kodą į DSP Derinimo programinę įrangą ir veikiant reikalingam modeliavimui buvo įjungta bendravimo VisualDSP ++ ant PC ir vertinimo sistema.



2. pav. Daugiavartotojiška DCMA skaitmeninė komunikavimo sistema.



3. Pav. Daugiavartotojiškos CPSK sistemos funkcinis blokas.

Daugiavartotojiška CPSK sistema, pateikta viršuje esančiame paveiksle buvo įgyvendintas DSP, kaip parodyta funkcinė blokinė diagrama. Reikėtų pažymėti, kad nors esminis synchronizacijos blokas (2 pav.) yra praktinės sistema, ji buvo ne šios dalies įgyvendinimą; tinkama synchronizavimo metu imtuvas.

Taikant BIR matavimą, analoginis komunikacijos kanalas imituojama viduje DSP naudojant AWGN generatorius. Todėl visą CPSK sistemos pavaizduotos trečiame paveiksle buvo įgyvendinta viename DSP luste.

Funkcinių blokų sistemos diagrama (3 pav.) yra įgyvendinama moduluojant kaip atskiras funkcijas C kalbos sistemos aprašyme. Poliarinis NRZ encoderis konvertuoja vartotojo žinutes (1,0) į tinkamą formatą (1, -1) į įėjimą į moduliatorių blokas, kuris yra tik daugiklis. Tiesiog paprastas kanalas sudeda savo įėjimo signalus. Kiekvienas individualus vartotojo pranešimas turi moduliuoti įvairias chaotines sekas, pagamintos iš unikalių pradinių sąlygų (IC). Tas pats IC turi būti

naudojamas ir imtuvo, kad iškoduotų pranešimą. Triukšmo papildoma dalis yra perduodama kartu su signalu, o kiekvienas koreliacijos procesas yra sudėtinis signalas.

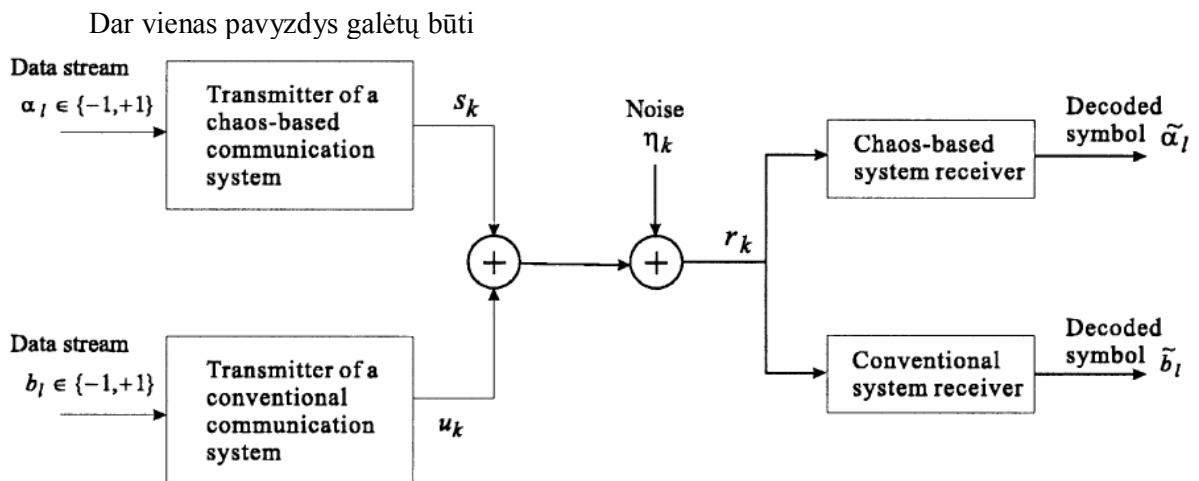
Chaotinis generatoriaus blokas yra pagrįstas žemėlapio logistika. Chaotinės sekos generuojamos ant DSP buvo išnagrinėtos nuo 1000 skirtingų pradinių sąlygų. Periodiškumo 2.240 mėginiai buvo beveik visada nepastebėti. Tai lemia kiekvienos sekos galutinį vertingumą 0,9996157.

Sekant tai, chaotinė seka įeina tuo pačiu 2240-imtųjų ilgai kartoto modelio. Unikalus eilės ilgis generuoja iki šio modelio yra interesų pasiekimo. Buvo pastebėta, kad daugiau nei ketvirtadalis pradinių sąlygų iširti gaminti unikalus seka per 5.800 mėginių ilgio, su ilgiausia tokių yra 8.275 pavyzdžių.

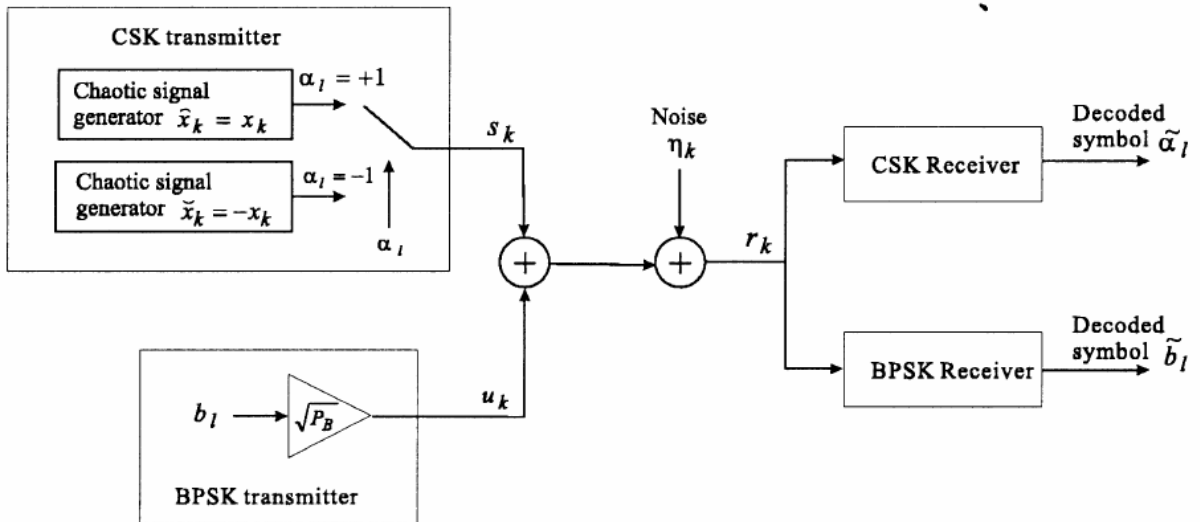
Užtikrinant geras koreliacijos savybes, kiekviena chaotinė seka, kuri naudojama, turi būti unikali. Tačiau dėl riboto DSP tikslumo, chaotiškas sekas generuotas esant unikalioms pradinėms sąlygoms galiausiai tampa identiška pasikartojančia seka. Siekiant to išvengti, chaotinis generatorius periodiškai vėl nustatomas į pradines sąlygas.

Taigi, chaotiškas generatorius veikia priskirdamas kiekvienai vartotojo skirtingų IC ir kartu unikalų sekos ilgį.

Koreliacijos bloko eksploatavimas sudaromas padauginus gautą signalą vietoje susidarančių chaotinių sekų ir skaičiavimo sukauptos sumos ir šių dauginimo rezultatų skaičius lygus plitimo veiksmui. Kaip pirminį pranešimą simboliai yra 1 arba -1. Todėl siekiant nustatyti geriausią įvertinimą perduodami žinutę, sprendimas grandinės blokas lygina koreliacijos išėjimo nulio slenkstis.



4. pav. Įprastos skaitmeninės komunikavimo sistemos blokas kombinuotas su chaoso pagrindu.



5. pav. CSK-BPSK komunikavimo sistemos blokas

Analoginės komunikavimo sistemų formos, chaosas, komunikacijos sistemos, kuriose naudojamos būdus, kaip chaotiškas maskavimas, chaotiškas moduliavimas, ir daugelis kitų, kurios buvo pasiūlytos. Dauguma šių analoginių schemų, tačiau perdavimo kanalas neveikia gerai kai kanale yra įprastiniai triukšmai. Tarp daugelio skaitmeninių schemų, chaosu pagrįstos schemas yra plačiausiai tyrinėjamos.

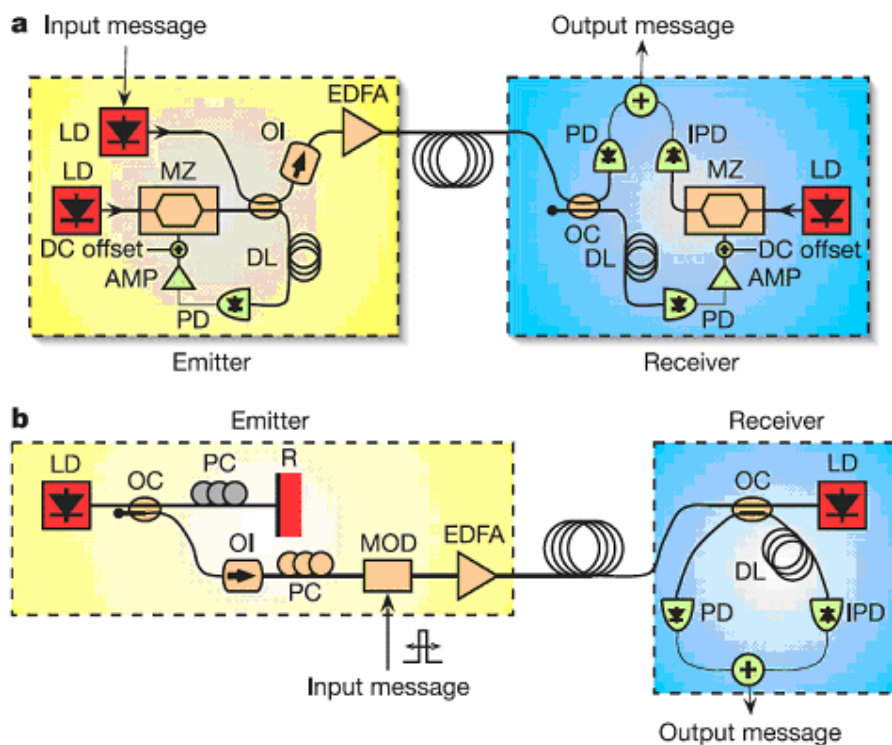
Paprastai, skaitmeninės chaoso pagrindo sistemose skaitmeninių simbolių sekos yra susietos su neperiodine chaotinė funkcijų seka. Aptikimo sistemas galima suskirstyti į nuoseklią ir nenuoseklią. Nuoseklioms aptikti, pavyzdžiui, chaotinių maskavimo ir nuoseklių CSK sistemų, imtuvas turi atkurti tokį pat chaotinį išlaikymą, kuris buvo naudojamas išlaikyti informacijai, dažnai vadinamas chaotiniu sinchronizavimu, kurį yra labai sunku išlaikyti praktikoje. Taigi, kol praktinės chaoso sinchronizavimo sistemos taps prieinamos, nuosekliu chaosu pagrįstos sistemos, lieka tik teoriniams tyrinėjimams. Nenuosekiose sistemose, chaotiniam išlaikymui neprivalo būti regeneruotas imtuve. Paprastai nenuosekliai aptikimui naudojamos kai kurios siunčiamų signalų savybės, kurios gali turėti tam būdingas deterministines savybes (pvz., nustatyti optimalų aptikimą, gražinimo planas grindžiamas nustatymu, maksimumo, kaip tikimybės metodas) arba sukurtos tam tikram nustatymui (pvz., DCSK), arba kai kurios statistinės savybės (pvz., bitų energijos aptikimas). Kol chaoso sinchronizavimas nėra reikalaujamas, nekoherentinės sistemos pristato, labiau praktinę chaoso bendravimo formą. Be to, nuoseklios sistemos, teoriškai lenkia savo nenuoseklias kolegas, ir koreliacija pagrįstas nuoseklus nustatymas, nustatymo požiūriu bitų klaidų efektyvumas yra optimalesnis.

Chaosu pagrįstos komunikavimo sistemos ir įprastos sistemos, kurių dažnių juostos, nemaža dalimi sutampa. Šios abi sistemos pavaizduotos trečiame ir ketvirtame paveiksle. Šioje sistemoje,

dviejų nepriklausomų duomenų srautus nuspręsta siųsti ta pačia duomenų perdavimo sparta. Šie du signalai susumuojami, taip pat su triukšmu kanale, prieš tai kol jie pasiekia kanalo pabaigą. Imtuve, remiantis gaunamu signalu, abiejų tipų sistemos bando susigrąžinti jų atitinkamose duomenų srautus. Nuoseklus arba nenuoseklus aptikimo sistemos gali būti taikomos imtuvuose, priklausomai nuo moduliacijos metodų siųstuvais.

