

Signalų analizė ir apdorojimas

Tadas Meškauskas

Vilniaus universitetas, Matematikos ir informatikos fakultetas,
Informatikos institutas

E-mail: tadas.meskauskas@mif.vu.lt

Atnaujinta 2021 m. sausio 28 d.

Turinys

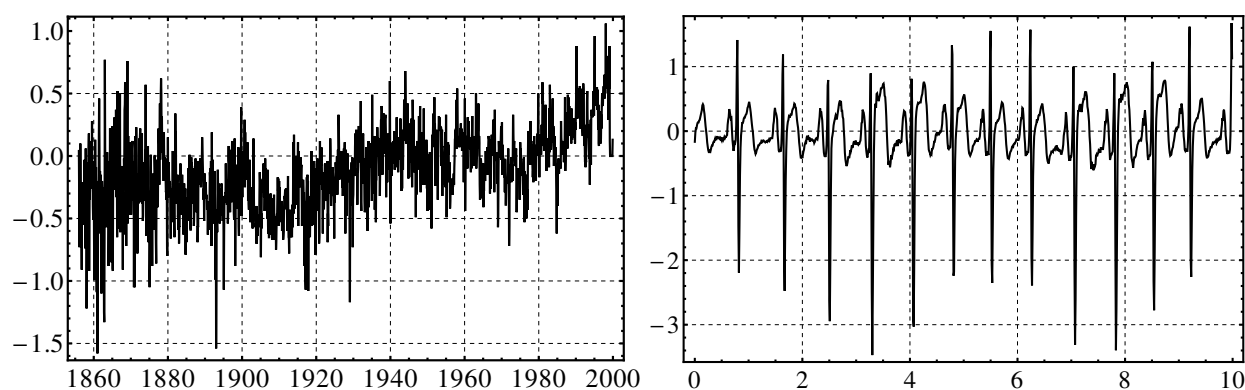
1	Signalų kilmė, klasifikacija ir analizės tikslai	3
1.1	Signalų kilmė	3
1.2	Signalų klasifikacija	5
1.2.1	Atsitiktiniai ir determinuoti signalai	5
1.2.2	Chaotiniai signalai	9
1.2.3	Stacionarūs signalai	10
1.2.4	Analoginiai ir skaitmeniniai signalai. Diskretizacijos dažnio pasirinkimas	11
1.3	Signalų analizės tikslai	14
2	Signalų glodinimo algoritmai	16
2.1	Signalų glodinimas	16
2.1.1	Slenkančio vidurkio algoritmas	17
2.1.2	Slenkančio vidurkio su svoriais algoritmas	20
2.1.3	Ekspontinis slenkančio vidurkio algoritmas	23
2.2	Sinchroninis triukšmų filtravimas periodiniams signalams	25
3	Signalų rekurentinės diagramos	28
3.1	Kiek sudėtingas yra tiriamas procesas?	28
3.2	Rekurentinės diagramos apibrėžimas ir savybės	29
3.2.1	Signalų būsenos sąvoka	31
3.2.2	Atstumo tarp dviejų būsenų skaičiavimo būdai	33
3.2.3	Rekurentinės diagramos savybės. Slenkstinio atstumo r parinkimas	34
3.3	Rekurentinių diagramų interpretavimas. Pavyzdžiai	36
4	Koreliacinis integralas ir koreliacinė dimensija	40
4.1	Laisvės laipsnių skaičiaus sąvoka	40
4.2	Signalų koreliacinis integralas	42
4.2.1	Koreliacinio integralo savybės	43
4.2.2	Koreliacinio integralo skaičiavimo optimizavimas	43
4.3	Signalų koreliacinė dimensija	44
4.3.1	Praktinės algoritmo detalės	45
4.3.2	Algoritmo parametrų D ir d pasirinkimas	49

5	Signalų koreliacija ir autokoreliacija	51
5.1	Vidurkis, dispersija ir koreliacijos koeficientas	51
5.2	Dviejų signalų tarpusavio koreliacija	55
5.2.1	Signalų tarpusavio padėties laike įtaka koreliacijai	55
5.2.2	Signalų tarpusavio koreliacijos funkcija	56
5.2.3	Tarpusavio koreliacijos grafiko interpretavimas. Pavyzdžiai	57
5.3	Signalų autokoreliacijos funkcija	59
5.3.1	Autokoreliacijos grafiko interpretavimas. Pavyzdžiai	61
6	Signalų komponentų išskyrimas. Furje skleidiniai	66
6.1	Furje skleidinio sąvoka ir pavyzdžiai	66
6.1.1	Furje hipotezė ir jos eksperimentinis patikrinimas	66
6.1.2	Lyginės ir nelyginės funkcijos	67
6.1.3	Furje skleidinio apibrėžimas. Signalų spektras	68
6.1.4	Furje koeficientų apskaičiavimas	70
6.1.5	Kai signalas apibrėžtas ne intervale $-\pi \leq t \leq \pi$	71
6.2	Furje skleidinys kompleksinių funkcijų bazėje	72
6.2.1	Kompleksinių skaičių aritmetika	72
6.2.2	Furje skleidinio kompleksinių funkcijų bazėje apibrėžimas	73
6.3	Kompleksinio spektro savybės	76
6.3.1	Paprastosios spektro savybės	76
6.3.2	Signalų, padauginto iš konstantos, ir dviejų signalų sumos spektras	77
6.3.3	Signalų energijos išsilaikymas spektre (Parsevalio teorema)	77
6.3.4	Baigtinio Furje skleidinio paklaida. Gibbs'o efektas	78
7	Diskrečioji Furje transformacija. Furje filtras	82
7.1	Skaitmeninio signalų spektras. Diskrečioji Furje transformacija	82
7.1.1	Diskrečiosios Furje transformacijos (ir jai atvirkštinės) apibrėžimai	82
7.1.2	Diskrečiosios Furje transformacijos savybės	85
7.1.3	Spekto ir realių dažnių ryšys	88
7.1.4	Konvoliucijos (sąsukos) ir koreliacijos ryšys su Furje transformacija	88
7.2	Greitoji Furje transformacija	90
7.2.1	Greitosios ir diskrečiosios Furje transformacijų palyginimas	90
7.2.2	Cooley–Tukey greitosios Furje transformacijos algoritmas	92
7.2.3	Cooley–Tukey algoritmo optimizavimas	94
7.3	Dažnių komponentų filtravimas Furje filtru	97

1 Signalų kilmė, klasifikacija ir analizės tikslai

1.1 Signalų kilmė

Tiek kasdieniniame gyvenime, tiek technologijų pasaulyje – pastebėdami tai ar nepastebėdami, nuolat susiduriame su įvairiausiais reiškiniais, kurie vadinami signalais. Kai kurie signalai gali įspėti mus apie gendančią mašiną, pavyzdžiui cypimas ar tuksenimas (jų priežastimi gali būti atsipalaidavęs varžtas arba sulūžęs guolis). Kiti signalai suteikia galimybę spėti kas mūsų laukia ateityje – meteorologai prognozuoja orus, remdamiesi turimais stebėjimais, ekonomistai vertina būsimą pelną ar nuostolius remdamiesi sukauptais kainos, apyvartos, infliacijos bei kitais duomenimis. Gydytojai vertina žmogaus sveikatos būklę, analizuodami elektrokardiogramą, encefalogramą ir kitus medicinines kilmės signalus. 1 pav. pateikti realiuose stebėjimuose registruotų signalų pavyzdžiai.



1 pav.: Iliustracija kairėje: vidutinė oro temperatūra Šiaurės pusrutulyje (abscisių ašyje atidėti metai), pagal [7]. Iliustracija dešinėje: vieno iš encefalogramos kanalų fragmentas (abscisių ašyje atidėtos sekundės), pagal [8].

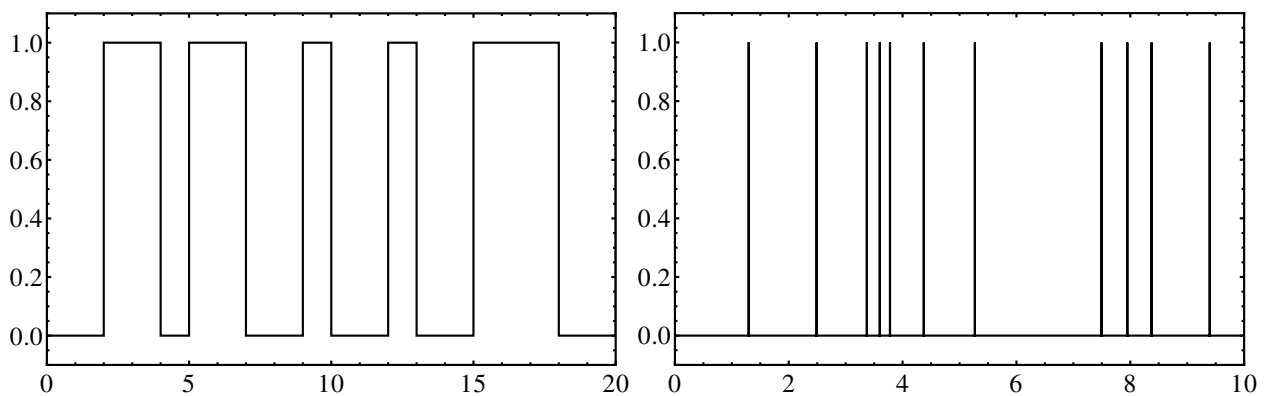
Signalas – tai fizikinis dydis, dažniausiai kintantis laike ir reprezentuojantis tam tikrą informaciją. Tiriant signalus labai svarbu suvokti su koku būtent fizikiniu dydžiu ir kokiame laiko kintamojo mastelyje dirbame. Kartais signalu vadinama priklausomybė ne nuo laiko kintamojo, bet nuo kurio nors kito fizikinio parametro. Pavyzdžiui, reljefo signaluose (tarkime, tiriamo paviršiaus nelygumus) registruojama priklausomybė nuo koordinatės kintamojo. Mes laikysime, kad tiriamo signalo argumentas yra laikas, kurį žymėsime t . Tokio tipo signalai kaip garsas, temperatūra, tekančios srovės stiprumas, kraujo spaudimas ir pan. gali būti stebimi ir registruojami, panaudojant tam tikslui skirtus prietaisus. Tačiau egzistuoja ir sunkiai registruojami signalai, pavyzdžiui – skonis, kvapas. Kai kurių signalų užregistravimas iš viso techniškai neįmanomas.

Signalai ir jų apdorojimas domina medikus (elektrokardiogramos, encefalogramos, elektroneurogramos, kraujo spaudimo priklausomybės ir t. t.), finansininkus ir marketologus (valiutų kursai, akcijų kainos, importo ir eksporto apimtys, kompanijų pelno rodikliai, reklamos išlaidų ir gauto pelno santykis ir t. t.), meteorologus (iškritusių kritulių, vėjo krypties ir intensyvumo, temperatūros, santykinio oro drėgnumo momentinės ir vidutinės vertės ir t. t.), inžinierius, geofizikus, seismologus, astronomus, audio ir video medžiagos įrašų specialistus, sociologus, demografus, netgi politikus (asmenybių ir partijų populiarumas priklausomai nuo laiko, likusio iki rinkimų ir pan.) bei daugelio kitų sričių atstovus. Knygose [1, 2, 3] aprašyti daugelis skaitmeninio signalų apdorojimo algoritmų.

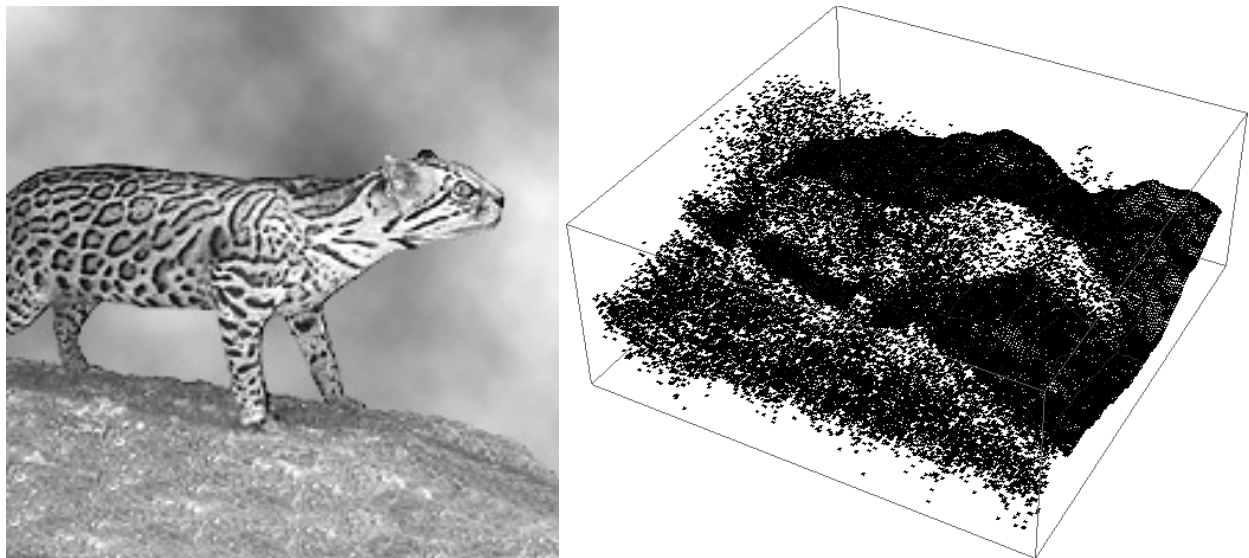
Įvairios kilmės testinių taikomųjų signalų duomenų galima nemokamai atsisiųsti, pavyzdžiui, iš [4, 5, 6, 7, 8, 9, 10] ir daugelio kitų šaltinių.

Tam tikruose taikymuose sutinkami ypatingo formato signalai. Tokių signalų pavyzdžiais gali

būti dvejetainis procesas ir taškinis procesas (žr. 2 pav.). Dvejetainiu procesu vadinamas signalas, įgyjantis tik dvi reikšmes (tarkime, turime jungiklį, kuris gali būti įjungtas arba išjungtas), dažniausiai koduojamas 0 ir 1. Dvejetainiai procesai dažni komunikacijų inžinerijoje. Kitas specialaus tipo signalo pavyzdys – taškinis procesas, taip pat įgyjantis tik dvi reikšmes. Šis signalas būna “išjungtas” beveik visą laiką ir “įsijungia” tik mirksniui, kada įvyksta kažkoks mus dominantis įvykis (pavyzdžiui, susitraukia širdies skilveliai, arba per sankryžą pravažiuoja automobilis). Taigi, taškiniame procese svarbios ne pačios signalo reikšmės, bet laiko momentai, kada įvyko įvykis. Tolesnėje analizėje į specialaus tipo signalus nesiorientuosime.