Aišku, tiek chaotiškų tiek ir tradicinių sistemų signalai interferuoja vienas su kitu ir dėl to kiekvienos sistemos veikimas bus blogesnis

Dar vienas pavyzdys kalba apie tai kaip taikoma chaoso teorija naudojant didelio duomenų perdavimo sąsajas.



6. Pav. Du schematiniai nustatymai optinei chaoso komunikacijai.

Chaotiniai signalai buvo pasiūlyti kaip plačiajuostės informacijos perdavimas teikiant aukšto lygio patikimumo potencialo ir privatumo duomenų perdavimui. Laboratorinės demonstracijos optinio ryšio chaoso pagrindu jau parodė šios technologijos potencialą, o lauko bandymai, naudojant komercinius optinius tinklus, iki šiol nepradėti. Čia įrodyti greitųjų tarp miestinių ryšių, remiantis chaoso sinchronizacijos virš komercinių optinio pluošto kanalais. Optinė nešančioji generuojama chaotiniu lazeriu naudojamu užkoduoti žinutę perdavimo per 120km optinio pluošto tarp didmiesčių Atėnų ir Graikijos. Žinutei iššifruoti naudojamas antras tinkama lazeris, kuris, sinchronizuojamas su chaotiniu siuntėju, leidžia atskirti siuntėją ir pranešimą. Perdavimo greitis gigabitsais per sekundę pasiektas šiame diapazone, atitinkamai-bitų klaidų mažiau nei 10⁻²⁷. Sistema naudoja porų

puslaidininkų lazeriai, kaip chaotiškus siųstuvus ir imtuvus, ir off-the-shelf optinio pluošto telekomunikacijos komponentus. Rezultatai rodo, kad informacijos gali būti perduodami aukšti bitų kursai naudojant deterministinius chaosus taip, kad būtų atsparus trukdžiams ir kanalo sutrikimams kurie realiomis sąlygomis neišvengiami.

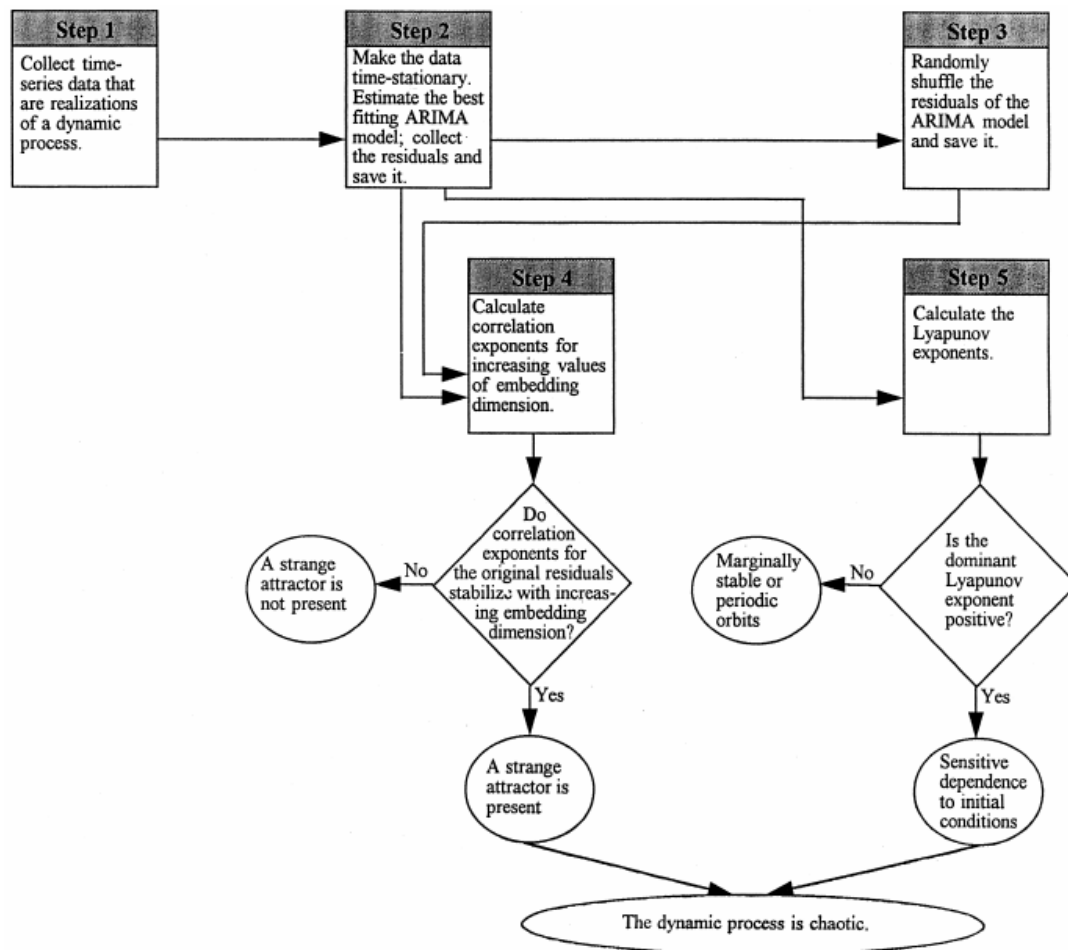
Kas ta chaoso teorija

Chaoso teorija yra fizikos, ekonomikos ir filosofijos studijų elgesio dinaminės sistemos, kurios yra labai jautrios pradinėms sąlygoms. Šis jautrumas yra populiariai vadinamas drugelio efektu. Maži pradinių sąlygų skirtumai (pavyzdžiui, dėl apvalinimo paklaidų) lemia labai skirtingus rezultatus dėl chaotiškai sistemų ir todėl padaryti ilgalaikę prognozę apskritai neįmanoma.

Trumpa chaoso teorijos apžvalga yra subrinkinys dinaminės sistemos teorijoje, kurių kilmę galima atsekti pagal Poincare 1899 tyrimų planetinę dinamiką. Ankstyvuose darbuose apie chaoso teorijos tyrimus ir taikymus yra įvykdyta biologijoje, meteorologijoje, fizikoje, chemijoje ir kompiuterių moksle. Pastaraisiais metais chaoso teorija krypsta į socialinių mokslų tyrinėtojus, nes ji suteikia galimybę išnagrinėti sudėtingą elgseną kuri anksčiau buvo vengiama paaiškinti Etheridge ir Sriram, 1993, Parker ir Stacey, 1994, Levy, 1994; Thie'tart ir Forgues, 1995. Straipsnio apžvalgoje Feichtinger ir Kopel 1993. pasiūlytos dvi kategorijos chaoso teorijos taikymų ekonomikoje ir vadyboje. susitelkiant į pirmą kategoriją yra vystoma modelių simuliacija, kuri reiškia, kad chaosas yra loginė dinaminio proceso išvada. Bendri chaoso teorijos taikymai kurie priklauso šiai kategorijai yra atsargų valdymas. Pavyzdžiui, yra studijų, kuriose chaosas yra gautas per vėlinimus gamyboje ir pardavimuose Serman, 1988, Mosekilde ir Larsen, 1988, Rasmussen ir Mosekilde, 1988, Levy, 1994. Susikoncentruojant į antrą taikymų kategoriją yra nagrinėjami empiriniai duomenys tam, kad būtų aptiktas chaotiškas elgesys pagrindiniuose dinaminuose procesuose. Pavyzdžiui, studijuojant ekonomiką ir finansus, kur randamas empirinis chaotinių kelių palaikymas laiko duomenų linijose priklausančiuose šiai kategorijai. Brock, 1986, Brock ir Sayers, 1988, Scheinkman ir LeBaron, 1989. Be to, nesenose studijose valdymo literatūroje pagal Cheng ir Van de Ven 1996, kurie parodo kad veiksmai ir rezultatai susiduria su naujovių grupėmis rodančiomis chaotiškus modelius pradiniam naujovių vystymo laikotarpyje, kuris priklauso antrai kategorijai. Buvo surinktos laiko duomenų linijos, kurių realizacija skirta naujovių įgyvendinimo procesui aukštųjų technologijų gamybos įmonėms. Metodinių požiūrių tikslas yra išnagrinėti ar tirti duomenys turi dvi fundamentalias chaotinio elgesio savybes: neįprastos traukos buvimas ir jautrumo priklausomybė nuo pradinių sąlygų.

Chaosu suvaldymas šiandien yra plačiai tyrinėjamas. Šios srities vystymas buvo paskatintas straipsnio, kurį parašė E. Ott, C. Grebogi ir J. Yorke į „Physics Review Letters“ 1990 metais, kur terminas „chaoso kontrolė“ buvo paminėtas pirmą kartą. Jis pabrėžė svarbų pokytį chaotiškos sistemos elgesyje, kuris gali būti pasiektas „maža“ sistemos parametrų korekcija. Ši išvada atvėrė galimybes kontroliuoti natūralias sistemas nekeičiant būdingų sistemos savybių. Idėja buvo greitai

įvertinta gamtos mokslų srityje. Fradkov ir Evans padarė „chaoso kontrolės“ sistemos tyrimą su detalia pagrindinių tyrimų šakų diskusija. Taikymai ir galimybės įvairiose mokslo ir inžinerijos srityse buvo tirti ir įrodyti. Jie padarė išvadą, kad „chaoso kontrolės“ sistema sulaukė daug dėmesio daugybėje inžinerijos sričių, tačiau daugelis problemų liko neišspręstos. „Chaoso kontrolės“ idėja yra labai patraukli. Čia nepaminti pastatų tyrimai kurie naudoja chaoso kontrolę. Priežastys yra tos, kad sąvokos „jautrumas savo pradiniam išėjimui taškui“ ir „keista trauka“ nėra visuotinai suprantama. Kitais žodžiais tariant, kontrolės atvejis pažangiuose pastatuose būtų labai svarbus ir įdomu tyrinėtojams užstatytoje aplinkoje kai gaunami neišskūs arba netikslūs signalai, kurie anksčiau buvo laikomi atsitiktiniais, o dabar pasirodė jog yra deterministiškai paaiškinami keista trauka.



7. pav. Duomenų analizės etapai, siekiant išbandyti chaotiškas dinamiško proceso ypatybes. Šaltinis: Shekhar Jayanthi, Kingshuk K. Sinha „Innovation implementation in high technology manufacturing: A chaos-theoretic empirical analysis“

Chaotinę elgseną galima pastebėti daug kur, pavyzdžiui gamtoje. Pati chaoso teorija gana tiksliai apibrėžia kas tai yra chaotinės sistemos. Ši teorija teigia, kad dinaminės sistemos turi pasižymėti tokiomis savybėmis:

- Atsižvelgimas į pradinės sąlygas;
- Topologinis maišymasis;
- Jos periodinės trajektorijos turi būti tankios.

Jautrumas pradinėms sąlygoms reiškia, kad kiekvienas tokios sistemos taškas yra apytiksliai artimas kitiems taškams, tačiau ateityje šios trajektorijos skiriasi. Taigi, kad ir koks mažas nukrypimas nuo dabartinės trajektorijos, ateityje gali būti labai skirtingos trajektorijos.

Pradinių sąlygų jautrumo pasekmė yra ta, kad jei mes turėsime tik ribotą kiekį informacijos apie sistemą (kaip paprastai pasitaiko praktikoje), tada po tam tikro laiko sistema nebebus nuspėjama. Tai labiausiai matoma oro prognozių atveju, kurios paprastai yra nuspėjamos tik apie savaitę į priekį.

Lyapunov rodiklis apibūdina jautrumą pradinėms sąlygoms. Kiekybiškai, dvi fazės erdvės trajektorijos pradiniu atskyrimu δZ_0 diverguoja:

$$|\delta Z(t)| \approx e^{\lambda t} |\delta Z_0|$$

kur λ yra Lyapunov rodiklis. Atskyrimas gali būti skirtingas priklausomai nuo pradinio atskyrimo vektoriaus. Taigi, yra visas Lyapunov eksponentinių rodiklių spektras – jų kiekis priklauso nuo fazės dimensijų ir yra jam lygus. Akivaizdu, kad tik didžiausias, t.y. maksimalus Lyapunov rodiklis (MLE) gali nustatyti bendrą sistemos nuspėjamumą. Teigiamas MLE paprastai suprantamas, kad sistema yra chaotiška. Toliau trumpai panagrinėjamos anksčiau išvardintos chaotinės sistemos savybės.

Chaotinės sistemos savybės

Topologinis maišymasis

Topologinis maišymasis (arba topologinis Tranzityvumas) reiškia, kad sistema laikui bėgant bus tobulinama taip, kad konkreti sritis ar atvira erdvė ilgainiui iš dalies sutampa su kitų sričių dalimis. Ši matematinė sąvoka „mišinys“ atitinka standartinę spalvos maišymo skysčiuose situaciją ir yra chaotiškos sistemos pavyzdys.

Topologinis maišymasis dažnai praleidžiamas chaoso teorijoje, kuris chaosą prilygina pradinių sąlygų jautrumui. Tačiau chaosas vien jautrumo nuo pradinių sąlygų nesuteikia. Pavyzdžiui, paprastos dinaminės sistemos gaminamos ne kartą padvigubinant pradinę vertę. Ši sistema jautri priklausomybę nuo pradinių sąlygų visur, nes bet kokia pora esanti netoliese gali tapti plačiai atskirta. Tačiau šiame pavyzdyje nėra topologinio maišymosi ir todėl neturi jokio chaoso. Iš tiesų, sistema labai paprasta: visuose taškuose, išskyrus 0 artėjama prie begalybės.

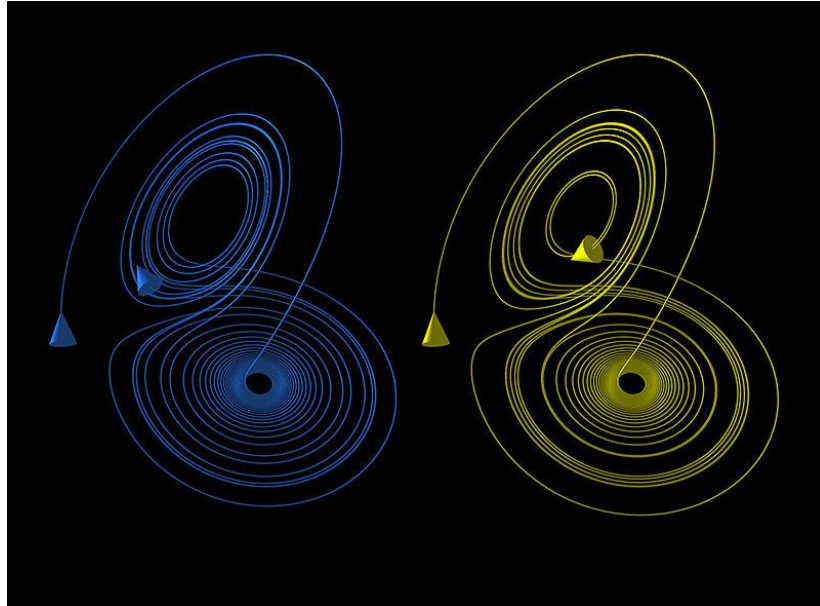
Periodinių orbitų tankis

Periodinės orbitos tankis reiškia, kad kiekvienas erdvės taškas yra vertinamas subjektyviai atidžiai periodiškai trajektorijai. Topologiškai besimaišančios sistemos gali būti nejautrios pradinėms sąlygoms ir gali būti ne chaotiškos. Pavyzdžiui, neracionalus apskritimo sukimasis gali būti topologiškai tranzityvus, bet nėra tankios periodinės orbitos, ir tokiu būdu nėra jautri priklausomybė nuo pradinių sąlygų. Vienmatė logistikos žemėlapyje apibrėžta $x \rightarrow 4x(1-x)$ yra vienas iš paprasčiausių periodinės orbitos sistemų. Pavyzdžiui, $\frac{5-\sqrt{5}}{5} \rightarrow \frac{5+\sqrt{5}}{5} \rightarrow \frac{5-\sqrt{5}}{5}$ (Arba maždaug $0,3454915 \rightarrow 0,9045085 \rightarrow 0,3454915$) yra (nestabili) antro periodo orbita ir panašios orbitos egzistuoja ir kitiems periodams: 4, 8, 16, ir t.t. (visi periodai nurodyti Sharkovskii's teorema).

Sharkovskii's teorema yra pagrindas Yorke ir Li (1975) įrodymo, kad bet kuri vienmatė sistema, kuri naudoja trečio periodo reguliarų ciklą taip pat naudoja kitus reguliarius ciklus bet kokio kito dydžio taip pat gerai kaip chaotinės orbitos.

Atraktorius

Kai kurios dinaminės sistemos, tokio kaip vienmatė sistema logistikos žemėlapyje apibrėžta $x \rightarrow 4x(1-x)$, yra chaotiška visur, tačiau daugeliu atvejų chaotiškas elgesys randamas tik etapo pogrupyje. Daugumos interesų atvejais atsiranda tada, kai chaotiškas elgesys vyksta atraktoriuje, nes tada daug pradinių sąlygų bus orbita, kad artėja prie šio chaotiško regiono.



8. pav. Lorenz Atraktorius rodo chaotinę elgesį. Šie du brėžiniai įrodyti jautri priklausomybę nuo pradinių sąlygų, kaip apibrėžtas etapas plotas iki atraktoriaus regione.

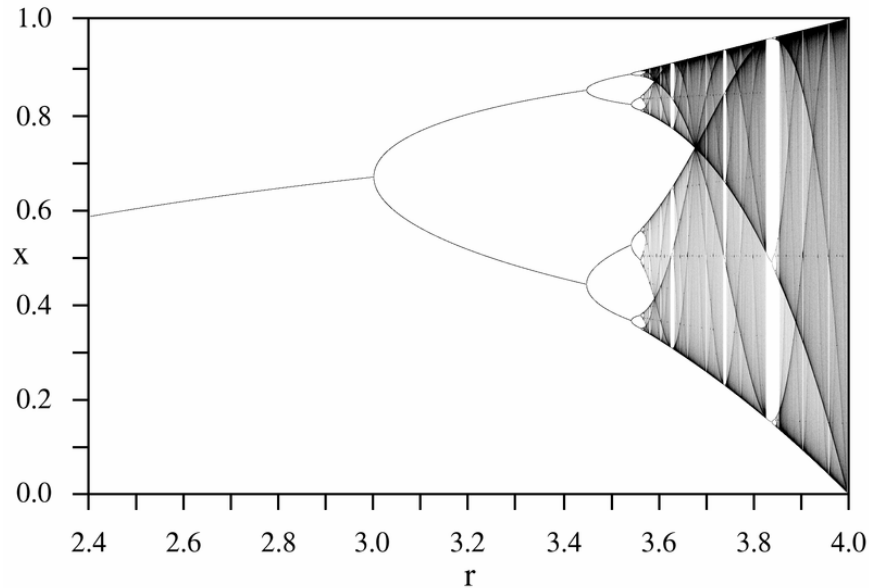
Paprastas būdas vizualizuoti chaotišką atraktorių yra pradėti taško traukos atraktoriaus baseine ir tiesiog supaprastinti orbitas erdvę. Dėl topologinio tranzityvumo sąlygų, tai gali vertinamas kaip galutinis atraktoriaus vaizdas ir iš tikrųjų orbita parodyta paveikslėlyje viršuje pateikia bendros Lorenz Atraktoriaus formos vaizdą. Šio atraktoriaus rezultatai yra paprasto trijų dimensijų Lorenz modelio oro sistema. Lorenz atraktorius turbūt vienas iš žinomiausių chaotiškos sistemos schemų, tikriausiai todėl, kad jis buvo ne tik vienas pirmųjų, tačiau ji taip pat yra vienas iš sudėtingiausių, todėl kelia labai įdomus modelis, kurie panašūs į drugelio sparnų modelį.

Skirtingai nuo fiksuoto Atraktoriaus taško ir ribinių ciklų, Atraktorius, kuris sudaro chaotiškas sistemas, žinomas kaip Atraktorius, labai detalizuotas ir sudėtingas. Neįprasti Atraktoriai aptinkami ir nuolatos dinaminės sistemos (pvz., Lorenz sistema), ir kaip atskiros sistemos (pvz., Xenon struktūra). Kitos diskretinės dinaminės sistemos turi atmušimo struktūrą, vadinama Žiuljeno, kuris yra tarp atraktorių baseinų ar nustatytų taškų – Žiuljeno nustatymai gali būti suvokiami kaip keisti atmušėjai. Abu neįprasti Atraktoriai ir Žiuljeno nustatymai paprastai Fraktalo struktūros ir Fraktalas dimensijos gali būti jiems apskaičiuoti.

Minimalaus sudėtingumo chaotinė sistema

Diskretinės chaotiškos sistemos, tokios kaip logistinis žemėlapis, gali būti Atraktoriaus naudojamos nepriklausomai nuo jų dimensijų. Poincaré-Bendixson teorema rodo, kad neįprastas Atraktorius gali iškilti tik tuomet, kai nuolat dinaminės sistemos (nurodyta diferencialinė lygtis), jei

jis turi tris arba daugiau matmenų. Baigtinių matmenų linijinė sistema niekada nebūna chaotiška dinaminėms sistemoms, kad būtų rodomas chaotiškas elgesys ji turi būti arba netiesinė arba begalybės matmenų.



9. pav. Logistikos žemėlapis Bifurkacija schema $x \rightarrow R \times (1 - x)$. Kiekviena vertikali dalis rodo Atraktoriaus konkrečią reikšmę r . Diagrama rodo periodo dubliavimą kai r didėja, galiausiai gaunamas chaosas.

Poincaré-Bendixson teorema teigia, kad dviejų matmenų diferencialinė lygtis turi labai reguliarių elgesį. Lorenz Atraktorius, kuris buvo aptartas anksčiau, gaunamos iš trijų diferencialinių lygčių su septyniomis sąlygomis dešinėje pusėje iš viso, iš kurių penkios yra tiesinės sąlygos ir iš kurių dvi yra kvadratinės (ir todėl netiesinės). Kitas gerai žinomas chaotiškas atraktorius yra Rössler lygčių su septyniomis sąlygomis dešinėje pusėje sugeneruojamas, iš kurių tik viena yra (kvadratinė) netiesinė. Sprott nustatė, trimatę sistemą tik su penkiomis sąlygomis dešinėje pusėje, ir tik vieną kvadratinę netiesiškumą, kuris parodo chaosas esant tam tikroms parametru vertėms. Zhang ir Heidel parodė, kad išsklaidytos ir konservatyvios kvadratinės sistemos, trijų dimensijų kvadratinės sistemos tik su trimis ar keturiomis sąlygomis dešinėje pusėje negali parodyti chaotiško elgesio. Taip yra todėl, kad sprendžiant tokias sistemas asimptotiniai dviejų matmenų paviršiaus sprendimai yra lengvai pritaikomi.

Poincaré-Bendixson teorema reiškia, kad nuolat dinaminės sistemos Euklido plokštumoje negali būti chaotiškas, dvimatės sistemos nenutrūkstamo srauto sistemos su ne Euklidine geometrija gali parodyti chaotišką elgesį.

Atsitiktinis chaotinių duomenų išrinkimas

Iš duomenų gali būti sunku pasakyti fiziniai ar kiti pastebėti procesai yra atsitiktiniai ar chaotiniai, nes praktiškai nėra tokių laiko eilučių kurias sudarytų iš grynų signalų. Visada bus kažkiek trukdžių, net jei jis yra suapvalinimas arba vietinė paklaida. Taigi bet kokia reali laiko eilutė, net jei daugiausia deterministinė, turės šiek tiek atsitiktinumo.

Visi deterministinių ir atsitiktinių procesų metodai skiriami remtis aplinkybe, kad deterministinė sistema visada vystosi taip pat iš nurodyto pradinio taško. Taigi, testuojant determinizmą laiko serijomis, galima:

Pasirinkti bandymo būseną;

Surasti panašias laiko eilutes arba artimas būsenas;

Palyginti atitinkamus laiko pokyčius.

Skirtumų nustatymas tarp laiko evoliucijos ir testo būsenos panašaus t evoliucijos būseną. Deterministinė sistema turės klaidą, kuri arba yra nedidelė (stabilus, reguliarus sprendimas) arba didėja eksponentiškai su laiku (chaosu). Atsitiktinė sistema turės atsitiktinę klaidą.