2 pav.: Iliustracija kairėje: dvejetainis procesas. Iliustracija dešinėje: taškinis procesas.



3 pav.: Iliustracija kairėje: kompiuterinis vaizdas. Iliustracija dešinėje: to paties vaizdo taškų (pikselių) intensyvumo funkcija – dvimatis signalas.

1 pav. ir 2 pav. pateikti signalų pavyzdžiai, kaip ir absoliuti dauguma taikymuose analizuojamų signalų priklauso nuo vienintelio nepriklausomo kintamojo (laiko argumento t). Tačiau nesunku pateikti ir signalų, priklausančių nuo dviejų ar daugiau kintamųjų pavyzdžių – kad ir kompiuterinio vaizdo signalas. Nagrinėdami rasterizuotą vaizdą kaip kiekvieno jį sudarančio taško (pikslio) intensyvumą, gauname nuo dviejų koordinatų (taško adresų pagal horizontalę ir pagal vertikale) priklausančią intensyvumo funkciją – dvimatį signalą (žr. 3 pav.). Taigi, vaizdai gali būti sutapatinti su dvimačiais signalais. Vienmačiai signalai gali būti gaunami kaip dvimačių signalų pjūviai. Toliau tirsime tik vienmačius signalus.

Matematiškai vienmačius signalus galima susieti su nuo laiko kintamojo priklausančia funkcija

$$f(t), \quad t \geq 0.$$

Praktiškai tiek eksperimentuose, tiek rašydami kompiuterines programas, funkciją $f(t)$ pakeičiame jos reikšmių masyvu

$$f(t_0), f(t_1), \dots, f(t_N), \quad \text{arba trumpai} \quad f_i, \quad i = 0, 1, \dots, N, \quad (1)$$

čia naudojame pažymėjimą

$$f_i = f(t_i).$$

Pavyzdžiui, matuojame oro temperatūrą kas tris valandas ir išsaugome gautus skaičius – signalo $f(t)$ reikšmes. Žinoma, eksperimentuose beveik visada susiduriame su matavimų paklaidomis (kurios gali būti didesnės ar mažesnės, priklausomai nuo naudojamų matavimo prietaisų kokybės ir darbuotojų kvalifikacijos), taigi šios reikšmės yra apytikslės. Kartais paklaidas galime įvertinti, kartais ne. Detaliau su signalų diskretizavimu susijusius klausimus aptarsime 1.2.4 skyrelyje.

Kartais, nors ir pakankamai retai, analizuojamas signalas $f(t)$ gaunamas ne eksperimentiniu, bet matematiniu keliu, pavyzdžiui išsprendus tam tikrą matematinį modelį (dažniausiai diferencialinę lygtį) arba tiesiog užduotas konkrečia formule, sakykime $f(t) = 2 \sin(2\pi t + \pi/6)$. Tokia situacija sutinkama nagrinėjant mus dominančius procesus teoriškai (jeigu jų dinamikos teorija egzistuoja). Vis dėlto, gyvenime vykstantys procesai labai sudėtingi – kaip taisyklė juos gerai aprašančių teorijų paprasčiausiai nėra ir lieka eksperimentiniai analizės metodai.

Nepriklausomai nuo to ar iš karto (iš eksperimento) turime signalo reikšmių masyvą, ar diskretizuojame matematiškai apibrėžtą funkciją, kalbėdami apie algoritmus operuosime tik signalo reikšmių seka (1). Kadangi dirbama su diskrečiomis signalo reikšmėmis, pamatuotomis konkrečiais (dažniausiai iš anksto pasirinktais) laiko momentais, literatūroje signalai dar vadinami *laiko eilutėmis* (angliškai “time series”).

1.2 Signalų klasifikacija

1.2.1 Atsitiktiniai ir determinuoti signalai

1 pav. pavaizduoti signalai pasižymi svarbia savybe: net ir disponuojant signalo reikšmėmis ilgame laiko intervale, neįmanoma tiksliai prognozuoti kokia bus signalo reikšmė nurodytu laiko momentu ateityje. Pavyzdžiui, negalime žinoti kokia bus tiksli oro temperatūra ateinančiais metais, nors ir turime daugiamečių stebėjimų duomenis. Signalai, pasižymintys šia savybe, vadinami *atsitiktiniais* arba *stochastiniais*.

Tačiau egzistuoja ir visiškai nuspėjami signalai. Pavyzdžiui, kamertono virpėjimas visada generuoja to pačio dažnio garso bangą, nepriklausomai nuo to kaip ilgai jo beklausytume. Iš fizikos žinome, kad kamertono skleidžiama banga yra išreiškiama periodine trigonometrine funkcija (2), kuri bus užrašyta toliau. Išmatavę šią bangą keletu laiko momentų, galime vienareikšmiškai apskaičiuoti jos vertes bet kuriuo laiko momentu. Tokie signalai – kurie gali būti prognozuojami kaip norima ilgame laiko intervale, vadinami *determinuotais*. Determinuotais signalais vadinami ir sudėtingesnės nei periodinės, bet žinomos priklausomybės signalai. Taip pat determinuoti signalai gali būti gaunami išsprendus tam tikrą matematinį modelį, su sąlyga, kad šis matematinis modelis nepriklauso nuo atsitiktinių dydžių ir nėra chaotinis. Apie chaotinius matematinius modelius žr. 1.2.2 skyrelį.

Kai kuriais atvejais galime turėti *mišrų* signalą, kurio fragmentai (tam tikruose laiko intervaluose) yra determinuoti (pavyzdžiui periodiniai), o kartais signalas pereina į atsitiktinį režimą.

Determinuotų signalų pavyzdžiai

Kokie būna determinuoti signalai? Be abejo, egzistuoja labai didelė tokių signalų įvairovė ir visų jų išvardinti neįmanoma. Apžvelgsime keletą dažniausiai pasitaikančių determinuotų signalų tipų ir iliustruosime juos pavyzdžiais.

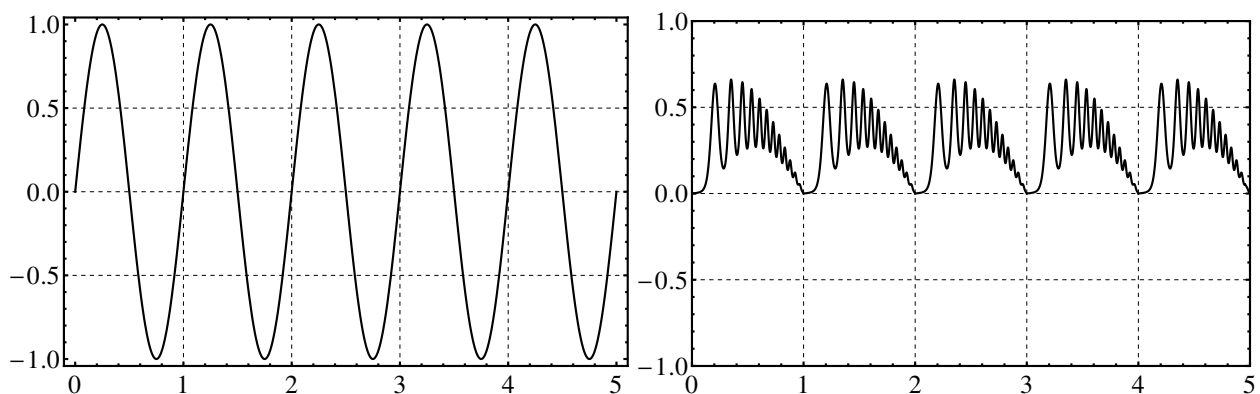
Vienu iš determinuoto signalo pavyzdžiu gali būti jau paminėta (kalbant apie kamertoną) sinusinė banga (žr. 4 pav., iliustracija kairėje)

$$f(t) = A \sin(2\pi ft + \varphi), \quad t \geq 0. \quad (2)$$

Čia $A \geq 0$ yra signalo dydį apibrėžiantis amplitudės koeficientas, o φ – pradinė fazė. f yra bangos dažnis, matuojamais hercais (Hz). Vienas hercas atitinka vieną virpesį per sekundę, kitaip tariant – bangą, kuri daro vieną ciklą per sekundę. Vienas kilohercas (kHz) yra lygus tūkstančiui hercų, o vienas megahercas (MHz) – milijonui hercų. Kartais vietoje dažnio f naudojamas kampinis dažnis

$$\omega = 2\pi f.$$

Tarkime, turime vieno metro ilgio strypą, besisukantį pastoviu greičiu apie vieną iš itvirtintų galų. Tuomet, prisiminę apskritimo ilgio formulę $L = 2\pi r$ (kur L yra apskritimo ilgis, o r – apskritimo spindulys), matome, kad ω atitinka judančio strypo galo nueitą kelią per vieną sekundę. Dydžiai $2\pi ft$ ir φ turi kampo, matuojamo radianais (rad), prasmę.



4 pav.: Periodinių signalų (su periodu $T = 1$) pavyzdžiai. Iliustracija kairėje: sinusinė banga (2) su $\varphi = 0$. Iliustracija dešinėje: sudėtingesnis periodinis signalas.

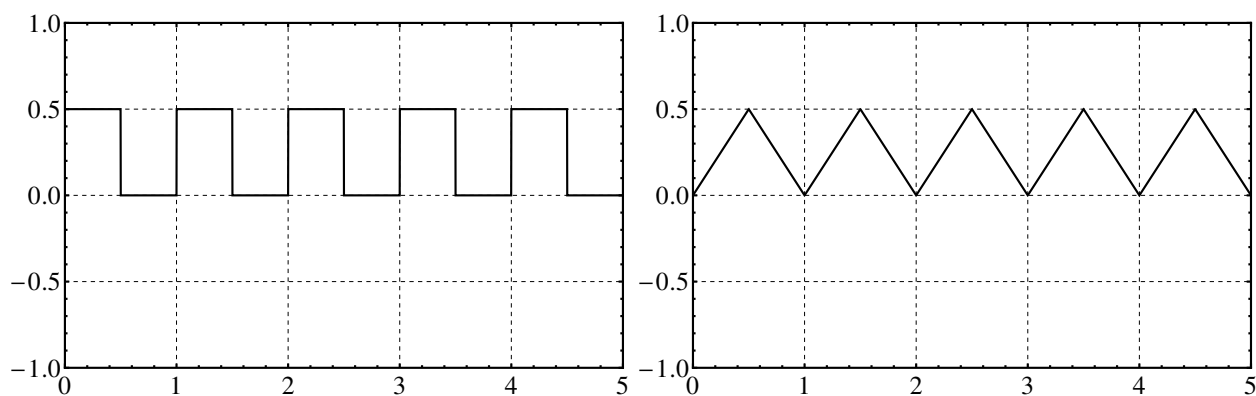
Pastebėkime, kad signalo, išreikšto formule (2), forma pasikartoja kas laiką $T = 1, 2T = 2, 3T = 3$ ir t. t. Analogiškai sinusinei bangai signalai, atkartojantys savo formą kas tam tikrą laiko tarpą T , vadinami *periodiniais*, o T vadinamas periodu. Periodinių signalų pavyzdžiai pateikti 4 pav.

Pastūmę laiką periodinį signalą į kairę arba į dešinę per sveiką periodų skaičių gausime tą patį signalą:

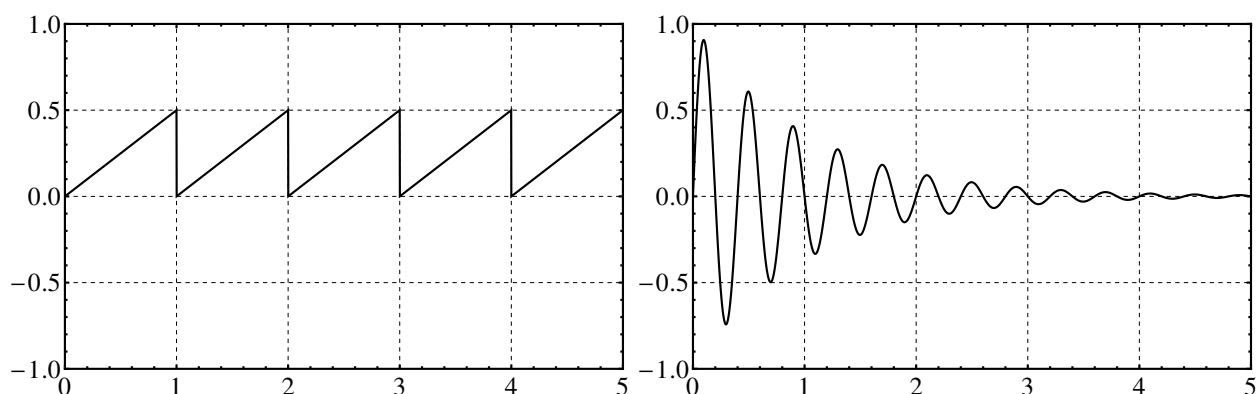
$$f(t + kT) = f(t), \quad k = 0, \pm 1, \pm 2, \dots$$

Vadinasi, bet kuris periodinis signalas yra determinuotas.

Beje, periodinis (su periodu T) signalas taip pat turi periodus, lygius $2T, 3T$ ir t. t. Pats trumpiausias periodinio signalo periodas vadinamas pagrindiniu periodu.



5 pav.: Periodinių signalų (su periodu $T = 1$) pavyzdžiai. Iliustracija kairėje: stačiakampis signalas. Iliustracija dešinėje: trikampis signalas.



6 pav.: Iliustracija kairėje: pjūklo pavidalo periodinis signalas (su periodu $T = 1$). Iliustracija dešinėje: gęstantis signalas.

Pateiksime dar keletą taikymuose pasitaikančių (bent jau apytiksle prasme) periodinių signalų pavyzdžių. 5 pav. pavaizduoti schematiniai stačiakampio signalo ir trikampio signalo grafikai, o 6 pav. iliustracijoje kairėje – pjūklo pavidalo signalo schematinis grafikas.

Periodinį signalą skleidžiančio šaltinio energija nuolat periodiškai atsinaujina. Tačiau tai nebūdinga uždarams energijos požiūriui sistemoms. Jeigu šaltinio energija yra baigtinė, tai signalo reikšmės neišvengiamai ilgainiui artėja prie nulio. Tokie signalai vadinami *gęstančiais*. Gęstančio signalo pavyzdys pateiktas 6 pav. iliustracijoje dešinėje. Periodiniai signalai nėra gęstantys.