Pagrindinės chaotinės sistemos problemos

Pagrindinė problema, su kuria susiduriama analizuojant laiko linijas ieškant chaotinės elgsenos yra autokoreliacija. Autokoreliacijos atsitiktinumumas gali būti lengvai supainiotas su chaosu. „Liekamasis testas“ yra aprašytas Brock 1986, ir Brock ir Sayers 1988 ir yra priemonė šiai problemai spręsti. Tai apima pritaikymą patikimiems modeliams laiko linijų duomenims ir analizei likusių duomenų naudojant nustatymą aptikti chaotiniam elgesiui.

Duomenų analizės veiksmi:

Pirmas žingsnis. Surenkami laiko linijų duomenys pagal kiekio, ciklo trukmės ir aktyvumo parametrus, kurie yra realizuoti dinaminiame procese pagal tyrimus, naujovių diegimus aukštų technologijų gamyboje.

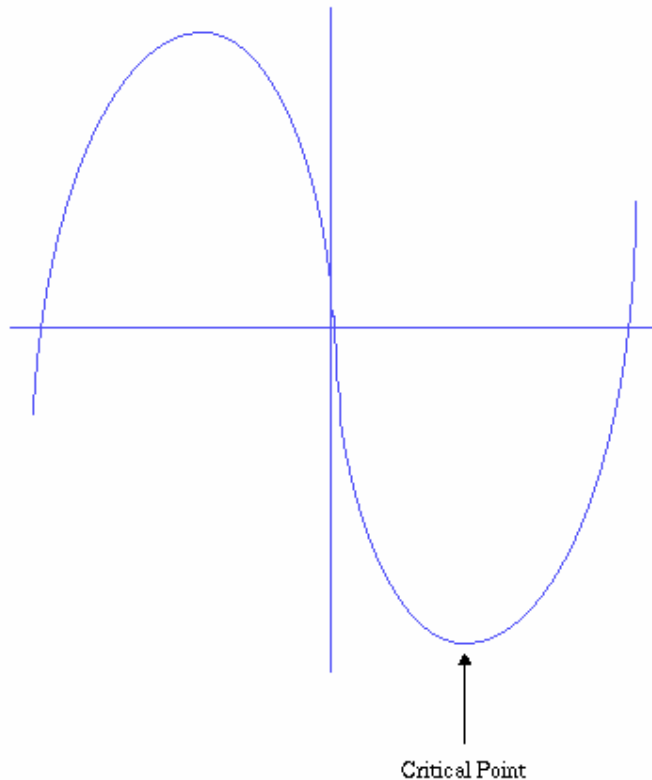
Antras žingsnis. Padaromi duomenys laiko stabdyme 13 ir įvertinamas geriausiai tinkantis ARIMA modelis stacionarių duomenų rinkiniui pagal kiekio, ciklo trukmės ir aktyvumo parametrus. ARIMA modelis atitinkamai įvertina automatiškai grįžtančių ir slenkančių vidurkinių parametru modelyje. Daugelį kartų originalūs laiko linijų duomenys yra skirtingi, kai jie yra stacionarūs. Įprasti kriterijai yra naudojami atrinkti geriausiai tinkamam modeliui. Tai įtraukia stacionarumą, parametro reikšmę, diagnostinį tikrinimą išliekamam balto triukšmo atvaizdavimo charakteristikoms ir Akaike informacinio kriterijaus minimizavimui (AIK (AIC - Akaike Information Criterion)). Likę duomenys atitinkamai geriausiai tinka ARIMA modelio duomenų kiekiui, ciklo trukmei ir aktyvumui yra surenkami ir išsaugomi.

Trečias žingsnis. Atsitiktinis maišymas atsitiktinių duomenų atitinkamai geriausiai tinkančių duomenų ARIMA modelio duomenų kiekiui, ciklo trukmei ir aktyvumui yra atskirai išsaugomi.

Ketvirtas žingsnis. Apskaičiuojami koreliacijos rodikliai 14 skirtingų apimamų aspektų – didėjančia tvarka, pradedant ms verte 1, 2 ir taip toliau, naudojant originalius likutinius duomenis geriausiai tinkančius ARIMA modelį ir jo atitinkamai atsitiktinai išmaišytai daliai. Jei koreliacijos rodikliai originaliems likutiniams duomenims stabilizuoti didinant išplėtais dydžiais, tuomet jis siūlo keistos traukos vaizdavimą. Tačiau jei koreliacijos rodikliai ir toliau didėja su išplėtais dydžiais, kaip būtų tuo atveju, kai atsitiktinai maišomi likutiniai duomenys, tuomet jis siūlo, kad keista trauka nebūtų vaizduojama ir originalūs laiko linijų duomenų rinkiniai yra atsitiktinio proceso realizacija. Nuo tada kai analizuojami ARIMA likutiniai duomenys, naudojamas išplėstinį vėlinimą. Išplėstinis vėlinimas yra apskaičiuotas, kai mažiausia vėlinimo vertė, dėl kurios autokoreliacijos funkcija yra apytiksliai lygi nuliui.

Penktas žingsnis. Lyapunov eksponentės apskaičiavimas naudojant originalius likutinius duomenis geriausiai tinkančius ARIMA modelio duomenų kiekiui, ciklo trukmei ir aktyvumui. Jei

dominuojanti padėtis Lyapunov eksponentėje yra teigiama, tuomet siūloma vaizduoti dinaminį procesą priklausantį nuo pradinių sąlygų; nulinė eksponentė siūlo šiek tiek stabilų laiko kelią ir neigiama eksponentė siūlo periodinio laiko kelią. Dinaminis procesas laikomas chaotišku kai čia yra empirinis palaikymas abiejų neįprastos traukos ir jautrumo priklausomybės nuo pradinių sąlygų atvaizdavimas.



10. pav. Kritiniai taškai. Šaltinis: <http://library.thinkquest.org/3493/noframes/chaos.html>

Didžioji dalis kritinių taškų yra veiksniai geriau paliekami į matematinius klausimus, vis dėl to jie taikomi į chaosą ir ypač fraktaluose. Kritiniai taškai yra naudojami kaip pradinės reikšmės specifinėms reikšmėms funkcijoje, kol apskaičiuojami fraktaliniai rinkiniai. Pavyzdžiui, Mandelbrot rinkinyje, pradinė z reikšmė funkcijoje $z = z^2 + c$, yra visuomet nulis, nes nulis yra kritinis funkcijos taškas.

Skaičiuojant kritinius funkcijos taškus reikalauja įvairių darbo žinių arba ieškojimas tangento funkcijos linijinio nuolydžio. Norėdami apskaičiuoti funkcijos kritinius taškus, Jūs turite rasti tašką, kuriame funkcija kinta nuo didėjimo į mažėjimą. Pirmas darinys gali būti naudojamas nustatyti kiekviename duotame taške ar funkcija yra didėjanti ar funkcija yra mažėjanti pagal paprastą taisyklę: kai funkcija yra auganti, pirma išvestinė yra teigiama, ir kai funkcija yra mažėjanti, pirma išvestinė yra neigiama. Beveik visais atvejais, kritiniai taškai bus išdėstyti ten, kur pirma išvestinė yra nulis, nors kai kuriais atvejais jie gali būti taške, kuriame funkcija yra neapibrėžta.

Chaotinių sistemų matematinų lygčių pavyzdžiai

Rössler Atraktorius

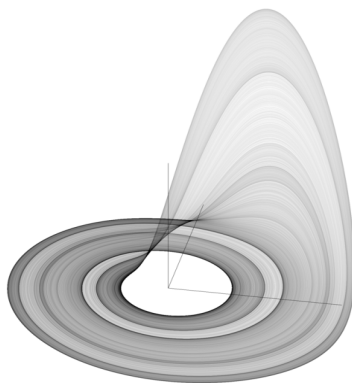
Rössler Atraktorius yra Atraktorius už Rössler sistemos, kurią sudaro netiesinės paprastosios diferencialinės lygtys. Šios diferencialinės lygtys apibrėžia nepertraukiamo laiko dinaminę sistemą, kuri parodo atraktoriaus chaotišką dinamiką susijusią su fraktalinėmis savybėmis. Kai kurios Rössler sistemos savybės gali būti išreikštos linijiniais metodais, pavyzdžiui, vektorius, tačiau pagrindinės sistemos savybės reikalauja, netiesinių būdų, pavyzdžiui, Poincaré žemėlapių ir Bifurkacijos diagramų. Originaliame Rössler straipsnyje teigiama, kad Rössler Atraktorius turėjo veikti panašiai kaip Lorenz atraktorius, bet taip bus lengviau analizuoti kokybiškai. Orbita atraktoriuje seka spiralės išore x , y plokštuma aplink nestabilią fiksuotą tašką. Kai kreivė pakankamai išorėje, antra fiksuoto taško įtakos grafikas, todėl padidės ir sukrumo z aspektas. Laiko srityje, tampa akivaizdu, kad nors kiekvienas kintamasis svyravimo per tam tikrą verčių ribą, virpesiai yra chaotiški. Šis atraktorius turi kai kurių panašumų su Lorenz Atraktorius, tačiau yra paprastesnis ir turi tik vieną pagrindinį lauką. Otto Rössler sukūrė Rössler Atraktorių 1976 m., tačiau vėliau buvo nustatyta, kad teorinė lygtis naudinga modeliuojant cheminių reakcijų pusiausvyrą. Apibrėžta lygtis yra:

$$\frac{dx}{dt} = -y - z$$

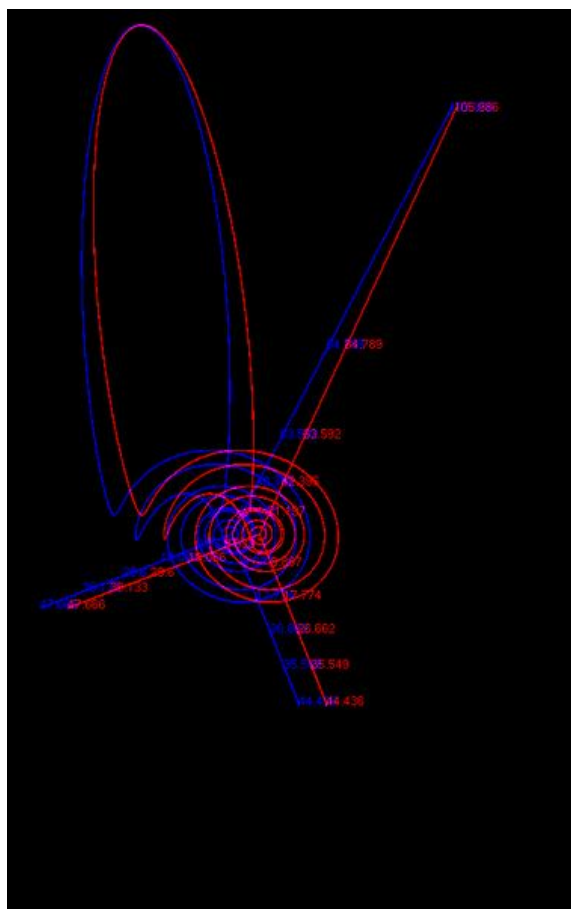
$$\frac{dy}{dt} = x + ay$$

$$\frac{dz}{dt} = b + z(x - c)$$

Rössler studijavo chaotiškus Atraktorių su $a = 0,2$, $b = 0,2$, $c = 5,7$, nors savybės $a = 0,1$, $b = 0,1$ ir $c = 14$ buvo dažniau naudojamos.



11. pav. Rössler Atraktorius. Šaltinis www.wikipedia.org



12. pav. Rössler Atraktorius kaip Stereograma su $a = 0,2$, $b = 0,2$, $c = 14$. Šaltinis www.wikipedia.org

Logistikos žemėlapis

Logistikos žemėlapis yra polinominis žemėlapių (analogiškai pasikartojimo atžvilgiu) ir antro laipsnio, dažnai minima kaip archetipinių pavyzdį, koks sudėtingas gali būti chaotiškas elgesys gali atsirasti dėl labai paprastos netiesinės dinaminės lygties. Struktūra buvo populiarinama 1976 biologo Robert May straipsnyje iš dalies kaip diskretinio laiko demografinis modelis analogiškas logistikos lygtims sukurtoms Pierre François Verhulst. Matematiškai logistikos žemėlapyje parašyta

$$x_{n+1} = rx(1 - x_n)$$

kur: x_n yra reikšmė tarp nulio ir vieneto ir pristato populiaciją n -taisiais metais, vadinasi x_0 yra pradinė populiacija; r yra teigiamas skaičius, ir parodo kombinuotą reprodukcijos ir bado normą.