Pasirodo, ne visi iš determinuotųjų matematinių modelių (kurių formulavime nėra jokių atsitiktinių faktorių) gaunami signalai yra determinuoti – prognozuojami kaip norima ilgame laiko intervale. Šį įdomų pastebėjimą detaliau aptarsime 1.2.2 skyrelyje, kalbėdami apie chaotinius signalus.

Sintetinės kilmės stochastinių signalų programavimas. Atsitiktinių dydžių generavimas

Kartais kyla poreikis sugeneruoti tam tikromis statistinėmis savybėmis pasižymintį signalą kompiuteriu. Tokie signalai vadinami *sintetinės kilmės* signalais arba signalų *simuliacijomis*, kadangi jie nėra jokio realioje gyvenime vykstančio proceso eksperimentiniai duomenys, o tik jų matematinė imitacija.

Sintetinės kilmės signalus galima lengvai ir greitai generuoti, tuo tarpų realių eksperimentinių duomenų surinkimas reikalauja nepalyginamai daugiau laiko ir kitų išteklių. Tokie sintetiniai signalai gali būti ir yra naudojami akademiniams tikslams, matematinių modelių palyginimui su eksperimentiniais rezultatais, kompiuterinių algoritmų realizacijų testavimui. Pavyzdžiui, labai patogu

kompiuterinės simuliacijos būdu imituoti įvairius triukšmus, pasižyminčius reikiamomis savybėmis.

Sintetiniai signalai gaunami iš tam tikros matematinės formulės arba išsprendus duotą matematinį modelį (pavyzdžiui, rekurentinius sąryšius, diferencialinę lygtį ir pan.). Matematinė formulė arba modelis gali būti determinuoti (be atsitiktinai kintančių dydžių). Tokiu atveju tiesiog užprogramuojame šią formulę arba algoritmą duotam modeliui spręsti.

Tačiau dažnai reikia programuoti signalų simuliacijas, kuriose yra vienas ar daugiau atsitiktinai kintantys dydžiai. Pavyzdžiui, simuliuojant triukšmą, atsitiktinai klaidžiojančios dalelės judėjimą, miesto transporto atvykimo laikus ir t. t.

Trumpai aptarsime dažniausiai sutinkamų atsitiktinių dydžių: *standartinio normaliojo* atsitiktinio dydžio ir *Puasono* atsitiktinio dydžio kompiuterinio generavimo metodus.

Laikysime, kad žinome kaip kompiuteriu gauti pseudoatsitiktinį skaičių su vienoda tikimybe įgyjantį bet kurį intervalo $[0, 1)$ tašką (dar vadinamą atsitiktiniu dydžiu, tolygiai pasiskirsčiusiu intervale $[0, 1)$). Tokius skaičius žymėsime u_1, u_2 , ir taip toliau. Visos programavimo kalbos ir aplinkos turi priemones tokiems skaičiams gauti.

Reikėtų pasakyti, kad dauguma programavimo kalbų ir aplinkų instrumentų šiuos pseudoatsitiktinius skaičius generuoja gana primityviai, todėl jų atsitiktinumas yra paviršutiniškas. Todėl rimtuose taikymuose reikėtų naudoti specialias programų bibliotekas, taikančias specializuotus algoritmus (pavyzdžiui, “Mersenne twister” metodą) kokybiškų pseudoatsitiktinių skaičių u_1, u_2 , ir t. t. gavimui.

Standartinis normalusis atsitiktinis dydis $X \sim N(0, 1)$ arba atsitiktinis dydis iš standartinio (su vidurkiu, lygiu 0 ir dispersija, lygia 1) normaliojo pasiskirstymo (dar vadinamo Gauso pasiskirstymu, angliškai “normal distribution”, “Gaussian distribution”) labai dažnai naudojamas triukšmo (signalų reikšmėse atsirandančių paklaidų) generavimui, oro temperatūros, žmonių ūgio, svorio svyravimams simuliuoti ir daugelyje kitų pavyzdžių, kur reikia generuoti atsitiktinius skaičius su žinomu vidurkiu.

Puasono atsitiktinis dydis $X \sim P(\lambda)$ arba atsitiktinis dydis iš Puasono pasiskirstymo (angliškai “Poisson distribution”) su parametru λ naudojamas aptarnavimo teorijos modeliuose, pavyzdžiui atsitiktinai generuojant laiką, kurį klientui reikės laukti kirpykloje, poliklinikoje, autobuso stotelėje ir pan., kai žinoma statistiškai vidutinė laukimo trukmė $\lambda > 0$. Taigi, parametras $\lambda > 0$ nusako Puasono atsitiktinio dydžio vidurkį, taip pat ir dispersiją (šiam pasiskirstyme vidurkis ir dispersija sutampa). Reikia pažymėti, kad Puasono atsitiktinis dydis įgyja tik neneigiamas sveikąsias reikšmes.

Metodas standartinio normaliojo atsitiktinio dydžio generavimui. Tarkime, kad $u_1, u_2 \in [0, 1)$ yra tarpusavyje nepriklausomi, pseudoatsitiktiniai skaičiai, tolygiai pasiskirstę intervale $[0, 1)$. Tuomet pseudoatsitiktinis skaičius ξ , apskaičiuotas pagal formulę

$$\xi = \cos(2\pi u_1) \sqrt{-2 \ln u_2},$$

bus iš standartinio (su vidurkiu, lygiu 0 ir dispersija, lygia 1) normaliojo pasiskirstymo.

Jeigu reikia simuliuoti normalųjį atsitiktinį dydį su vidurkiu m ir dispersija $d > 0$, taikome transformaciją $\xi \rightarrow \sqrt{d} \xi + m$.

Pavyzdys. Sugeneruosime sintetinį atsitiktinai klaidžiojančios dalelės signalą (dar vadinamą Brauno judėjimo, angliškai “Brownian motion”, signalu):

$$f_{i+1} = f_i + \xi_i, \quad f_i = 0, \quad i = 0, 1, \dots,$$

čia ξ_0, ξ_1, \dots yra nepriklausomi (kiekvieną jų generuojame iš naujo) atsitiktiniai skaičiai iš standartinio normaliojo pasiskirstymo. Taigi, signalo reikšmės gaunamos rekurentiškai: nauja reikšmė apskaičiuojama prie buvusios reikšmės pridėjus atsitiktinumą. Su Brauno judėjimo signalu susiję pavyzdžiai bus pateikti toliau, žr. 8 pav., 30 pav.

Metodas Puasono atsitiktinio dydžio generavimui. Tarkime, kad $u_1, u_2, \dots \in [0, 1)$ yra tarpusavyje nepriklausomi, pseudoatsitiktiniai skaičiai, tolygiai pasiskirstę intervale $[0, 1)$. Skaičiuojame sandaugas: $u_1, u_1u_2, u_1u_2u_3$ ir t. t., kol sandauga tampa mažesne už $e^{-\lambda}$. Tuomet dauginamųjų skaičius n priešpaskutinėje sandaugoje kurioje $e^{-\lambda}$ vis dar buvo viršijamas: $u_1u_2 \cdots u_n \geq e^{-\lambda}$ ir yra pseudoatsitiktinis skaičius iš Puasono pasiskirstymo su parametru $\lambda > 0$. Šio algoritmo pseudokodas:

```

 $a = e^{-\lambda};$ 
 $r = 1;$ 
 $n = -1;$ 
while  $r > a$  do
    sugeneruojame pseudoatsitiktinį  $u \in [0, 1)$ ;
     $r = r * u;$ 
     $n = n + 1;$ 
end while
return  $n;$ 

```

Atsitiktinių dydžių iš įvairių pasiskirstymų (tame tarpe ir mūsų nagrinėtų) kompiuterinis generavimas taip pat įmanomas naudojantis tam tikslui skirtais kompiuteriniais įrankiais.

1.2.2 Chaotiniai signalai

Gali pasirodyti, kad visi iš determinuotų matematinių modelių gauti signalai gali būti prognozuojami kaip norima ilgame laiko t intervale. Tačiau šis teiginys nevisada teisingas! Egzistuoja determinuoti matematiniai modeliai (neturintys jokių atsitiktinių faktorių), pasižymintys jautria priklausomybe nuo pradinių reikšmių, kurių sprendiniai dėl mažiausių fluktuacijų (kad ir apvalinimo paklaidų, kurios neišvengiamos skaičiuojant kompiuteriu) tampa neprognozuojamais. Pasižymintys šia savybe matematiniai modeliai vadinami chaotiniais modeliais. Chaotinio modelio pavyzdžiu gali būti logistinis atvaizdis

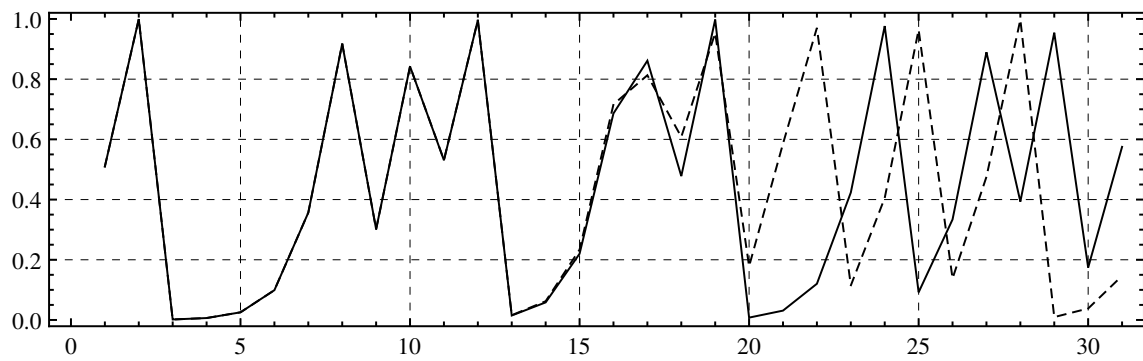
$$x_{n+1} = 4x_n(1 - x_n), \quad x_0 - \text{duotas skaičius iš intervalo } x_0 \in [0, 1], \quad n = 0, 1, \dots, \quad (3)$$

kurio tikslus sprendinys yra realiųjų skaičių seka x_n , apibrėžta formule

$$x_n = \sin^2(2^n \arcsin \sqrt{x_0}), \quad n = 0, 1, \dots \quad (4)$$

Matome, kad šioje sekoje labai greitai (dėl itin sparčiai augančio daugiklio 2^n , kai $n \rightarrow \infty$) kaupiasi paklaida, su kuria kompiuteryje priskiriame x_0 reikšmę. Taigi, jei nežinotume tikslios sprendinio išraiškos (4), kompiuteriu skaičiuodami seką x_n tiesiogiai iš matematinio modelio (3), pakankamai dideliems n dėl apvalinimo paklaidų gautume visiškai neprognozuojamus skaičius. Šio efekto demonstracija pateikta 7 pav. Taigi, kompiuterinio modeliavimo požiūriu, determinuotas matematinis modelis (3) generuoja neprognozuojamą signalą. Tokie signalai vadinami *chaotiniais*.

Nors pateiktas chaotinio modelio pavyzdys dirbtinis, chaotinės sistemos mūsų gyvenime jokių būdų nėra retai sutinkamos. Tarkime – važiuojame automobiliu. Vienas mažytis netikslus judesys ir gali įvykti avarija, daugelio žmonių gyvenimas gali susiklostyti pagal visiškai kitą scenarijų. Loterijose kiekvieną kartą “laimingais” tampa vis kiti skaičiai, nors naudojamas tas pats lošimo būgnas, tie patys kamuoliukai. Vis dėlto tai nereiškia, kad chaotinės sistemos negali būti modeliuojamos ir netgi kažkiek prognozuojamos. Įsivaizduokite didelį biljardo stalą, šiek tiek brokuotą – su vienu nelygumu. Ant šio nelygumo “užlipęs” rutulys gali labai netikėtai pakeisti trajektoriją. Tačiau juk taip atsitiks nevisada, dažniausiai jam pavyks šio nelygumo išvengti. Kol rutulys nepriartės prie šios



7 pav.: Logistinio atvaizdžio (3) kompiuterinių sprendinių (skaičiuojant dvigubu slankaus kabelio aritmetikos tikslumu) priklausomybės nuo n . Matome, kad du artimi sprendiniai, kurių pradinė reikšmė $x_0 = 0.51$ (ištiesinė kreivė) ir $x_0 = 0.51 + 10^{-6}$ (punktyrinė kreivė) vis labiau ir labiau išsiskiria: kai $n > 16$ šis skirtumas tampa matomas plika akimi ir toliau didėja.

brokuotos stalo vietos, žaidėjas galės visai neblogai įsivaizduoti (prognozuoti) tolimesnę trajektoriją. Tai reiškia, kad, pavyzdžiui, dažnai įmanoma patikimai prognozuoti orus trumpam laikui (vienai ar dviem dienoms), tačiau kuo labiau ilgalaikė prognozė, tuo ji nepatikimesnė. O retsykliais nutiks taip, kad ir trumpalaikė prognozė bus visai nevykusi.

Chaotiniai signalai nėra determinuoti (prognozuojami), taigi jie patenka į atsitiktinių signalų klasę. Iš tiesų, daugelis eksperimentuose registruojamų atsitiktinių signalų yra chaotinės kilmės, nors tikslūs (jie būtų labai sudėtingi) jų matematiniai modeliai ir nėra žinomi. Signalų duomenų chaotiškumo laipsniui įvertinti skaičiuojamas Liapunovo rodiklis ir kiti dydžiai, tačiau mes šiame kurse apie tai nekalbėsime.

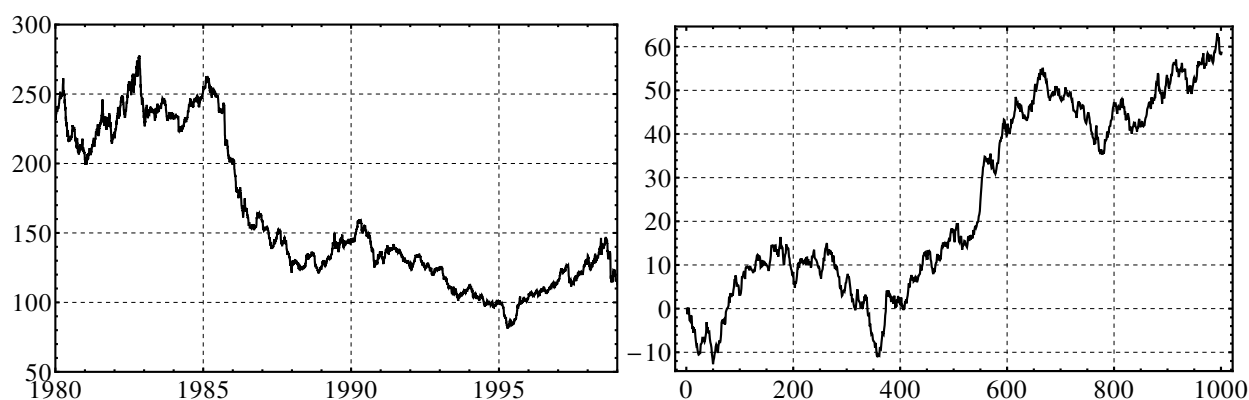
1.2.3 Stacionarūs signalai

Nepaisant to, kad atsitiktiniai signalai nėra prognozuojami, vis dėlto dažnai (bet ne visada) žinomos kai kurios statistinės jų savybės, pavyzdžiui vidutinė signalo vertė, apie kurią jis svyruoja. Pavyzdžiui, tirdami medicininės kilmės signalus (elektrokardiogramas, encefalogramas, žr. pvz, 1 pav. iliustraciją dešinėje) pastebime, kad atskiri signalo fragmentai tarpusavyje panašūs. Šiuo atveju panašumą suprantame ta prasme, kad bėgant laikui signalo reikšmių vidurkis išlieka beveik nepakitęs; tą patį galima pasakyti ir apie signalo reikšmių dispersiją.