Šis netiesinė diferencinė lygtis skirta užfiksuoti du reiškinius.

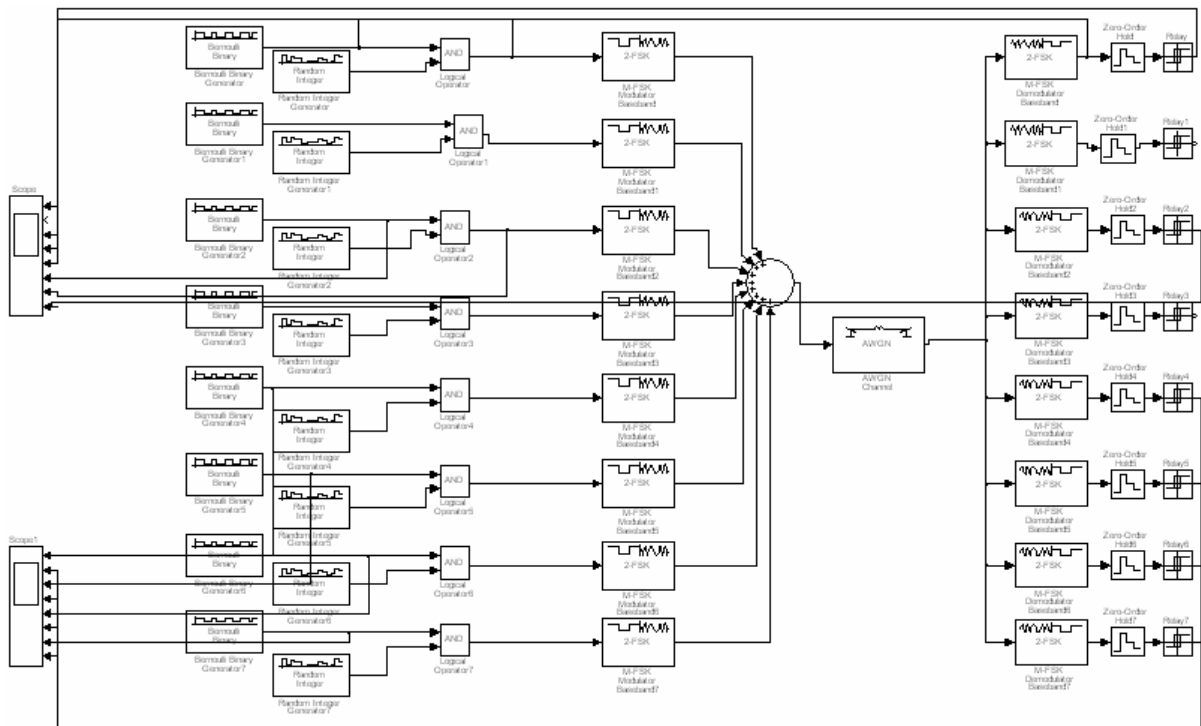
- reprodukcija, kai gyventojų populiacija padidės proporcingai dabartinei populiacijai, kai gyventojų skaičius yra nedidelis;

- badas (skaičius priklauso nuo mirtingumo) jei augimas mažės proporcingai apskaičiuotai vertei, remiantis teorine „keliamąja galia“ ir sumažins populiaciją.

Logistinio žemėlapių demografinis modelis turi patologinę problemą, kuri kai kurias pradines sąlygas ir parametrus veikia neigimai populiacijos dydį. Ši problema nepasirodo vėlesniuose Ricker modeliuose, kuris taip pat eksponuoja RCD.

Aštuonių kanalų duomenų perdavimo sistema

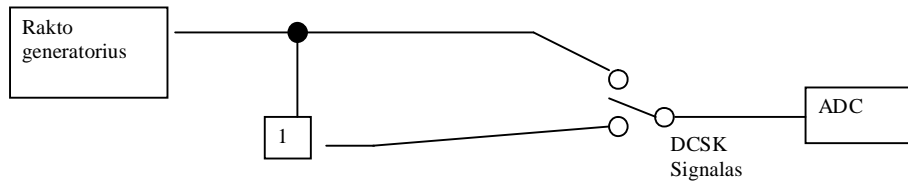
Darbo metu buvo bandoma naudojant MatLab 7.11.0 (R2010b) Simulink paketą sumodeliuoti aštuonių kanalų duomenų perdavimo sistemą, tačiau darbo metu nebuvo pasiektas norimas rezultatas – tinkamai užkoduotas kanalas. Sujungus aštuonis kanalus į vieną sistemą ir jų nekoduojant gaunami korektiški rezultatai, tačiau vos pabandžius koduoti kanalą signalai susimaišo ir jų nesugebu išskirti. Apačioje pateiktas paveikslas parodo vieną iš mano bandymų su MatLab 7.11.0 (R2010b) Simulink paketu sumodeliuoti aštuonių kanalų duomenų perdavimo sistemą.



13. pav. Aštuonių kanalų sistema.

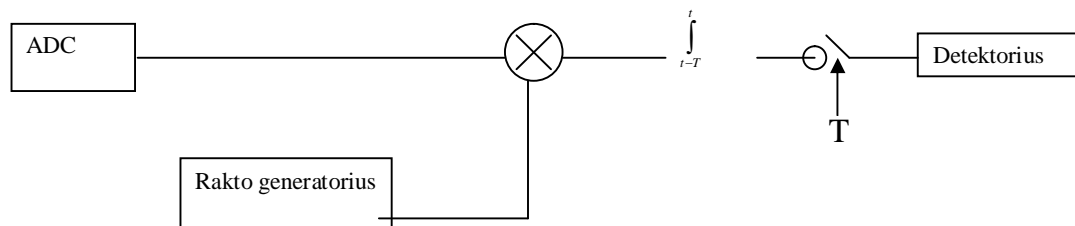
Nepavykus pasiekti norimo rezultato su MatLab 7.11.0 (R2010b) Simulink paketu, buvo nuspręsta imti jau padarytus siųstuvą ir imtuvą. Jų programinis kodas pateikimas pirmame ir antrame prieduose.

Siųstuvą sudarytas iš raktų generatoriaus, inverterio bei jungiklio. Bitų „1“ atveju generuojama chaotinė funkcija su raktu. Jei siunčiamas „0“ chaotinė funkcija invertuojama, bet raktas naudojamas tas pats.



14. pav. Siųstuvas.

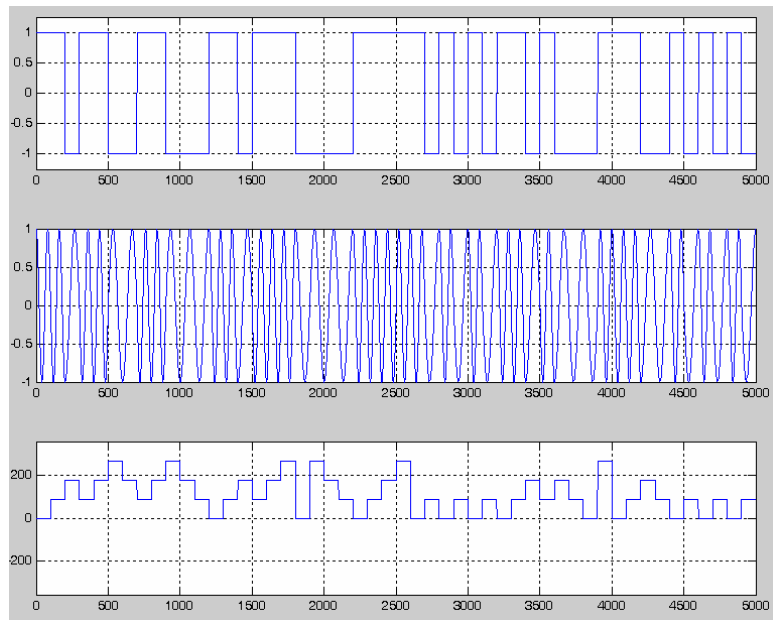
Imtuvas sudarytas iš rakto generatoriaus, kuris naudojamas gautam signalui koreliuoti duomenų atkūrimui, sumatoriaus bei detektoriaus, kuris pagal tam tikrą lygį nustato gaunamos informacijos turinį t.y. “1” arba “0”. Imtuvas pavaizduotas 15 paveikslyje:



15. pav. Imtuvas.

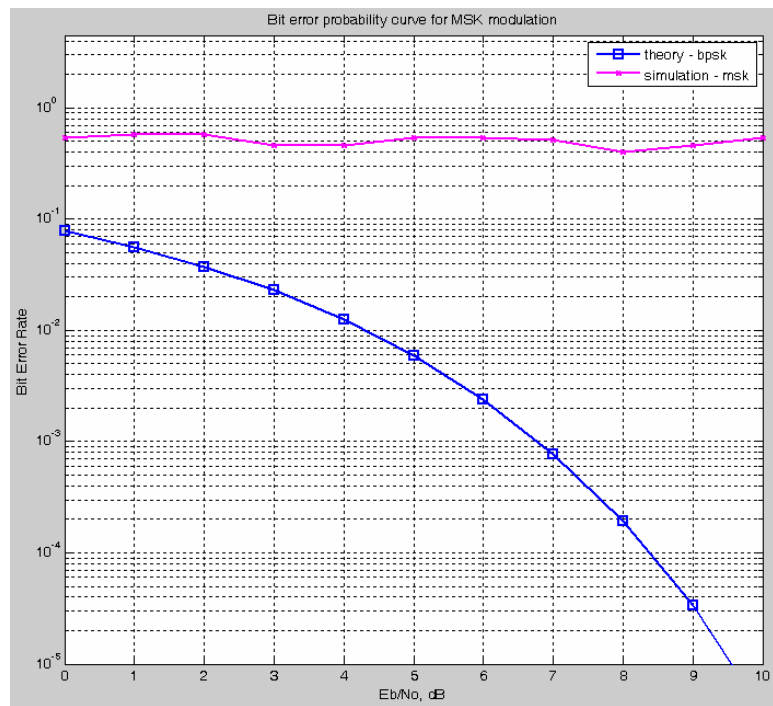
Pirmame priede pateikiamas kodas, kuriuo siekiama įgyvendinti duomenų perdavimą vienu kanalu, apskaičiuoti realų BER¹. Analizuojant įvairius MatLab scriptus, galima pastebėti, kad duomenų perdavimui naudojami siųstuvai ir imtuvai skiriasi, todėl darbui buvo pasirinktas vienas iš galimų variantų. Jų realizacija pateikta taip pat prieduose. Šešioliktajame paveiksle pavaizduoti trys grafikai. Pirmame vaizduojama atsitiktinė bitų seka. Antrasis vaizduoja generuojamą fazę. Trečiajame grafe matyti fazės ir signalo sandaugos rezultatas. Simuliacija atlikta imant dažnį lygų 10Hz. Toks nedidelis dažnis pasirinktas dėl per mažo turimos kompiuterinės įrangos pajėgumo.

¹ Bit Error Rate.

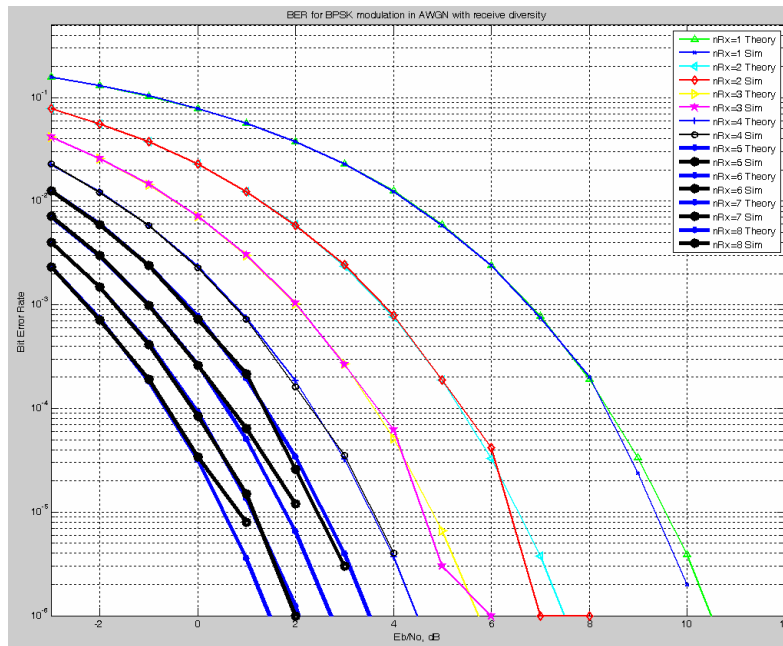


16. pav. Iš viršaus į apačią: Bitų seka, generuojama fazė ir bitų sekos ir generuojamos fazės sandauga.

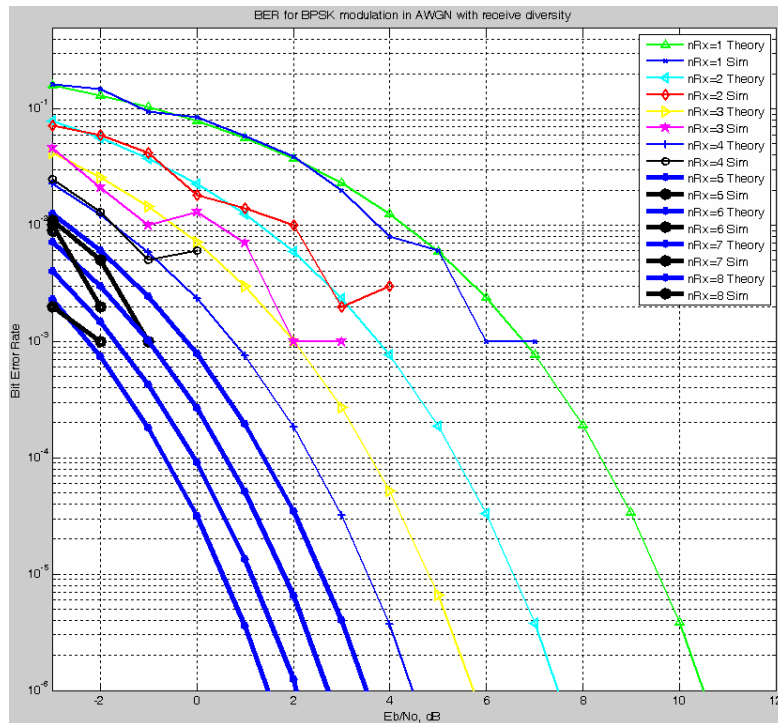
Toliau apačioje pateikiami trys paveikslėliai iliustruoja skirtingus BER skaičiavimo rezultatus.



17. pav. BER skaičiavimo grafiniai rezultatai, kai yra paduodama 20 bitų seka.



18. pav. BER skaičiavimo grafiniai rezultatai, kai paduodama 10^6 bitų seka. Grafiką generuojantis kodas pateiktas antrame priede.



19. pav. BER skaičiavimo grafiniai rezultatai, kai paduodama 10^3 bitų seka. Grafiką generuojantis kodas pateiktas antrame priede.

Darbo metu naudojami tokie raktai:

Pirmasis raktas: $R_1 = r + \cos(x^2 + 2)$;

Antrasis raktas: $R_2 = r + \sin(x^2)^2 + \cos(x^2)$;

Trečiasis raktas: $R_3 = \sin(x) + \cos(\frac{x}{2})$;

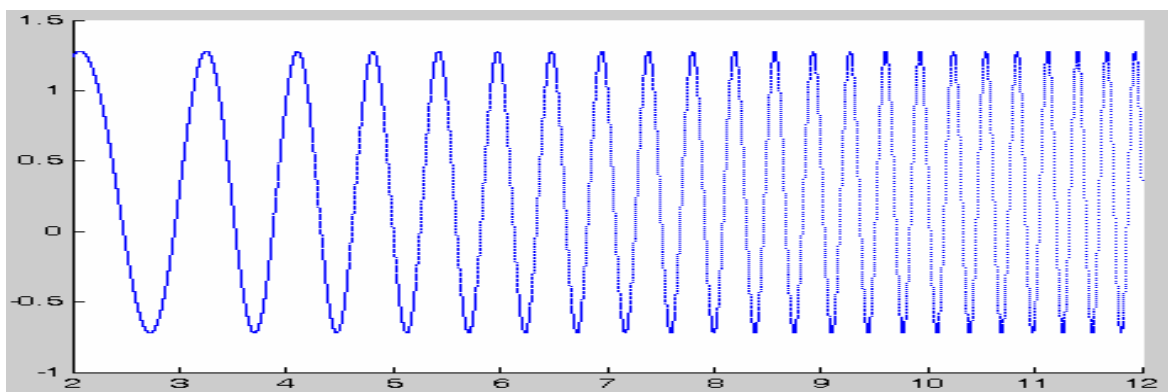
Ketvirtasis raktas: $R_4 = (\sin(x + 2) + \cos(\frac{x}{2}))^2$;

Penktasis raktas: $R_5 = (3 * \sin(x + 3) + \cos(\frac{x}{2}) - 1)^2$;

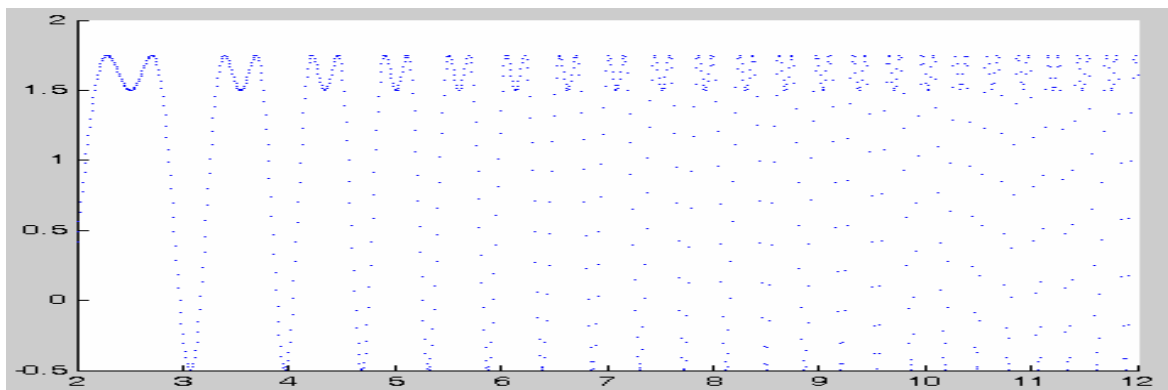
Šeštasis raktas: $R_6 = (\sin(x + 2) + \cos(\frac{x}{2}))^2$;

Septintasis raktas: $R_7 = (\sin(2x + 2) + \cos(\frac{x}{2}))^2 + \sin x$;

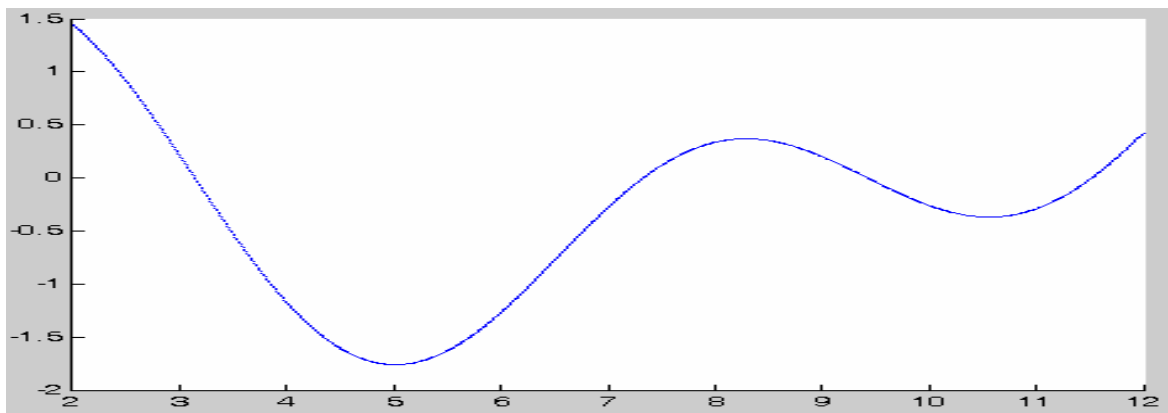
Aštuntasis raktas: $R_8 = (\sin(x + (\frac{\pi}{4})) + \cos(\frac{x}{5}))^2 + \sin x \cos 2x$



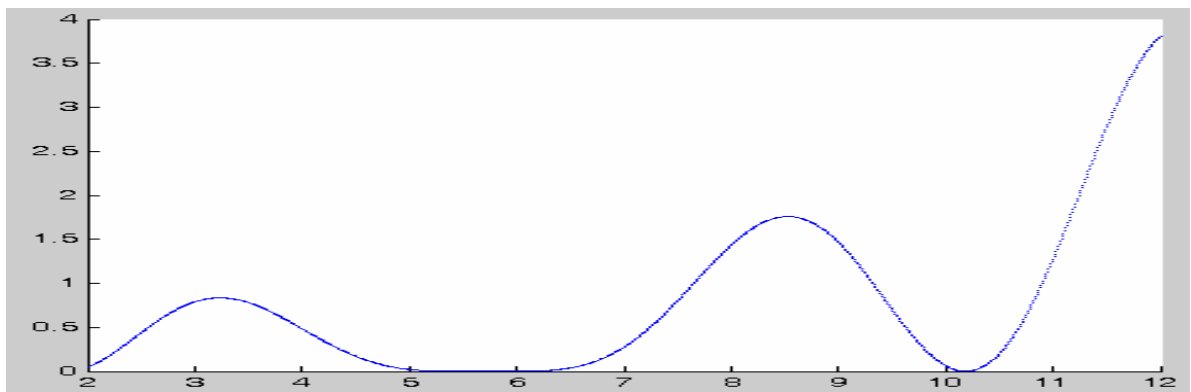
20. pav. Pirmas raktas.



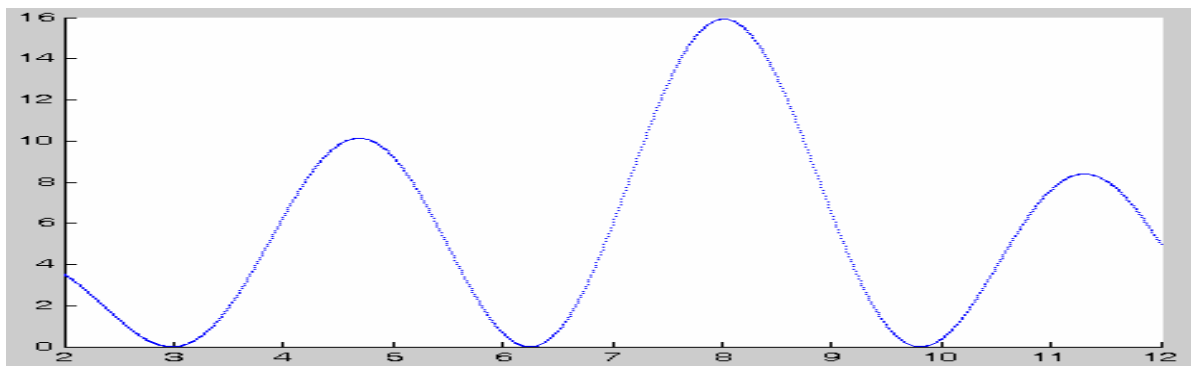
21. pav. Antras raktas.



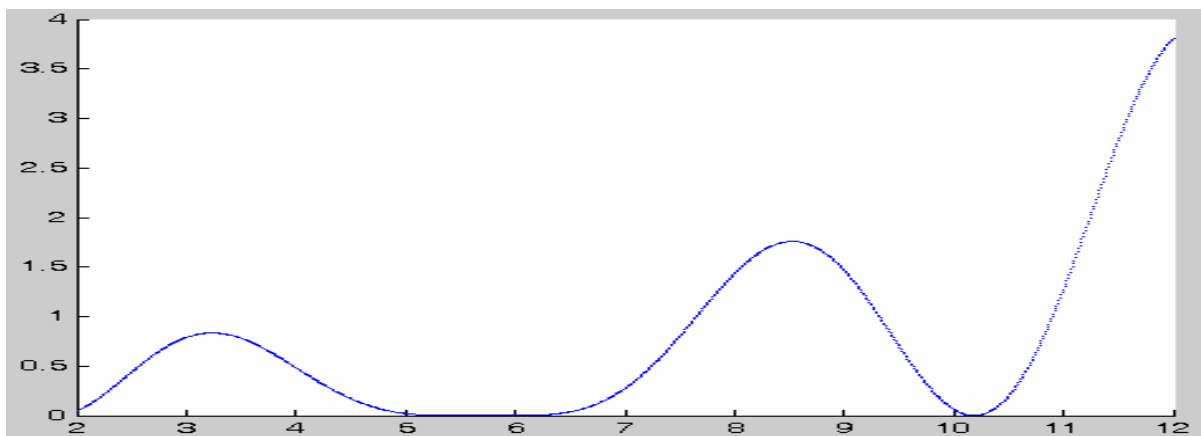
22. pav. Trečiasis raktas.