Tuo tarpu kiti signalai (ekonominės, meteorologinės ir kt. kilmės), tarkime Lietuvos bendrojo vidutinio produkto statistiniai duomenys, nepasižymi nusistovėjusiu vidurkiu ir dispersija. Šalies bendrasis vidutinis produktas turi tendenciją augti, išskyrus krizės laikotarpius. Panašūs pastebėjimai galimi nagrinėjant 1 pav. iliustracijoje kairėje pavaizduotą vidutinės metinės oro temperatūros Šiaurės pusrutulyje signalą.

Taigi, signalus galima klasifikuoti į tokius, kurių tam tikros statistinės savybės (vidurkis, dispersija) nekinta bėgant laikui – *stacionariusius* signalus ir tokius, kurių savybės keičiasi bėgant laikui – *nestacionariusius* signalus. Tokia yra stacionaraus signalo samprata plačiąja (kokybine) prasme, nors tikslus matematinis apibrėžimas būtų techniškai sudėtingesnis – paprastumo dėlei jo nepateiksime.

Praktikoje sutinkami signalai paprastai priklauso vienam iš trijų tipų: a) signalai, pasižymintys stacionarumo savybe ilguose laiko intervaluose (dauguma medicininių signalų, elektrinių prietaisų skleidžiami triukšmai ir kt.); b) signalai, pakankamai stacionarūs trumpuose laiko intervaluose (pvz. atmosferos turbulentiškumo matavimai, daugelis meteorologinių ir ekonominių signalų); c) absoliučiai nestacionarūs signalai, kurių matomos savybės nuolat kinta laike.



8 pav.: Nestacionarių signalų pavyzdžiai. Iliustracija kairėje: JAV dolerių keitimo kurso į Japonijos jenas svyravimas. Kasdieniniai duomenys nuo 1979 m. gruodžio 31 d. iki 1998 m. gruodžio 31 d., pagal [4]. Iliustracija dešinėje: Brauno judėjimo (atsitiktinai klaidžiojančios dalelės judėjimo, angliškai “Brownian motion”) signalas.

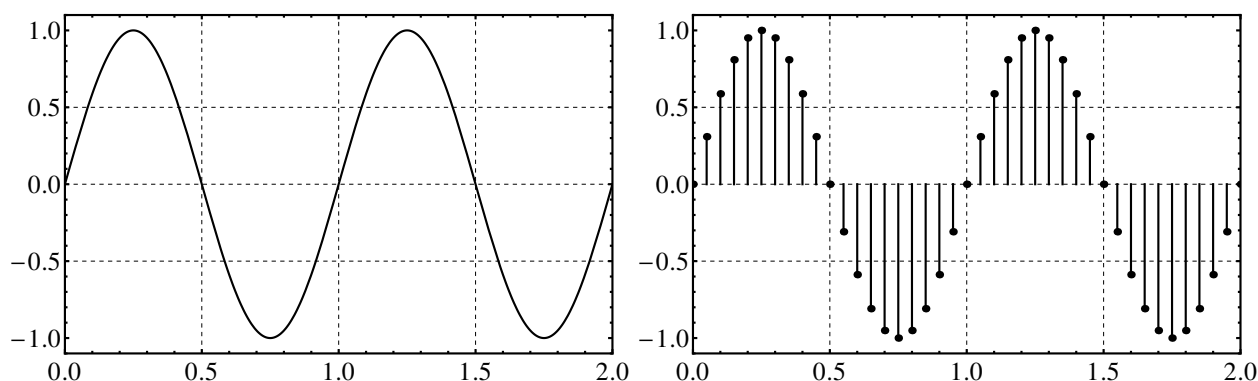
Nestacionarių signalų grafikų pavyzdžiai pateikti 8 pav.

Jeigu turime keletą stacionarių signalų, gautų stebint panašius, bet ne identiškus šaltinius, gali būti svarbu palyginti jų pastoviuosius statistinius dydžius (vidurkį, dispersiją ir kt.). Nestacionariųjų signalų atveju tiesiogiai to padaryti neišeina. Tačiau, jeigu nestacionarusis signalas yra pakankamai paprastas, pavyzdžiui, yra stacionaraus signalo ir tiesinės laiko t funkcijos (trendo) suma, galima pritaikyti nestacionarumą eliminuojančią transformaciją ir toliau apdoroti tik stacionariąją signalo dalį. Tiesa, daugelyje taikymų (pvz. valiutų kursų, akcijų kainų kitimo signaluose) nestacionariosios signalo komponentės (pvz. trendai) gali būti kur kas informatyvesnės už stacionariuosius statistinius dydžius.

1.2.4 Analoginiai ir skaitmeniniai signalai. Diskretizacijos dažnio pasirinkimas

Kaip taisyklė mus supančio pasaulio procesai tolydžiai (nepertraukiamai) kinta laike. Tai reiškia, kad tokių procesų pagimdyti signalai taip pat yra tolydžios laiko kintamojo t funkcijos. Pavyzdžiui, oro temperatūrą iš principo galime matuoti kaip norima dažnai, tokiu būdu gaudami temperatūros signalo priklausomybę kaip norima trumpame laiko intervale. Kiti pavyzdžiai būtų: sinusinės bangos (2) signalas, iš mikrofono išeinantis signalas, automobilio greitis ir t. t. Signalas, kuriame laikas kinta tolydžiai, vadinamas *analoginiu* signalu arba *tolydaus laiko* signalu.

Pastebėkime, kad analoginio signalo sąvoka tinka tik teoriniam nagrinėjimui. Ekperimentuose, pavyzdžiui stebėdami oro temperatūrą (kurios reikšmės apibrėžiančios tikslios matematinės formulės nežinome) arba registruodami paciento elektrokardiogramą, turime apsibrėžti kaip dažnai matuosime tiriamąjį signalą. Akivaizdu, kad kuo dažniau tai darysime, tuo daugiau informacijos turėsime apie signalą. Tačiau didinant matavimų dažnumą neišvengiamai didėja kompiuterinės atminties, reikalingos duomenims laikyti, kiekis, taip pat auga algoritmų, apdorojančių signalą, vykdymo trukmė. Susimąstome – o kiek detalūs duomenys turi būti mūsų praktinėms reikmėms užtikrinti? Jeigu kalbame apie oro temperatūrą, ji kaip taisyklė beveik nepasikeičia per vienos sekundės ar netgi vienos minutės laiko tarpą. Tą patį galima pasakyti ir apie matuojamų duomenų tikslumą. Tarkime, kad šiuo metu oro temperatūra yra 18.7397856 laipsnio. Ar toks didelis tikslumas mums reikalingas? Dažniausiai visiškai pakanka pamatuoti oro temperatūrą vienos dešimtosios laipsnio tikslumu, taigi gauname 18.7 laipsnio. Daugumoje meteorologinių tarnybų oro temperatūra registruojama kas valandą vienos dešimtosios laipsnio tikslumu – to pilnai pakanka.



9 pav.: Iliustracija kairėje: analoginis signalas. Iliustracija dešinėje: skaitmeninis signalas, gautas diskretizuojant iliustracijoje kairėje pavaizduotą analoginį signalą.

Taigi, eksperimentatoriai neišvengiamai susiduria su signalo reikšmių sekomis, išreikštomis pavidalu (1). Signalas, apibrėžtas savo reikšmių seka iš anksto pasirinktais laiko momentais, vadinamas *skaitmeniniu* signalu, arba *diskretizuotu* signalu.

Analoginio ir jį atitinkančio skaitmeninio signalo pavyzdžiai pavaizduoti 9 pav.

Skaitmeninis signalas gali būti nesunkiai gaunamas iš analoginio, nuskaitant reikšmes su pasirinktu laiko diskretizacijos žingsniu. Tačiau atvirkštinė operacija: iš skaitmeninio signalo vienareikšmiškai atstatyti analoginį, neįmanoma – diskretizuojant signalą informacija prarandama negrįžtamai.

Tiek eksperimentuose, tiek verčiant analoginius signalus į skaitmeninius, dažniausiai naudojami vienodai vienas nuo kito nutolę diskretizacijos laikai t_i . Pavyzdžiui, matuojame temperatūrą tiksliai kas vieną valandą, niekada nedažnindami ir nevėlindami vienas po kito sekančių matavimų. Registruojant greitai kintančius medicinines kilmės signalus (elektrokardiogramas, encefalogramas ir kt.), diskretizacijos laikai kur kas dažnesni – sakykime, kas vieną milisekundę. Tačiau kartais imamas ir netolygus diskretizacijos taškų tinklas, t. y. laiko momentai t_i (kada matuojame signalą) nevienodai išsidėstę vienas kito atžvilgiu. Tokia situacija dažna stebint kokybinį signalo elgsenos pasikeitimą tam tikruose laiko intervaluose, pavyzdžiui laikai t_i sutankinami pastebėjus, kad signalas pradėjo osciliuoti (virpėti su staiga išaugusiu dažniu).

Diskretizacijos dažnio pasirinkimas

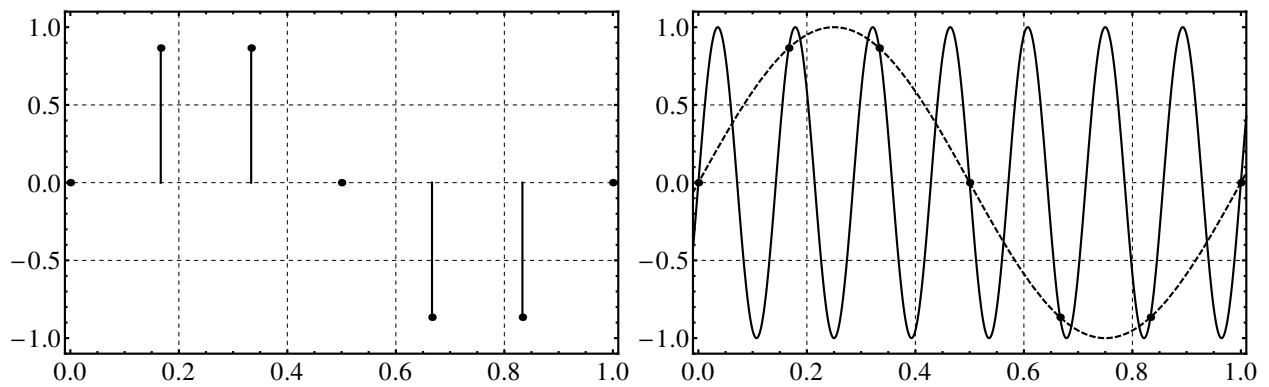
Tarkime, turime analoginį signalą ir norime jį paversti skaitmeniniu, naudodami tolygų diskretizacijos žingsnį

$$\Delta t \equiv t_{i+1} - t_i, \quad i = 0, 1, \dots$$

Detaliau panagrinėsime svarbų klausimą: *kaip pasirinkti laiko diskretizavimo žingsnį Δt ?* Iš vienos pusės, taupydami resursus, norime, kad Δt būtų kaip galima didesnis. Tačiau iš kitos pusės – Δt privalo būti pakankamai mažas, tam, kad diskretizuotos signalo reikšmės išsamiai atspindėtų analoginio signalo formą. Atsakymas į suformuluotą klausimą padės suprasti kokie galimi pavojai susiję su šiuo pasirinkimu. Kartais vietoje sąvokos “diskretizavimo žingsnis” vartojamas susijęs terminas: “diskretizacijos dažnis”. Diskretizacijos dažnis f_t yra atvirkščiai proporcingas diskretizavimo žingsniui Δt :

$$f_t = \frac{1}{\Delta t}.$$

Sakykime, gavome skaitmeninį signalą, apibrėžtą 10 pav. iliustracijoje kairėje pavaizduotomis



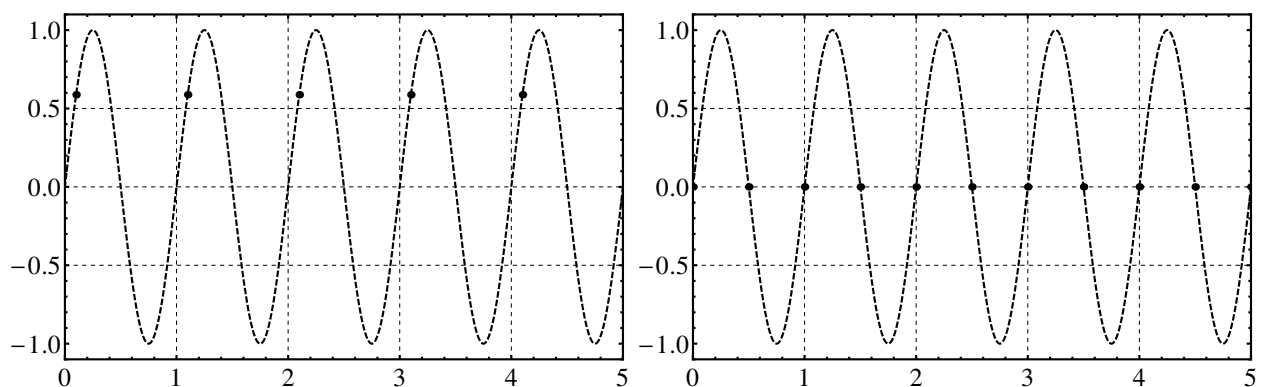
10 pav.: Nevienareikšmiškumas tarp skaitmeninio ir analoginio signalų. Iliustracija kairėje: skaitmeninis signalas. Iliustracija dešinėje: du sinusinės bangos tipo analoginiai signalai, kiekvienas iš kurių atitinka skaitmeninio signalo, pavaizduoto iliustracijoje kairėje, reikšmes.

reikšmėmis

$$t_i = i \Delta t, \quad f_i = \sin(14\pi t_i), \quad \Delta t = \frac{1}{6}, \quad i = 0, 1, 2, \dots, 6.$$

Kažkam gali pasirodyti, kad šios reikšmės atitinka periodinį sinusinės bangos tipo signalą, kurio grafikas nubraižytas 10 pav. iliustracijoje dešinėje punktyrine kreive. Tačiau, kitas asmuo gali teigti, kad analoginio signalo atitikmuo pavaizduotas 10 pav. iliustracijoje dešinėje ištisine kreive, kadangi duotiems t_i galioja $f_i = \sin(14\pi t_i) = \sin(2\pi t_i)$. Jie abu bus teisūs, nes net jeigu žinome, kad duotosios skaitmeninio signalo reikšmės atitinka sinusinės bangos diskretizaciją, iš šių reikšmių neįmanoma vienareikšmiškai nustatyti tikrąjį signalo dažnį.