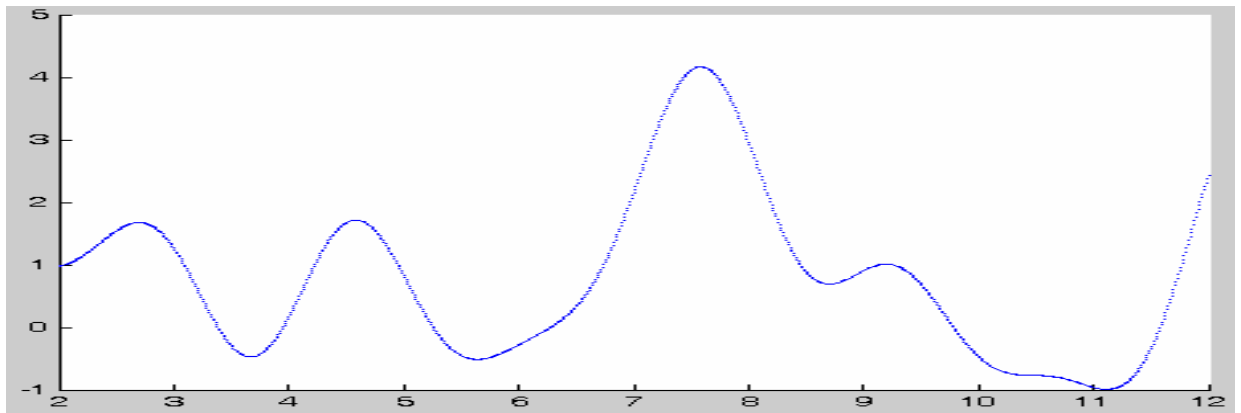
23. pav. Ketvirtasis raktas.



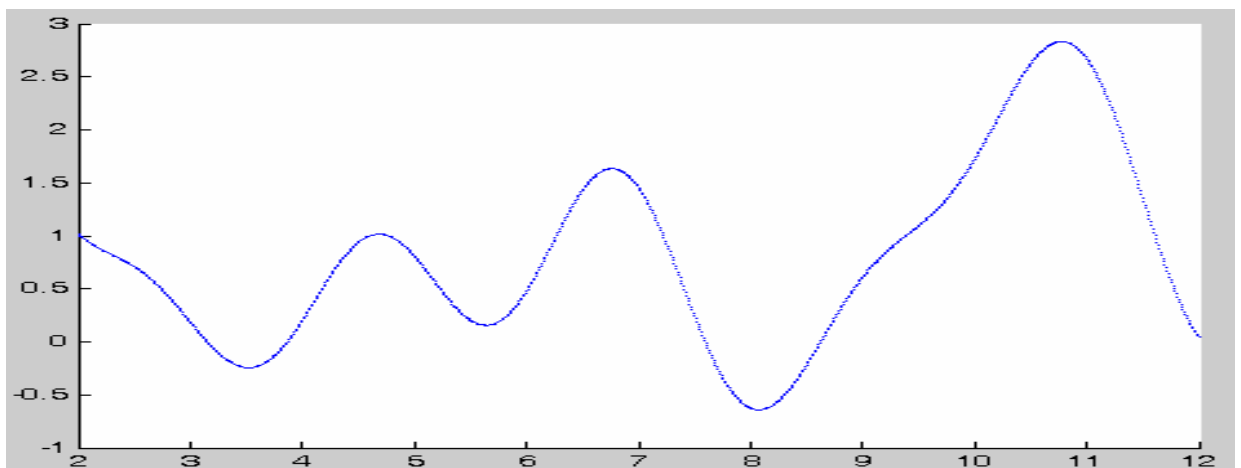
24. pav. Penktasis raktas.



25. pav. Šeštasis raktas.



26. pav. Septintasis raktas



27. pav. Aštuntas raktas

Išvados ir pasiūlymai

Trumpai apžvelgiau esamas sistemas veikiančias chaoso pagrindu. Kiekviena sistema turi savų privalumų ir trūkumų. Negalima išskirti vienos geriausios sistemos, būtina atsižvelgti į konkrečią situaciją.

Korektiškai ir be klaidų sumodeliuoti sistemos naudojant paketą MatLab 7.11.0 (R2010b) Simulink man nepavyko, todėl darbas užsitęsė.

Darbo metu daugiausia problemų kelia vienareikšmiškai matematiškai neapibrėžtas chaosas, todėl praktiškai chaosą generuoti labai sunku. MatLab 7.11.0 (R2010b) Simulink paketas siūlo random funkciją, tačiau atidiau panagrinėjus, galima pastebėti, kad ji po kiek laiko ima kartotis, kas rodo, kad sistema nėra chaotiška.

Remiantis chaoso teorija, kad labai maži sistemos pokyčiai gali lemti labai dideles pasekmes, reikia atsižvelgti ir į tų pokyčių priežastis bei kilmę, nes tokius pokyčius gali generuoti ne tik tyrinėjama sistema, bet ir aplinkos veiksniai: tai įvairiausi atspindžiai, triukšmai ir pan.

Kaip jau buvo minėta aukščiau, chaoso generuoti praktiškai beveik neįmanoma, todėl kiekvienu konkrečiu atveju reikia apsibrėžti kas yra lakoma chaosu, bei identifikuoti aplinkybes, kurios gali turėti lemiamos reikšmės darbo rezultatams.

Naudota literatūra:

1. Shintaro ARAI, Yoshifumi NISHIO, Takaya YAMAZATO „Two-Bit Modulation and Demodulation Schemes for Chaos-Based Noncoherent Communications Using Chaotic Dynamics“ Prieiga per internetą [žiūrėta 2010-01-25]: <http://nlab.ee.tokushima-u.ac.jp/nishio/Pub-Data/CONF/C324.pdf>
2. Georges Kaddoum, Martial Coulon, Daniel Roviras, Pascal Chargé „PERFORMANCE OF MULTI-USER ASYNCHRONOUS CHAOS-BASED COMMUNICATION SYSTEMS THROUGH M-DISTRIBUTED FADING CHANNEL“ Prieiga per internetą [žiūrėta 2010-01-25]:
<http://www.eurasip.org/Proceedings/Eusipco/Eusipco2009/contents/papers/1569185108.pdf>
3. Ljiljana Simic, Stevan Berber „Performance Analysis of a Chaos-Based Multi-User Communication System Implemented in DSP Technology“
4. Francis C. M. Lau, Chi K. Tse, Ming Ye, and Sau F. Hau „Coexistence of Chaos-Based and Conventional Digital Communication Systems of Equal Bit Rate“
5. Apostolos Argyris, Dimitris Syvridis, Laurent Larger, Valerio Annovazzi-Lodi, Pere Colet, Ingo Fischer, Jordi Garcí'a-Ojalvo, Claudio R. Mirasso, Luis Pesquera & K. Alan Shore „Chaos-based communications at high bit rates using commercial fibre-optic links“ Prieiga per internetą [žiūrėta 2010-01-25]: <http://nlp.eps.hw.ac.uk/Pubs/Nature-ChaosComm-05.pdf>
6. Internetinė enciklopedija Vikipedija. Prieiga per internetą [žiūrėta 2010-06-20]: <http://library.thinkquest.org/3493/noframes/chaos.html>
7. Internetinė enciklopedija Vikipedija. Prieiga per internetą [žiūrėta 2010-06-20]: http://en.wikipedia.org/wiki/Chaos_theory
8. Shekhar Jayanthi, Kingshuk K. Sinha „Innovation implementation in high technology manufacturing: A chaos-theoretic empirical analysis“, Journal of Operations Management 16 1998. 471–494
9. Xiaoshu Lu, Derek Clements-Croome, Martti Viljanen „Integration of chaos theory and mathematical models in building simulation Part II: Conceptual frameworks“, Automation in Construction 19 (2010) 452–457
10. Internetinė enciklopedija Vikipedija. Prieiga per internetą [žiūrėta 2010-06-20]: http://en.wikipedia.org/wiki/Rössler_attractor
11. Chaotic Spreading Spectrum System Development and Implementation Using FPGA/VHDL for Secure Wireless Communications
12. D. Majumdar, H. Leung, B. J. Maundy „A Novel Chaos-based Transceiver Solution with High Data Rate Capability“

13. Francis C. M. Lau, Chi K. Tse „Performance of Chaos-based Digital Communication Systems in the Presence of a Pulsed-Noise Jammer“
14. Prieiga per internetą [žiūrėta 2011 05 15]: <http://www.dsplog.com/2008/06/10/ofdm-bpsk-bit-error/>
15. Prieiga per internetą [žiūrėta 2011 05 15]: <http://www.dsplog.com/2007/12/31/minimum-frequency-spacing-for-having-orthogonal-sinusoids/>
16. Prieiga per internetą [žiūrėta 2011 05 15]: <http://www.dsplog.com/2008/01/19/simulating-minimum-shift-keying-transmitter/>
17. Ctirad Kratochvíl, Pavel Heriban, Martin Houfek, Lubomír Houfek, Petr Šv'eda „Experimental Research of Chaos in Drive Systems“, Engineering MECHANICS, Vol. 16, 2009, No. 3, p. 239–248 239
18. Ctirad Kratochvíl, Lubomír Houfek, Martin Houfek, Jiří Krejsa, Petr Šv'eda „The Generators of Chaotic noise and its Usage in Modeling Dynamic Properties of Technical Systems“, Engineering MECHANICS, Vol. 17, 2010, No. 2, p. 123–133.
19. Nikolai F. Rulkov, Lev S. Tsimring „Synchronization Methods for Communication with Chaos Over Band-Limited Channels“, International Journal of Circuit Theory and Applications Int. J. Circ. theor. Appl., 27: 555–567 (1999)
20. Christopher P. Silva, Albert M. Young „Introduction to Chaos-Based Communications and Signal Processing“

PRIEDAS Nr. 1

```
function Txd = Transmitter( sBit, nBit, Precision, StartFreq, EndFreq, BitRate,
Tx )

% sBit      - length of one informational bit in samples
% nBit      - number of informational bits to simulate
% Precision - number of bits per sample
% StartFreq - the LoCut frequency of the bandpass filter
% EndFreq   - the HiCut frequency of the bandpass filter
% BitRate   - bitrate per second

% constants
SampleRate = BitRate * sBit;           % sample rate per second
sSim        = sBit * nBit;             % simulation length in samples
center      = pow2(Precision)/2-1;     % the center value of the sample scale

%-----Chaos generator-----
% calculate filter coefficients
[ch_k_b, ch_k_a] = cheby1( 5, 3, [StartFreq/SampleRate EndFreq/SampleRate] );

% generate random noise
chaos = rand( sSim, 1 ) - 0.5;

% filter
chaos = filter( ch_k_b, ch_k_a, chaos );

% calculate amplitude
ampl_n = pow2( Precision );
ampl_o = max(chaos)-min(chaos);

% scale
chaos = chaos * ( ampl_n / ampl_o );

% lift to the center
% chaos_min = min(chaos);
% chaos = chaos - chaos_min + (ampl_n/2);

fCh = round( chaos );

%----- Transmitter -----

% generate transmitter output
Txd = zeros(sSim,1);
i = 1;
for ii = 1 : nBit
    for iii = 1 : sBit
        if mod( i-1, sBit ) < sBit/2
            Txd(i) = fCh(i);
        else
            if Tx(ii)==1
                Txd(i) = fCh(i-sBit/2);
            else
                Txd(i) = - fCh(i-sBit/2);
            end;
        end;
        i = i + 1;
    end;
end;
end;
```