11 pav. parodyti dar du problematiški diskretizacijos žingsnio Δt pasirinkimai. Iliustracijoje kairėje turime atvejį kai diskretizacijos žingsnis sutampa su analoginio signalo periodu. Akivaizdu, kad gaunamos skaitmeninio signalo reikšmės neatkartoja analoginio signalo formos. Vadinasi, pasirinktas per didelis diskretizacijos žingsnis.



11 pav.: Iliustracija kairėje: parinkę diskretizacijos žingsnį Δt lygų analoginio signalo periodui T , gauname, kad signalo reikšmės nesikeičia bėgant laikui. Iliustracija dešinėje: jeigu $\Delta t = T/2$, galime gauti vien tik nulines reikšmes.

Sumažinkime Δt iki analoginio signalo periodo pusės: $\Delta t = T/2$. Vis dėlto, kaip rodo 11 pav. iliustracija dešinėje, yra galimybė, kad diskretizuodami nuskaitysime tik nulines signalo reikšmes, taigi surinkta informacija toli gražu nebus pilna. Darome išvadą, kad diskretizacijos žingsnis turi būti dar labiau sumažintas.

Išnagrinėtas pavyzdys rodo, kad norint diskrečiomis reikšmėmis aprašyti sinusinės bangos (2) analoginio signalo formą, diskretizacijos žingsnis Δt turi tenkinti nelygybę $\Delta t < T/2$, kur T yra analoginio signalo periodas. Šią išvadą performuluokime panaudodami diskretizacijos dažnio f_t sąvoką. Kadangi sinusinės bangos (2) dažnis f ir periodas T yra susiję sąryšiu $T = 1/f$, gauname, kad diskretizacijos dažnis f_t privalo tenkinti nelygybę

$$f_t > 2f.$$

Gautasis ribinis diskretizacijos dažnis $2f$ vadinamas *Naikvisto dažniu* (angliškai “Nyquist frequency”).

6 skyriuje parodysime, kad bet kurį signalą galima išreikšti įvairaus dažnio sinusinių ir kosinusinių bangų suma. Bendru atveju atsakymą į suformuluotą klausimą apie tinkamo diskretizacijos dažnio pasirinkimą duoda Naikvisto–Šenono teorema (literatūros šaltiniuose anglų kalba dar vadinama “Whittaker–Nyquist–Kotelnikov–Shannon theorem”):

Tarkime, kad signalas $f(t)$ neturi aukštesnių dažnių nei B hercų. Tuomet signalą galima pilnai apibrėžti diskretizuojant (matuojant) jo reikšmes didesniu nei $2B$ diskretizacijos dažniu.

Kitaip sakant, jeigu, pavyzdžiui, turime tam tikruose laiko tarpuose osciliuojantį signalą ir iš anksto žinome, kad pati intensyviausia osciliacija neviršija B hercų, tai eksperimentiškai matuojant signalo reikšmes pakanka tai daryti nors šiek tiek dažniau nei kas $1/2B$ sekundę.

Beje, jeigu registruodami signalo vertes naudojame netinkamą (per mažą) diskretizacijos dažnį (mažesnę už Naikvisto dažnį), galime susidurti su taip vadinamu *stroboskopiniu efektu*, kuris iliustruotas 10 pav. Tarkime, kad tikrasis analoginis signalas pavaizduotas 10 pav. iliustracijoje dešinėje ištisine kreive. Tačiau pasirodo ir kitas žemo dažnio signalas–vaiduoklis, kurio grafikas nubraižytas punktyrine linija, taip pat einantis per duotąsias netinkamai diskretizuotas reikšmes. Šis efektas analogiškas stroboskopiškumui, kurį galima pastebėti įjungus ventiliatorių, esantį liuminescencinio apšvietimo fone: ventiliatoriui pradėjus sukis išvystame artefaktą, lėtai besisukantį arba ventiliatoriaus sukimosi kryptimi, arba priešinga. Stroboskopiškumas atsiranda esant tam tikram ventiliatoriaus sukimosi dažnio ir apšvietimo mirgėjimo dažnio santykiui.

Eksperimentuose visada registruojame skaitmeninius signalus. Taip pat ir realizuojami signalus apdorojančius algoritmus dirbame vien tik su skaitmeniniais signalais (o ne su analoginiais). Todėl toliau nagrinėsime tik skaitmeninius signalus, išreikštus pavidalu (1).

1.3 Signalų analizės tikslai

Nagrinėsime algoritmus, susijusius su skaitmeninių signalų apdorojimu. Šios žinios būtinos patikint sprendimų priėmimą kompiuteriui. Pavardinkime keletą tokių sprendimų pavyzdžių. Kompiuterizuotos metalo pjovimo staklės, kuriose pjūklas keičiamas nauju automatiškai, remiantis jo skleidžiamo garso signalu. Kompiuterinė diagnostika medicinoje, pavyzdžiui, miokardo infarkto rizikos laipsnio įvertinimas, remiantis elektrokardiograma. Turimas signalas užgožtas išorinių triukšmų (tarkime, šalia pjovimo staklių groja muzika) – būtina išskirti svarbiąją signalo dalį.

Signalų apdorojimą galima palyginti su užteršto vandens filtravimu. Pagrindinis tikslas – objektą (šiuo atveju signalą) sudarančių komponentų suskaidymas į pageidautinas (informatyvias, naudingas) ir nepageidautinas. Žinoma, jeigu iš anksto nemažai žinome kokiomis savybėmis pasižymi signalą skleidžiantis objektas, tai palengvina signalo apdorojimą. Tačiau dažnai tenka spręsti atvirkštinį uždavinį: iš turimo signalo nustatyti šaltinio savybes (pvz. kompiuterinė diagnostika medicinoje). Beje, tiriant signalus neretai pavyksta aptikti ir tokias juos skleidžiančių šaltinių savybes, apie kurias

nebuvo anksčiau žinoma. Taigi, signalų apdorojimas gali praversti net ir signalų sintezėje (pvz. kuriant kompiuterinį diktorių – žmogaus balso sintezatorių).

Algoritmai, skirti signalų apdorojimui, gali būti suskirstyti į veikiančius laiko t srityje ir į veikiančius dažnio srityje. Pirmajai grupei priklausančys algoritmai operuoja tiesiogiai su signalo reikšmėmis. Tuo tarpu antrosios grupės metodai pirmiausiai apskaičiuoja signalo spektrą, parodantį kokių dažnių svyravimai sudaro signalą bei koks yra atskirų dažnių intensyvumas. Šie dažnius nusakantys koeficientai (signalų spektras) gali būti apdorojami algoritmo, veikiančio dažnio srityje. Po spektro korekcijos belieka atstatyti signalo priklausomybę nuo laiko t . Apie metodus, skirtus dažnių signalo komponentų išskyrimui ir apdorojimui kalbėsime 6 ir 7 skyriuose.

Dirbant su signalu visada naudinga nubraižyti jo grafiką, iš kurio dažnai galime identifikuoti kai kurias paprasčiausias savybes. Pavyzdžiui, prekybinės kilmės ir meteorologinės kilmės signalams būdingas sezoniškumas (tam tikrų prekių apyvarta išauga vasarą ir sumažėja žiemą; analogiškai su oro temperatūra). Taip pat betarpiškai iš grafiko galima nustatyti tokias signalo tendencijas kaip *trendai* (tam tikrą laiko tarpą signalas nors ir svyruodamas, bet turi tendenciją tiesiškai augti arba mažėti). Trendai būdingi valiutų kursų, naftos, aukso ir akcijų kainos, orų temperatūros ir daugeliui kitų signalų. Pavyzdžiui, 1 pav. iliustracijoje kairėje galime išskirti keletą trendų: vidutinė oro temperatūra Šiaurės pusrutulyje akivaizdžiai buvo linkusi didėti laikotarpyje tarp 1920 ir 1940 metų, taip pat laikotarpyje nuo 1970 iki 2000 metų.

Toliau gilinsimės į ne tokius trivialius algoritmus kaip vizualinė signalo analizė. Tačiau tai nereikia, kad betarpiškas grafiko tyrimas visai nesvarbus. Apart paprasčiausių signalo savybių nustatymo vizualinė patikra gali padėti aptikti brokuotus signalo fragmentus (laiko tarpus kada registravimo aparatūra dirbo nekorektiškai – pavyzdžiui, generavo netolydų signalą arba už prasmingų verčių intervalo esančius skaičius). Verta atkreipti dėmesį, kad nepaisant didelio tam tikslui skirtų kompiuterinių programų pasirinkimo, signalo grafiko braižymas gali būti ne tokia paprasta užduotis kaip atrodo iš pirmo žvilgsnio. Duomenų kiekis dažnai labai didelis ir vizualizuojant tenka juos skaidyti fragmentais. Taip pat būtina teisingai pasirinkti koordinačių ašių mastelį (kai kuriais atvejais teisingas pasirinkimas vienai arba abiem ašims yra logaritminis mastelis, kitais atvejais – tiesinis mastelis).

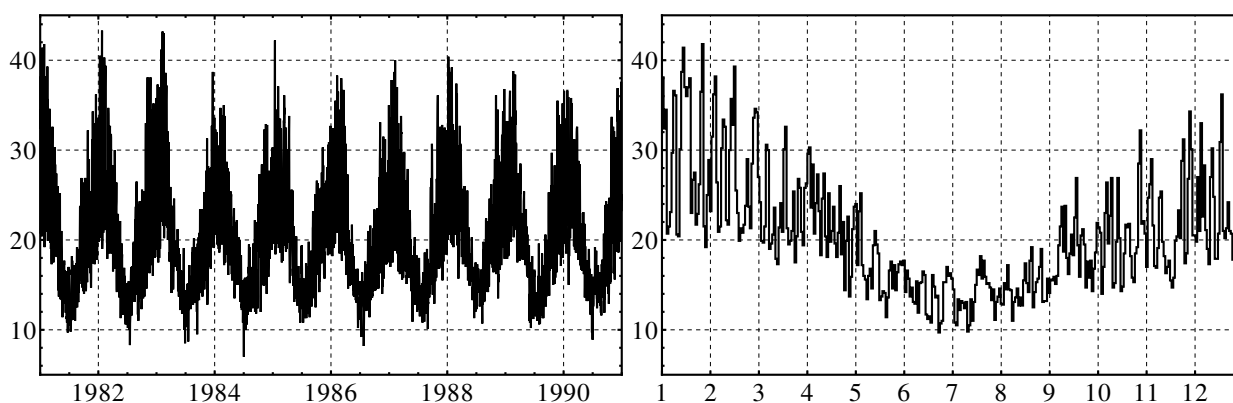
2 Signalų glodinimo algoritmai

2.1 Signalų glodinimas

Šiame skyriuje panagrinėsime paprasčiausius signalų apdorojimo algoritmus, kurie dažnai naudojami pirmojoje eksperimentinių duomenų kompiuterinės analizės stadijoje. Pagrindinis šio etapo tikslas – paruošti signalą tolesniam apdorojimui kitais metodais. Tiriamus algoritmus iliustruosime realių duomenų pavyzdžiais.

Jeigu norime gauti bendrą signalo svyravimų vaizdą, dažnai pirmiausiai pageidautina jį padaryti “glodžiu”, t. y. pašalinti nežymius registravimo triukšmus bei atmesti nedidelius signalo svyravimus (osciliacijas). Toks signalo apdorojimas vadinamas *signalu glodiniu*.

Signalu glodinimas gali būti prasmingas ir norint gauti trumpalaikę prognozę. Pavyzdžiui, prekiautojams valiutomis Forex (angliškai “Forex – foreign exchange market”) rinkoje labai svarbu atspėti koks bus konkrečios valiutų poros kursas rytoj. Iš pirmo žvilgsnio, galėtume pasikliauti šios dienos kursu, manydami, kad jis turėtų išsilaikyti. Tačiau tai nėra pats geriausias spėjimas – įmanoma, kad šiandieninis kursas yra signalo svyravimo apie glodžiąją signalo dalį maksimume arba minimume. Tuomet ateityje jis greičiausiai bus linkęs sugrįžti prie pusiausvyros. Taigi, dažniau geresnę prognozę galima padaryti remiantis trendu, kuris išryškėja suglodinus signalą.

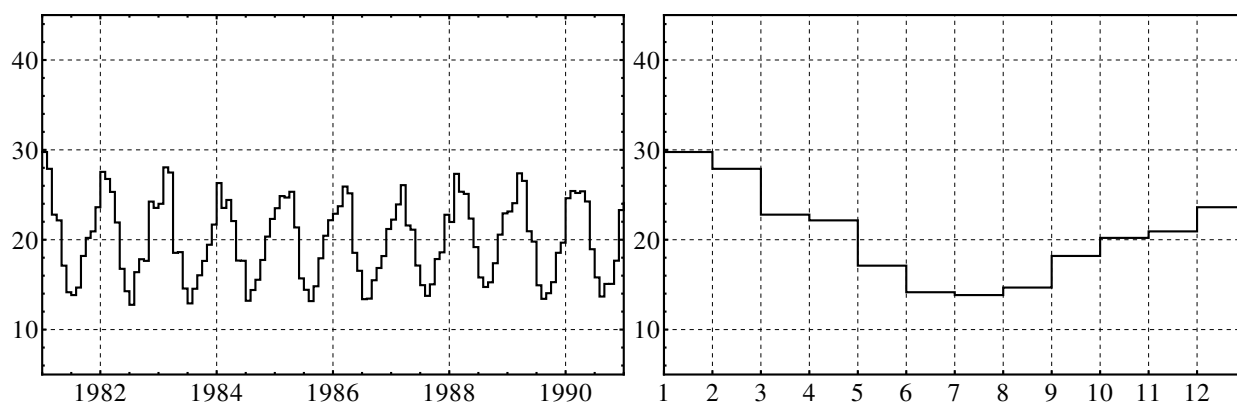


12 pav.: Dienos maksimalios oro temperatūros Melburne (Australija) svyravimas, pagal [4]. Iliustracija kairėje: kasdieniniai duomenys nuo 1981 iki 1991 metų. Iliustracija dešinėje: iliustracijoje kairėje fragmentas – kasdieniniai duomenys 1981 metais (abscisių ašyje atidėti mėnesiai).

Pradėkime nuo konkretaus signalo pavyzdžio – kiekvieną dieną registruotos maksimalios oro temperatūros Melburne (Australija) svyravimų laikotarpyje nuo 1981 iki 1990 metų (žr. 12 pav.). Pastebėkime, kad turimas signalas nėra glodus – jame labai daug osciliacijų, nors tai ir turi būti būdinga oro temperatūros svyravimams. Vienas iš galimų sprendimų siekiant eliminuoti lokalius laike svyravimus signale yra pereiti nuo turimų kasdieninių duomenų prie vidutinės mėnesio temperatūros reikšmių sekos signalo (žr. 13 pav.). Iš tiesų, osciliacijų nebelieka – vidutinė mėnesio temperatūra kinta kur kas tolygiau, nei kiekvienos dienos jos vertė.

Tačiau toks vidurkinimo metodas turi trūkumą – smarkiai sumažėja duomenų kiekis. Pereinant prie vidutinių mėnesio verčių, vietoje 365 signalo taškų per metus lieka tik 12 taškų. Dėl šios priežasties 13 pav. grafikai įgijo kampuotą laužtės formą. Tai ypač pastebima stebint iliustracijoje dešinėje pavaizduotą vieno metų trukmės signalo fragmentą.

Norėdami suprasti kaip galėtume suglodinti signalą išvengdami duomenų kiekio sumažėjimo, aptarsime 13 pav. pavaizduotos vidutinės mėnesio temperatūros skaičiavimo mechanizmą. Paprastumo vardan tarkime, kad visi mėnesiai turi po 30 dienų ir duoto signalo reikšmių sekos ilgis dalinasi iš 30.



13 pav.: Iš 12 pav. pavaizduotų duomenų gautas vidutinės mėnesio temperatūros reikšmės svyravimo signalas.

Taigi, pradinio signalo reikšmių seka f_0, f_1, \dots, f_N pirmiausiai suskaidoma į nesikertančius tarpusavyje posekius: pirmojo mėnesio duomenis f_0, f_1, \dots, f_{29} , antrojo mėnesio duomenis $f_{30}, f_{31}, \dots, f_{59}$, ir taip toliau. Paskutinįjį posekį sudaro paskutiniojo mėnesio duomenys $f_{N-29}, f_{N-28}, \dots, f_N$. Visi posekiai turi po 30 elementų. Tuomet apskaičiuojame pirmojo posekio reikšmių aritmetinį vidurkį, antrojo posekio reikšmių aritmetinį vidurkį ir visų kitų posekių reikšmių aritmetinius vidurkius – iki pat paskutiniojo posekio imtinai. Gautos vidurkių reikšmės ir apibrėžia vidutinės mėnesio temperatūros signalą. Be abejo šiame suvidurkintame signale yra tiek reikšmių, kiek sudarėme posekių – taigi duomenų kiekis vidutinės mėnesio temperatūros signale lygus mėnesių skaičiui per stebėtą laikotarpį. Šis duomenų kiekis yra 30 kartų mažesnis už pradinio 12 pav. pavaizduoto signalo duomenų kiekį.

2.1.1 Slenkančio vidurkio algoritmas

Dabar jau galime pastebėti galimą mūsų taikyto vidurkinimo algoritmo apibendrinimą. Prisiminkime, kad pradinį signalą skaidėme į nesikertančius tarpusavyje posekius. Tačiau juk niekas nedraudžia imti besikertančius posekius, kad ir taip: pirmasis posekis f_0, f_1, \dots, f_{29} , antrasis posekis f_1, f_2, \dots, f_{30} , trečiasis posekis f_2, f_3, \dots, f_{31} ir taip toliau. Priešpaskutinis posekis būtų $f_{N-30}, f_{N-29}, \dots, f_{N-1}$, o paskutinis posekis $f_{N-29}, f_{N-28}, \dots, f_N$. Naujuoju būdu apibrėžti posekiai taip pat visi turi po 30 elementų. Jeigu nesikertančių tarpusavyje posekių skaidinys atitiko mėnesius, kurie prasideda 1-ąją ir baigiasi 30-ąją kalendorinio mėnesio dieną, tai naujasis – besikertančių tarpusavyje posekių skaidinys atitinka mėnesius, kurie apima 14 dienų, ėjusių prieš einamąją dieną, 1 einamąją dieną ir 15 dienų, sekančių po einamosios, taigi iš viso 30 dienų. Einamajai dienai slenkant per kalendorių, ir gauname šį besikertančių tarpusavyje posekių skaidinį. Kaip ir anksčiau, apskaičiavę kiekvieno iš posekio reikšmių aritmetinį vidurkį, gauname suvidurkintą signalą. Šia idėja paremtas signalo apdorojimas vadinamas *slenkančio vidurkio* (angliškai “moving average”) glodinimu. Matyti, kad slenkančio vidurkio metodu suglodintame signale yra beveik tiek pat reikšmių, kiek ir pradiniam signale – taigi galime sakyti, kad išsprendėme 13 pav. stebėtą duomenų kiekio sumažėjimo problemą. “Beveik” – nes skaidant į posekius einamoji diena negali prasidėti indeksu 0 (duomenyse neturėsime 14 prieš tai buvusių dienų), taip pat ji negali baigtis indeksu N (neegzistuos 15 dienų, einančių po einamosios). Taigi, skaidant duomenis į mėnesius, einamosios dienos indeksas slenka nuo 14 iki $N - 15$ imtinai.

Taip pat visiškai nebūtina prisirišti prie skaičiaus 30. Eksperimentiniuose signaluose laikas gali būti skaičiuojamas ne dienomis ir mėnesiais, bet visai kitame mastelyje, pavyzdžiui, milisekundėmis arba metais. Tačiau slenkančio vidurkio metodo idėja – suvidurkinti signalą pagal tam tikrą skaičių

kaimyninių reikšmių, visada gali būti pritaikoma, nepriklausomai nuo laiko kintamojo mastelio.

Matematiškai nuodugnai panagrinėsime slenkančio vidurkio algoritmą ir galimus jo apibendrinimus. Tarkime, skaitmeninis signalas apibrėžtas $N+1$ reikšmėmis

$$f_0, f_1, \dots, f_N.$$

Pirmiausiai pasirenkamas algoritmo parametras – teigiamas sveikas skaičius K . Taikydami slenkantį vidurkinimą aplink einamąją signalo reikšmę f_i , suvidurkinsime K prieš tai einančių reikšmių, K po to einančių reikšmių ir pačią centrinę reikšmę f_i . Naująsias suvidurkinto signalo reikšmes žymėsime g_i , kurios (matematinė kalba užrašius jų apibrėžimą) apskaičiuojamos pagal formulę

$$g_i = \frac{1}{2K+1} (f_{i-K} + f_{i-K+1} + \dots + f_i + \dots + f_{i+K}), \quad i = K, K+1, \dots, N-K,$$

arba lakoniškai

$$g_i = \frac{1}{2K+1} \sum_{j=-K}^K f_{i+j}, \quad i = K, K+1, \dots, N-K. \quad (5)$$

Pažymėkime, kad einamosios signalo reikšmės, atitinkančios indekso reikšmes $i = 0, 1, \dots, K-1$ ir $i = N-K+1, N-K+2, \dots, N$ praleidžiamos, nes vidurkindami kreiptumėmis į neegzistuojančias signalo reikšmes. Pavyzdžiui, paėmę indeksą $i = 0$, turėtume

$$g_0 = \frac{1}{2K+1} (f_{-K} + f_{-K+1} + \dots + f_0 + \dots + f_K),$$

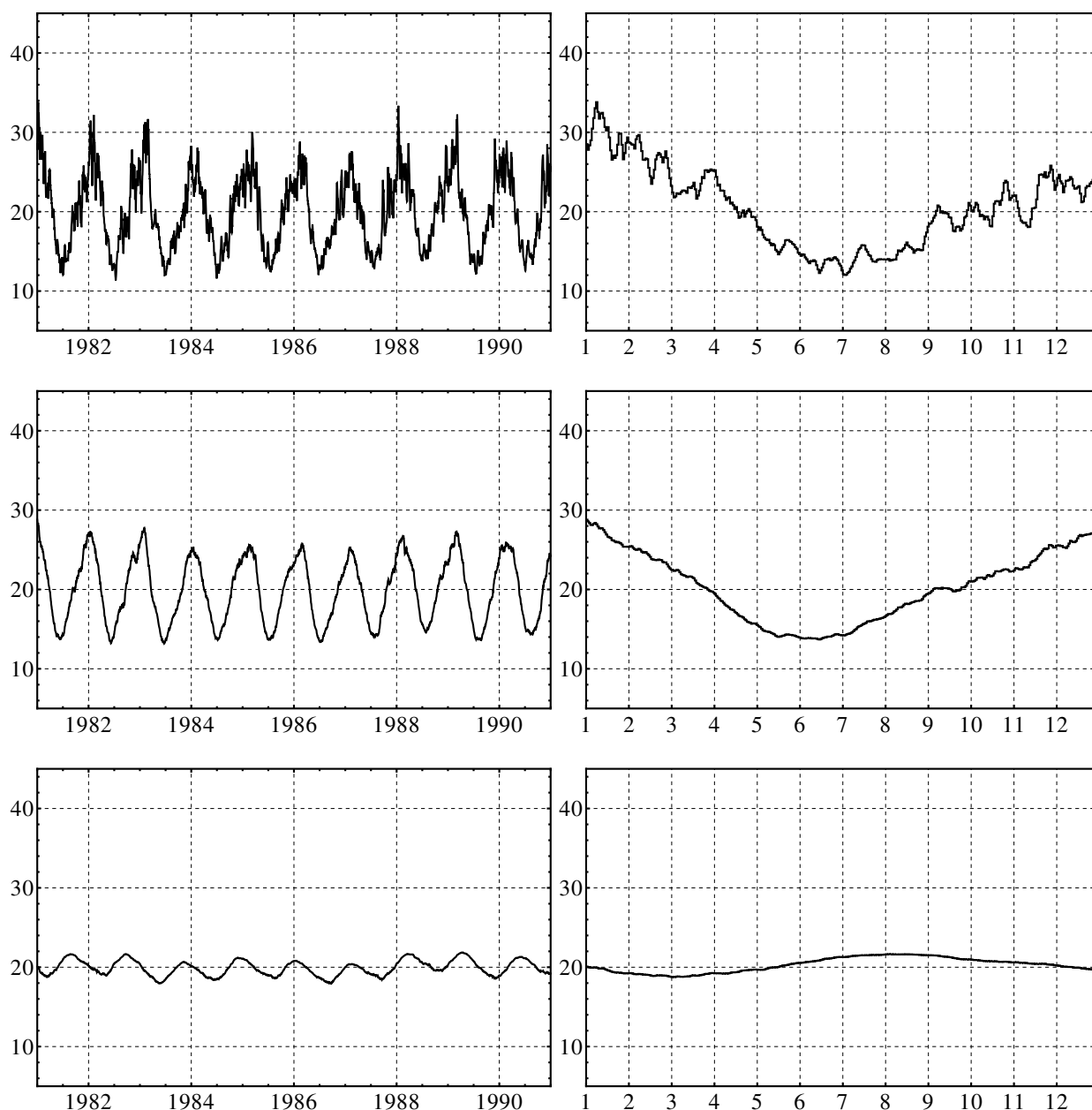
tačiau reikšmė f_{-K} neapibrėžta! Trūkstamas reikšmės g_0, g_1, \dots, g_{K-1} ir $g_{N-K+1}, g_{N-K+2}, \dots, g_N$ galime arba praleisti (turėsime suvidurkintą signalą šiek tiek trumpesniame intervale), arba kiekvieną iš jų apskaičiuoti su mažesniu K , arba tiesiog nukopijuoti atitinkamas nevidurkinto signalo reikšmes f_i .

Pastaba. Matome, kad skaičiuodami einamąjį vidurkį iš viso įskaitome $2K+1$ signalo reikšmių, taigi nelyginį jų skaičių. Pastebėkime, kad formaliai nutolstame nuo anksčiau nagrinėto pavyzdžio kai vidurkinome 30, t. y. lyginį skaičių reikšmių, tačiau taikymuose dažniausiai stengiamasi išlaikyti simetriją einamosios reikšmės atžvilgiu – todėl naudojamas slenkantis vidurkinimas su nelyginiu verčių kiekiu. Iš principo nesunku apibendrinti šį metodą, kad jis veiktų su bet koku (tiek lyginiu, tiek nelyginiu) vidurkinamų reikšmių skaičium, tačiau tai niekaip neįtakoja gaunamų rezultatų kokybės.

Parametro K pasirinkimas

Kaip pasirinkti parametą K ? Vienareikšmiško atsakymo nėra, tai priklauso ir nuo duomenų ir nuo mūsų tikslų – kaip smarkiai norime suglodonti signalą. Per daug suglodontas signalas gali netekti svarbių savo detalių, netgi įgyti iškreiptą formą.