PRIEDAS Nr. 2

```
function Rx = Receiver( tBit, tSim, SampleRate, Precision, Rxd, Tx, fig )

%constants
sT      = length( Rxd );      % simulation length in samples
sB      = sT / (tSim/tBit);   % length of one bit in samples
BitCount = tSim/tBit;        % number of bits simulated

%----- Receiver -----

%ADC (normalization)

%calculate amplitude
ampl_n = pow2( Precision );
ampl_o = max(Rxd)-min(Rxd);

%scale
Rxd = Rxd * ( ampl_n / ampl_o );

%lift to the center
%Rxd_min = min(Rxd);
%Rxd = Rxd - Rxd_min;

%quantize
Rxd = round( Rxd );

%correlation with delayd itself
Rxc = CorrDelay( Rxd, sB/2 );

%calculate filter coeficients
%[ch_k_b,ch_k_a] = cheby1( 5,3, (sB*2)/sT );
%filter
%Rxc = filter( ch_k_b, ch_k_a, Rxc );

%quantise correlation result to get received bitstream
Rx = zeros( BitCount, 1 );
cmin = Rxc(1);
cmax = Rxc(1);
counter = 2;
for i = 1 : length(Rxc)
    if Rxc(i) < cmin
        cmin = Rxc(i);
    end;
    if Rxc(i) > cmax
        cmax = Rxc(i);
    end;
    level = 0;
    if mod( i, sB ) == 0
        if Rxc(i) > level
            Rx(counter) = 1;
        else;
            Rx(counter) = 0;
        end;
        counter = counter + 1;
    end;
end;

%quantise correlation result to get a nice received sequence - internal plots
Bitsrx = zeros( length(Rxc), 1 );
```

```

cmin = Rxc(1);
cmax = Rxc(1);
for i = 1 : length(Rxc)
    if Rxc(i) < cmin
        cmin = Rxc(i);
    end;
    if Rxc(i) > cmax
        cmax = Rxc(i);
    end;
    level = 0;
    if Rxc(i) > level
        Bitsrx(i) = 0.8;
    else;
        Bitsrx(i) = 0.2;
    end;
end;

%define the initial bit sequence for nice viewing - internal plots
Bitstx = zeros( sT, 1 );
i = 1;
for ii = 1 : BitCount
    for iii = 1 : sB
        Bitstx(i) = Tx(ii) * 0.6 + 0.2;
        i = i + 1;
        if i > sT
            break;
        end;
    end;
end;

%----- Plots -----
---

%values for the time axis - neded for internal plots
T = tSim/sT : tSim/sT : tSim;

%uncomment for internal plots
figure(fig);
nplots = 4;
subplot(nplots,1,1), plot(T,Bitstx);
subplot(nplots,1,2), plot(T,Rxd);
subplot(nplots,1,3), plot(Rxc);
subplot(nplots,1,4), plot(Bitsrx);

```

PRIEDAS Nr. 3. Raktų generavimas.

```
function [raktas1, raktas2, raktas3, raktas4, raktas5, raktas6, raktas7,
raktas8] = darvienas(x, r)

figure(1)
hold on
for x=2:.001:12;
    r=0.28;
    raktas1 = r+cos(x^2+2);
    plot(x, raktas1);
end
figure(2)
hold on
for x=2:.01:12;
    r=0.5;
    raktas2 = r+sin(x^2)^2 + cos(x^2);
    plot(x, raktas2);
end
figure(3)
hold on
for x=2:.01:12;
    raktas3 = sin(x) + cos(x/2);
    plot(x, raktas3);
end
figure(4)
hold on
for x=2:.01:12;
    raktas4 = (sin(x+2) + cos(x/2))^2;
    plot(x, raktas4);
end
figure(5)
hold on
for x=2:.01:12;
    raktas5 = (3*sin(x+3) + cos((x/2)-1))^2;
    plot(x, raktas5);
end
figure(6)
hold on
for x=2:.01:12;
    raktas6 = (sin(x+2) + cos(x/2))^2;
    plot(x, raktas6);
end
figure(7)
hold on
for x=2:.01:12;
    raktas7 = (sin(2*x+2) + cos(x/2))^2 + sin(x);
    plot(x, raktas7);
end
figure(8)
hold on
for x=2:.01:12;
    raktas8 = (sin(x+(pi/4)) + cos(x/5))^2 + sin(x)*cos(2*x);
    plot(x, raktas8);
end
```


PRIEDAS Nr. 4. Programos kodas.

```
% Siustuvas, imtuvas, BER
clear all
close all
rand('seed',777); % initializing the random state
N = 10^2; % number of bits
ipBin = rand(1,N) > 0.5; % binary 0's and 1's

fs = 10;
fc = 1;
T = 1;

ts = [0:1/fs:T]; % generating the sampling instants
ts = ts(1:end-1);
tsR = kron(ones(1,N),ts);

ip = 2*ipBin-1; % converting 0's to -1, 1's to 1

fm = ip/(4*T);
fmR = kron(fm,ones(1,fs)); % repeating

% Siustuvas, imtuvas, BER
clear all
close all
rand('seed',777); % initializing the random state
N = 10^1; % number of bits

ipBin = rand(1,N) > 0.5; % binary 0's and 1's

fs = 100;
fc = 1;
T = 1;
ts = [0:1/fs:T]; % generating the sampling instants
ts = ts(1:end-1);
tsR = kron(ones(1,N),ts);

ip = 2*ipBin-1; % converting 0's to -1, 1's to 1

fm = ip/(4*T);
fmR = kron(fm,ones(1,fs)); % repeating

% generating the phase
theta1 = pi/2*filter([0 1],[1 -1],ip);
thetaR1 = kron(theta1,ones(1,fs)); % repeating

xt1 = cos(2*pi*(fc+fmR).*tsR + thetaR1 );

figure(2)
hold on
for x=2:.001:12;
    r=0.28;
    raktas1 = r*cos(x^2+2);
    plot(x, raktas1);
end

figure(10)
ko1 = xt1*raktas1
plot(ko1)
```

```

% genertaing the phase
theta2 = pi*filter([0 1],[1 -1],ip);
thetaR2 = kron(theta2,ones(1,fs)); % repeating

xt2 = cos(2*pi*(fc+fmR).*tsR + thetaR2 );

figure(3)
hold on
for x=2:.01:12;
    r=0.5;
    raktas2 = r+sin(x^2)^2 + cos(x^2);
    plot(x, raktas2);
end

figure(11)
ko2 = xt2*raktas2
plot(ko2)

% genertaing the phase
theta3 = pi*filter([0 1],[1 -1],ip);
thetaR3 = kron(theta3,ones(1,fs)); % repeating

xt3 = cos(2*pi*(fc+fmR).*tsR + thetaR3 );

figure(4)
hold on
for x=2:.01:12;
    raktas3 = sin(x) + cos(x/2);
    plot(x, raktas3);
end

figure(12)
ko3 = xt2*raktas3
plot(ko3)

% genertaing the phase
theta4 = pi*filter([0 1],[1 -1],ip);
thetaR4 = kron(theta4,ones(1,fs)); % repeating

xt4 = cos(2*pi*(fc+fmR).*tsR + thetaR4 );

figure(5)
hold on
for x=2:.01:12;
    raktas4 = (sin(x+2) + cos(x/2))^2;
    plot(x, raktas4);
end

figure(13)
ko4 = xt4*raktas4
plot(ko4)

% genertaing the phase
theta5 = pi*filter([0 1],[1 -1],ip);
thetaR5 = kron(theta5,ones(1,fs)); % repeating

xt5 = cos(2*pi*(fc+fmR).*tsR + thetaR5 );

figure(6)
hold on

```

```

for x=2:.01:12;
    raktas5 = (3*sin(x+3) + cos((x/2)-1))^2;
    plot(x, raktas5);
end

figure(14)
ko5 = xt2*raktas5
plot(ko5)

% genertaing the phase
theta6 = pi*filter([0 1],[1 -1],ip);
thetaR6 = kron(theta6,ones(1,fs)); % repeating

xt6 = cos(2*pi*(fc+fmR).*tsR + thetaR6 );

figure(7)
hold on
for x=2:.01:12;
    raktas6 = (sin(x+2) + cos(x/2))^2;
    plot(x, raktas6);
end

figure(15)
ko6 = xt6*raktas6
plot(ko6)

% genertaing the phase
theta7 = pi*filter([0 1],[1 -1],ip);
thetaR7 = kron(theta7,ones(1,fs)); % repeating

xt7 = cos(2*pi*(fc+fmR).*tsR + thetaR7 );

figure(8)
hold on
for x=2:.01:12;
    raktas7 = (sin(2*x+2) + cos(x/2))^2 + sin(x);
    plot(x, raktas7);
end

figure(16)
ko7 = xt7*raktas7
plot(ko7)

% genertaing the phase
theta8 = pi*filter([0 1],[1 -1],ip);
thetaR8 = kron(theta8,ones(1,fs)); % repeating

xt8 = cos(2*pi*(fc+fmR).*tsR + thetaR8 );

figure(9)
hold on
for x=2:.01:12;
    raktas8 = (sin(x+(pi/4)) + cos(x/5))^2 + sin(x)*cos(2*x);
    plot(x, raktas8);
end

figure(17)
ko8 = xt8*raktas8
plot(ko8)

% plotting

```

```

figure(1)
plot(kron(ip,ones(1,fs)));
axis([0 N*fs -1.25 1.25]);grid on
Eb_N0_dB = [-3:15]; % multiple Eb/N0 values
nRx = [ko1 ko2 ko3 ko4 ko5 ko6 ko7 ko8]; % number of receive antenna
for jj = 1:length([ko1 ko2 ko3 ko4 ko5 ko6 ko7 ko8])
    for ii = 1:Eb_N0_dB

        n = 1/sqrt(2)*[randn(nRx(jj),N)]; % white gaussian noise, 0dB variance
on each row

        % Noise addition
y = ones(nRx(jj),1)*ip + 10^(-Eb_N0_dB(ii)/20)*n;

        % equalization
yHat = sum(y,1)/nRx(jj);

        % receiver - hard decision decoding
ipHat = real(yHat)>0;

        % counting the errors
nErr(jj,ii) = size(find([ipBin- ipHat]),2);
    end
    TheoryBerAWGN(jj,:) = 0.5*erfc(sqrt(nRx(jj)*10.^(Eb_N0_dB/10))); %
theoretical ber
end
SimBer = nErr/N; % Simulated ber

% plot
figure(10)
semilogy(Eb_N0_dB,TheoryBerAWGN(1,:), 'g-^', 'LineWidth', 2);
hold on
semilogy(Eb_N0_dB,SimBer(1,:), 'bx-', 'LineWidth', 2);
semilogy(Eb_N0_dB,TheoryBerAWGN(2,:), 'c-<', 'LineWidth', 2);
semilogy(Eb_N0_dB,SimBer(2,:), 'rd-', 'LineWidth', 2);
semilogy(Eb_N0_dB,TheoryBerAWGN(3,:), 'y->', 'LineWidth', 2);
semilogy(Eb_N0_dB,SimBer(3,:), 'mp-', 'LineWidth', 2);
semilogy(Eb_N0_dB,TheoryBerAWGN(4,:), 'b+-', 'LineWidth', 2);
semilogy(Eb_N0_dB,SimBer(4,:), 'ko-', 'LineWidth', 2);
semilogy(Eb_N0_dB,TheoryBerAWGN(5,:), 'b+-', 'LineWidth', 4);
semilogy(Eb_N0_dB,SimBer(5,:), 'ko-', 'LineWidth', 4);
semilogy(Eb_N0_dB,TheoryBerAWGN(6,:), 'b+-', 'LineWidth', 4);
semilogy(Eb_N0_dB,SimBer(6,:), 'ko-', 'LineWidth', 4);
semilogy(Eb_N0_dB,TheoryBerAWGN(7,:), 'b+-', 'LineWidth', 4);
semilogy(Eb_N0_dB,SimBer(7,:), 'ko-', 'LineWidth', 4);
semilogy(Eb_N0_dB,TheoryBerAWGN(8,:), 'b+-', 'LineWidth', 4);
semilogy(Eb_N0_dB,SimBer(8,:), 'ko-', 'LineWidth', 4);

axis([-3 12 10^-6 0.5])
grid on
legend('nRx=1 Theory', 'nRx=1 Sim', 'nRx=2 Theory', 'nRx=2 Sim', 'nRx=3
Theory', 'nRx=3 Sim', 'nRx=4 Theory', 'nRx=4 Sim', 'nRx=5 Theory', 'nRx=5 Sim', 'nRx=6
Theory', 'nRx=6 Sim', 'nRx=7 Theory', 'nRx=7 Sim', 'nRx=8 Theory', 'nRx=8 Sim');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for BPSK modulation in AWGN with receive diversity');

```