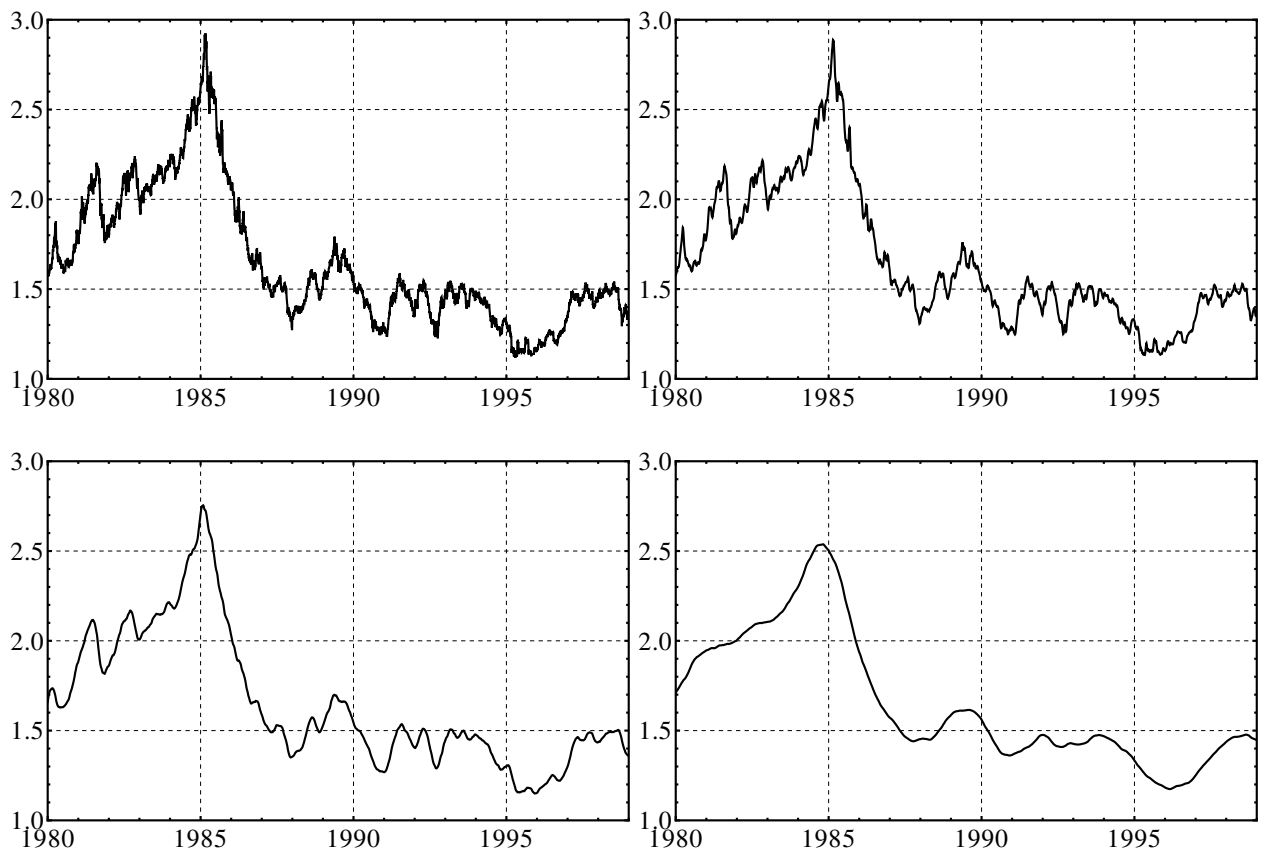
Iliustruokime slenkančio vidurkio algoritmo veikimą to pačio 12 pav. pavaizduoto signalo atžvilgiu. 14 pav. pateikti skaitinio eksperimento rezultatai su keletu parametro K reikšmių: $K = 5$, $K = 30$ ir $K = 150$. Iš šių grafikų gerai matyti, kad parinkus santykinai nedidelį K , signalas nors ir tampa šiek tiek glodesniu (lyginant su neglodontu originalu), tačiau vis dar turi nemažai neesminių osciliacijų; iš kitos pusės, jeigu K pernelyg didelis – signalas pasidaro panašus į tiesę, taigi prarandame ne tik neesmines, bet ir esmines detales. Taip pat, jeigu K neprotingai didelis, rizikuojame, kad signalo forma išsikreips, kaip kad nutiko 14 pav. iliustracijoje apačioje dešinėje pavaizduotam vienu metų trukmės signalo fragmentui (vidurkindami su dideliu $K = 150$ “užgriebėme” reikšmes iš sekančių metų).



14 pav.: Iš 12 pav. pavaizduotų duomenų slenkančio vidurkio algoritmu su įvairiomis parametro K reikšmėmis suglodonti temperatūros reikšmės svyravimo signalai. Kaip ir 12 pav., iliustracijose kairėje pavaizduotas visas signalas nuo 1981 iki 1990 metų, iliustracijose dešinėje šio signalo fragmentas 1981 metais (abscisių ašyje atidėti mėnesiai). Iliustracijos viršuje: atvejis $K = 5$. Iliustracijos viduryje: atvejis $K = 30$. Iliustracijos apačioje: atvejis $K = 150$.

Tuo pačiu slenkančio vidurkio metodu suglodonti signalai kitiems duomenims (valiutų kurso svyravimo signalui) matomi 15 pav. Jeigu 14 pav. atveju algoritmo parametro reikšmė $K = 150$ buvo akivaizdžiai per didelė, tai šiam signalui ji gali būti visai priimtina – jeigu mūsų tikslas, pavyzdžiui, yra trendų išryškinimas.

Kaip matyti iš nagrinėtų signalų pavyzdžių, slenkančio vidurkio algoritmas faktiškai eliminuoja iš signalo aukštų dažnių (greitai svyruojančias) komponentes. Dažnį, kurį viršijančios komponentės yra šalinamos, apsprendžia parametro K dydis. Kuo didesnis K , tuo žemesnių dažnių komponentės eliminuojamos.



15 pav.: JAV dolerių keitimo kurso į Šveicarijos frankus svyravimas. Kasdieniniai duomenys nuo 1979 m. gruodžio 31 d. iki 1998 m. gruodžio 31 d., pagal [4]. Iliustracija viršuje kairėje: originalus ne-vidurkintas signalas. Iliustracija viršuje dešinėje: slenkančio vidurkio algoritmu su $K = 5$ suglodintas signalas. Iliustracija apačioje kairėje: slenkančio vidurkio algoritmu su $K = 30$ suglodintas signalas. Iliustracija apačioje dešinėje: slenkančio vidurkio algoritmu su $K = 150$ suglodintas signalas.

2.1.2 Slenkančio vidurkio su svoriais algoritmas

2.1.1 skyrelyje išdėstyta slenkančio vidurkio algoritme einamajai apdorojamo signalo reikšmei f_i skaičiuojame *aritmetinę* jos kaimyninių reikšmių vidurkį, pagal formulę (5). Kitaip sakant, visos vidurkinamos reikšmės išraiškoje (5) turi vienodą svorį (reikšmingumą): daugiklį $1/(2K + 1)$ priešais kiekvieną narį sumoje.

Tačiau daugeliu atvejų yra logiška manyti, kad labiau nutolusios (nuo einamosios) reikšmės turėtų įgyti mažesnę svorį vidurkinimo procese, o artimiausių kaimyninių reikšmių svoris turėtų būti ženkliai didesnis. Tokie samprotavimai veda prie standartinio slenkančio vidurkio metodą apibrėžiančios formulės (5) apibendrinimo:

$$g_i = \sum_{j=-K}^K w_j f_{i+j}, \quad i = K, K+1, \dots, N-K, \quad (6)$$

čia neneigiami koeficientai $w_{-K}, w_{-K+1}, \dots, w_0, \dots, w_K$, pasižymintys savybe

$$\sum_{j=-K}^K w_j = 1, \quad (7)$$

vadinami vidurkinimo svoriais.

Apibendrintas signalo vidurkinimo metodas (6), (7) vadinamas *slenkančio vidurkio su svoriais* (angliškai “weighted moving average”) algoritmu.

Atskiru atveju, parinkę tarpusavyje lygius svorius

$$w_j \equiv \frac{1}{2K+1}, \quad j = -K, -K+1, \dots, 0, \dots, K,$$

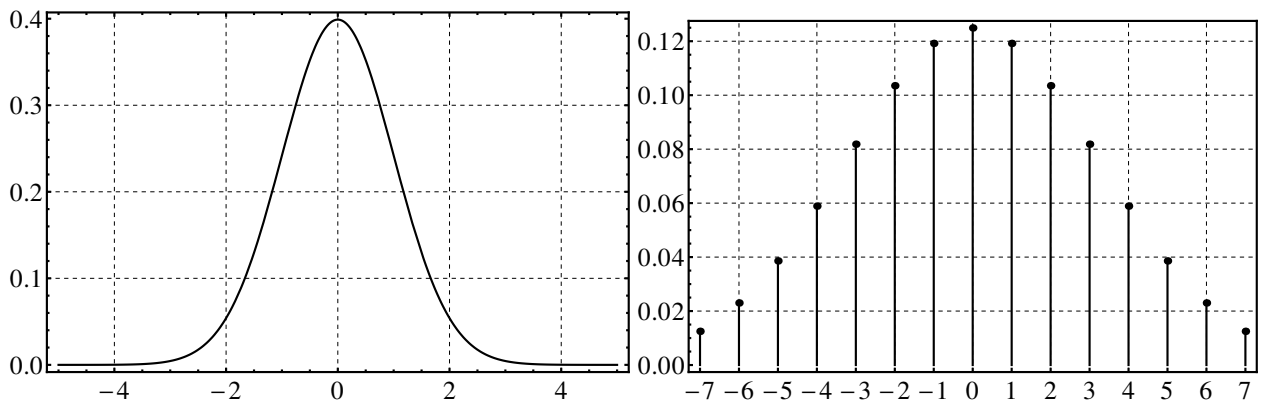
gauname standartinį slenkančio vidurkio algoritmą (5).

Vidurkinimo svorių parinkimas

Vidurkinimo svorius galima parinkti įvairiai, priklausomai nuo to, su koku reikšmingumu norime įskaityti kiekvieną iš dalyvaujančių vidurkinime reikšmių. Svarbu, kad svoriai tenkintų (7), kitu atveju iškreiptume vidurkinamo signalo reikšmių absoliutinį dydį. Dažnai vidurkinimo svoriai apibrėžiami remiantis standartinio normaliojo pasiskirstymo (labai dažnai sutinkamo tikimybių teorijoje ir matematinėje statistikoje) su vidurkiu 0 ir dispersija 1 tankio funkcija

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad (8)$$

kuri taip pat vadinama Gauso funkcija, žr. 16 pav., iliustraciją kairėje. Iš grafiko matyti, kad ši funkcija simetrinė taško $x = 0$ atžvilgiu.



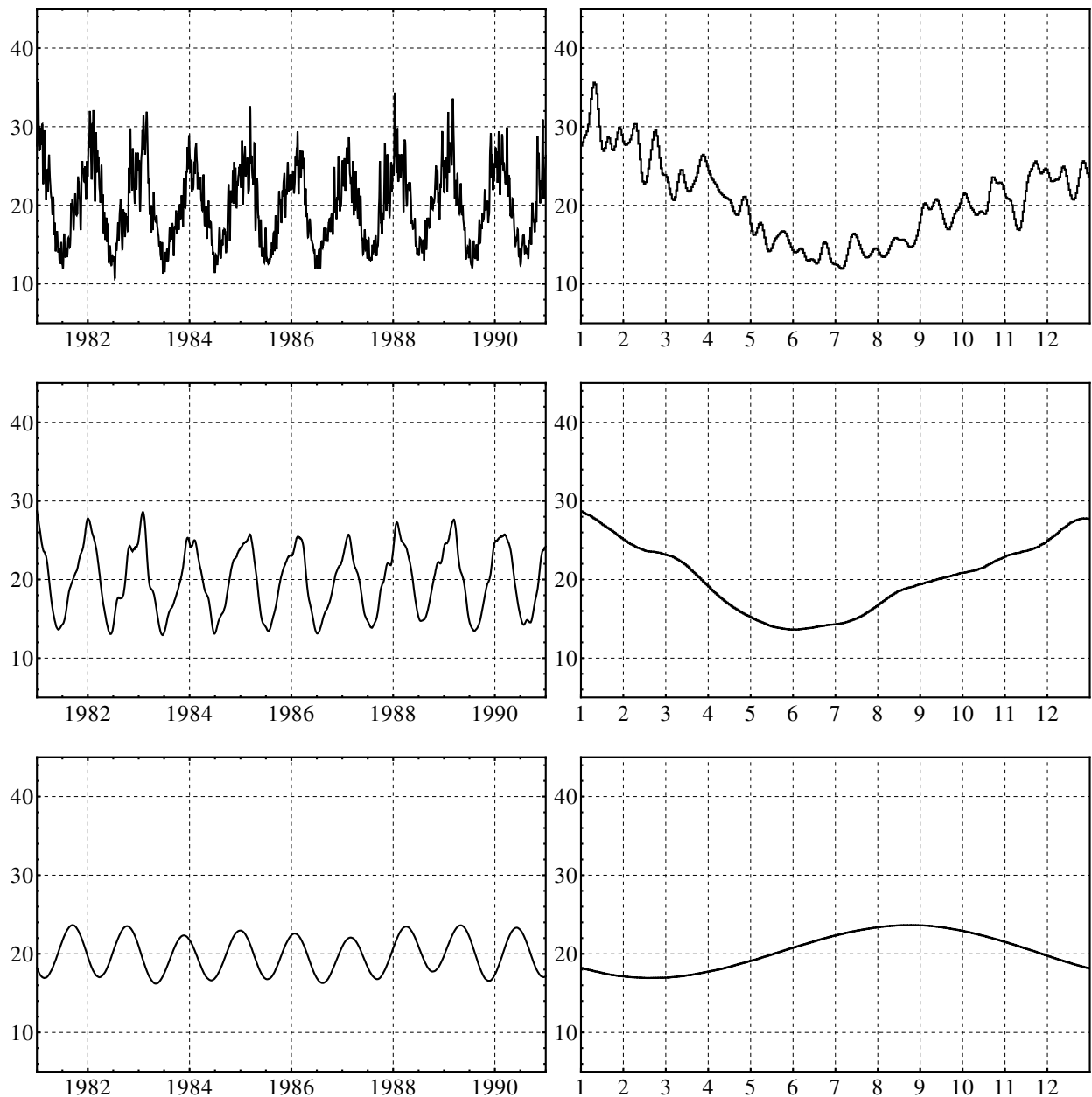
16 pav.: Iliustracija kairėje: Gauso funkcijos $p(x)$ grafikas. Iliustracija dešinėje: iš Gauso funkcijos (8) gauti vidurkinimo svoriai w_j , pagal formulę (9), su algoritmo parametrais $K = 7$, $L = 10$ (abscisių ašyje atidėti svorių w_j indeksai j).

Tarkime, norime, kad vidurkinimo formulėje (6) tolimiausių kaimyninių signalo verčių f_{i-K} ir f_{i+K} svoriai (reikšmingumai) būtų L kartų mažesni už centrinės vertės f_i svorį. Čia $L > 1$ – vartotojo pasirinktas slenkančio vidurkio su svoriais algoritmo parametras. Tuomet apskaičiuojame reikšmę x_K , ties kuria Gauso funkcija $p(x)$ sumažėja L kartų, palyginus su centrine reikšme $x_0 = 0$:

$$\frac{p(0)}{p(x_K)} = L \quad \Leftrightarrow \quad x_K = \sqrt{2 \ln L}.$$

Gauso funkciją $p(x)$ diskretizuojuame intervale $-x_K \leq x \leq x_K$:

$$p_j = p(x_j), \quad x_j = j \frac{x_K}{K}, \quad x_K = \sqrt{2 \ln L}, \quad j = -K, -K+1, \dots, 0, \dots, K,$$



17 pav.: Analogiški rezultatai kaip 14 pav., tik gauti taikant slenkančio vidurkio algoritmą su svoriais. Vidurkinimo svoriai apibrėžti iš Gauso funkcijos (8), pagal formulę (9), su parametru $L = 10$. Parametro K reikšmės analogiškos kaip 14 pav.

Gautų dydžių p_j suma nebūtinai lygi vienetui. Tam, kad užtikrinti savybės (7) galiojimą, vidurkinimo svorius, paremtus Gauso funkcija $p(x)$, apibrėžiame kaip normuotus dydžius p_j :

$$w_j = p_j \left/ \sum_{j=-K}^K p_j \right., \quad j = -K, -K+1, \dots, 0, \dots, K. \quad (9)$$

Tam pačiam signalui (pavaizduotam 12 pav.), palyginome paprastu (be svorių) slenkančio vidurkio algoritmu gautus rezultatus (žr. 14 pav.) ir slenkančio vidurkio su svoriais algoritmu gautus rezultatus (žr. 17 pav.). Abiem atvejais buvo eksperimentuojama su tomis pačiomis parametro K vertėmis: $K = 5$, $K = 30$ ir $K = 150$. 17 pav. vidurkinimo svorius apskaičiavome remdamiesi Gauso funkcija (8), pagal formulę (9), su parametru $L = 10$. Matome, kad esant dideliems K , signalas suglodinamas šiek tiek kokybiškiau taikant vidurkinimą su svoriais.

Praktiškai, glodinant signalus slenkančio vidurkio algoritmu reikia eksperimentuoti su įvairiomis parametro K reikšmėmis, o jeigu taikome vidurkinimą su svoriais, apibrėžtais per Gauso funkciją, tai ir su parametro L reikšmėmis.

Modifikuotas slenkančio vidurkio su svoriais algoritmas

Kartais reikia apskaičiuoti einamąją sugludinto signalo reikšmę g_i , remiantis tik originalaus signalo reikšmėmis iš “praeties”: $f_{i-K}, f_{i-K+1}, \dots, f_i$. Tokia situacija atsiranda kai signalas stebimas realiaame laike ir reikšmių iš ateities paprasčiausiai dar neturime, bet norima iš karto jį glodinti, galbūt skaičiuoti trendais paremtas prognozes. Pavyzdžiui, stebime akcijų kainos kitimą arba valiutų kurso svyravimą. Tuomet galima naudoti modifikuotą algoritmo (6), (7) variantą:

$$g_i = \sum_{j=0}^K w_j f_{i-j}, \quad i = K, K+1, \dots, N, \quad \sum_{j=0}^K w_j = 1.$$

Dar vieną signalo glodinimo algoritmą, pritaikytą paminėtai situacijai, išnagrinėsime 2.1.3 skyrelyje.

Skaičiavimų optimizavimo galimybė, naudojant greitosios Furje transformacijos algoritmą

Pažymėkime, kad jeigu apdorojamų duomenų kiekis N ir algoritmo parametras K yra pakankamai dideli skaičiai, galima šiek tiek modifikuoti slenkančio vidurkio su svoriais algoritmo formules taip, kad jos atitiktų 7.1.4 skyrelyje apibrėžtas išraiškas (konvoliuciją ir panašią į ją sumą). Tokiu būdu skaičiavimuose bus galima taikyti itin efektyvų (kompiuterio procesoriaus laiko požiūriu) greitosios Furje transformacijos algoritmą, aprašytą 7.2 skyrelyje.

2.1.3 Eksponentinis slenkančio vidurkio algoritmas

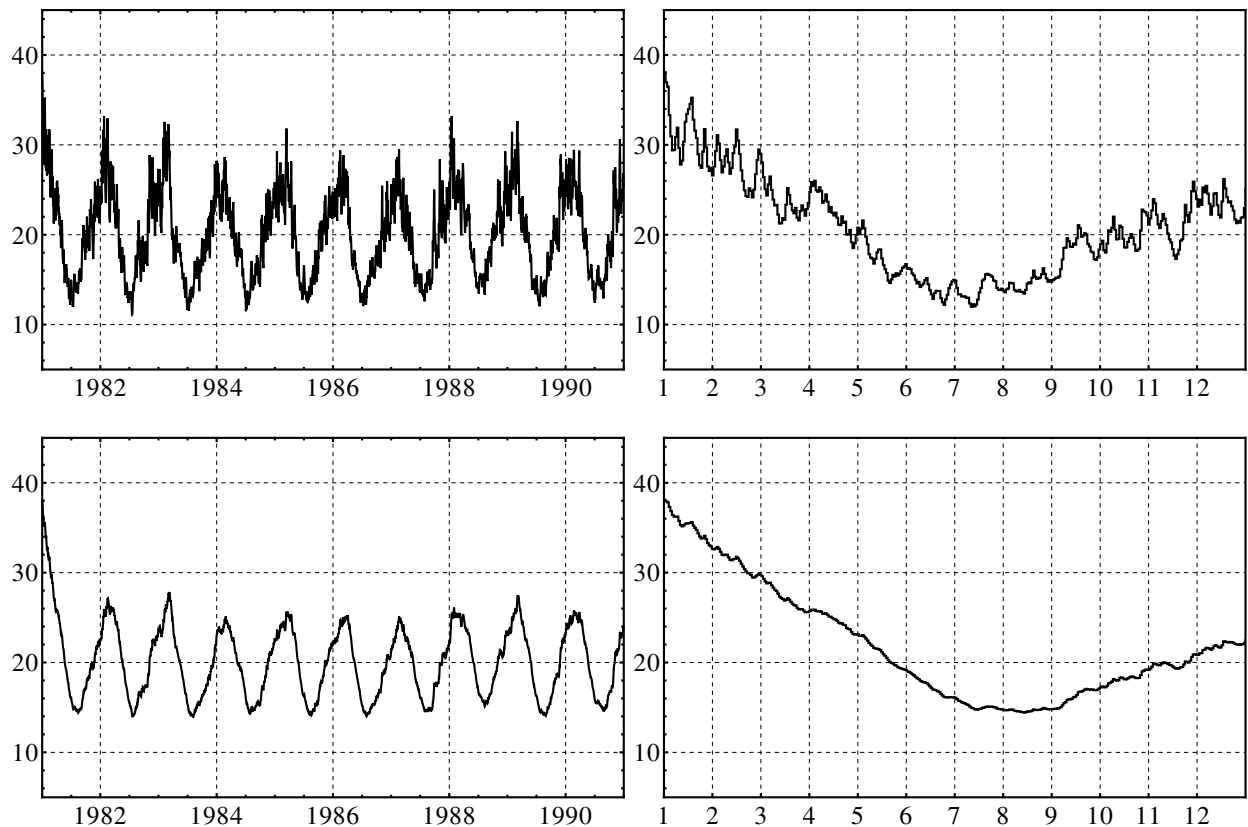
Trumpai apžvelgsime dar vieną slenkančio vidurkio idėja paremtą metodą – *eksponentinį slenkančio vidurkio* (angliškai “exponential moving average”) algoritmą. Šis metodas savo veikimo principu ir rezultatais panašus į slenkančio vidurkio su svoriais glodinimą, tačiau turi du privalumus:

1. Vietoje parametro K ir svorių rinkinio w_j yra tik vienas vartotojo kontroliuojamas algoritmo parametras: eksponentė (lietuviškai “daugiklis”) α ;
2. Signalą papildžius nauja reikšme pakanka žinoti tik šią naująją reikšmę bei paskutinę sugludinto signalo reikšmę – tuomet galime lengvai apskaičiuoti sugludinto signalo reikšmę naujuoju laiko momentu (prisiminkime, kad 2.1.1, 2.1.2 skyreliuose nagrinėti algoritmai vidurkinimo metu operuoja su ilgu signalo fragmentu).

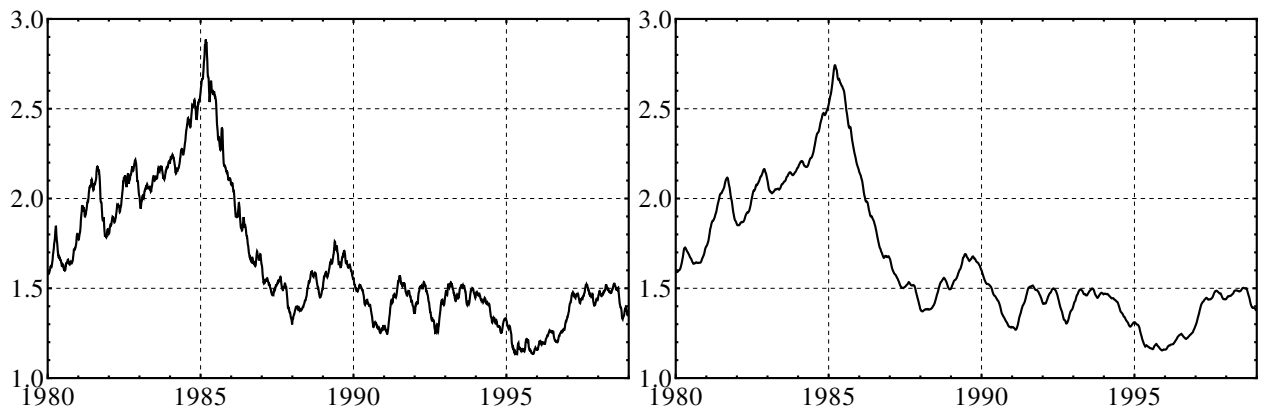
Tarkime, domimės valiutų kurso svyravimu, tai yra, signalo reikšmės f_i reiškia valiutų kursą i -tąją dieną. Kaip ir 2.1.1, 2.1.2 skyreliuose, g_i žymėsime sugludinto signalo reikšmes. Sakykime, šiandien yra diena, kurios numeris n , ir ką tik sužinojome šios dienos valiutų kursą f_n . Taip pat jau žinome vakar dienos sugludinto valiutų kurso signalo vertę g_{n-1} . Mūsų tikslas – apskaičiuoti šios dienos sugludintąją reikšmę g_n . Ją skaičiuojame pagal formulę

$$g_n = (1 - \alpha) g_{n-1} + \alpha f_n, \quad n = 1, 2, \dots, \quad g_0 = f_0, \quad (10)$$

čia $0 \leq \alpha \leq 1$ yra algoritmo parametras, kontroliuojantis signalo glodinimo lygį.



18 pav.: Analogiški rezultatai kaip 14 pav. ir 17 pav., tik gauti taikant eksponentinį slenkančio vidurkio algoritmą. Ilustracijos viršuje: $\alpha = 0.2$. Ilustracijos apačioje: $\alpha = 0.03$.



19 pav.: Analogiški rezultatai kaip 15 pav., tik gauti taikant eksponentinį slenkančio vidurkio algoritmą. Ilustracija kairėje: $\alpha = 0.2$. Ilustracija dešinėje: $\alpha = 0.03$.

Formulė (10) yra rekurentinė (dydis su aukštesniu indeksu g_n skaičiuojamas per jau apskaičiuotą dydį su žemesniu indeksu g_{n-1} , o pačio mažiausio indekso dydis g_0 yra duotas) ir reiškia skaičių g_{n-1} ir f_n vidurkį su svoriais $w_1 = 1 - \alpha$ ir $w_2 = \alpha$. Pastebėkime, kad analogiškai kaip (7), svoriai tenkina

$$w_1 + w_2 = 1.$$

Jeigu, pavyzdžiui, parinktume $\alpha = 1/2$, tai išraiška (10) reikštų aritmetinį vidurkinimą.

Kaip ir slenkančio vidurkio su svoriais algoritme, galima parodyti (iš formulės (10) išreiškus dydį g_n per dydžių f_0, f_1, \dots, f_n sumą ir sugrupavus narius prie vienodų α laipsnių), kad vidurkinimo procese originalaus signalo reikšmių f_i reikšmingumas (svorio koeficientas) tuo didesnis, kuo arti-

mesnis jų indeksas i einamajam indeksui n . Būtent todėl šis vidurkinimas ir yra praktiškai prasmingas: gyvenime naujausi duomenys (pavyzdžiui, šios dienos ar vos keleto dienų senumo valiutų kursai) yra kur kas svarbesni einamajai vidutinei reikšmei, palyginus su pasenusiais duomenimis.

Paėmus $\alpha = 1$, formulė (10) transformuojasi į $g_n = f_n$, taigi glodinimas nevyksta („suglodintas“ signalas tapatingas originaliam signalui). Pradėjus mažinti parametą α (primename, kad norint išlaikyti vidurkinimo su svoriais prasmę, galime imti tik $0 \leq \alpha \leq 1$), išraiškoje (10) atsiranda buvusios suglodintos reikšmės g_{n-1} įtaka ir prasideda signalo glodinimas. Kuo mažesnis $\alpha > 0$, tuo glodinimas intensyvesnis. Esant ribinei reikšmei $\alpha = 0$, gauname

$$g_n = g_{n-1}, \quad n = 1, 2, \dots, \quad g_0 = f_0,$$

taigi visi $g_n \equiv f_0$ – pernelyg suglodintas signalas tampa tiese. Kaip prisimename, lygiai taip pat nutinka 2.1.1 skyrelyje išnagrinėto slenkančio vidurkio algoritmui, neprotingai daug didinant parametą K .

Vadinasi, norint pasiekti geros glodinimo kokybės, reikia parametą α sumažinti, bet ne per daug. Praktiškai tai reiškia, kad konkrečiam signalui (ar signalų klasei) optimalus α parenkamas eksperimentiškai (analogiškai kaip eksperimentiškai parenkame geriausią K vertę slenkančio vidurkio algoritme).

Kompiuterinio eksperimento, taikant eksponentinį slenkančio vidurkio algoritmą signalams, tirtiems 2.1.1 ir 2.1.2 skyreliuose, rezultatai matomi 18 pav. ir 19 pav. Buvo skaičiuojama su dviem parametru α vertėmis: $\alpha = 0.2$ ir $\alpha = 0.03$. Paėmus $\alpha = 0.01$ ar dar mažesnes vertes, jau gaunami pernelyg suglodinti (pradėję esmines detales) signalai. Kaip matyti iš pateiktų iliustracijų, eksponentiniu slenkančio vidurkio algoritmu gaunami grafikai labai panašūs į paprasto slenkančio vidurkio ir slenkančio vidurkio su svoriais metodais gaunamus rezultatus.

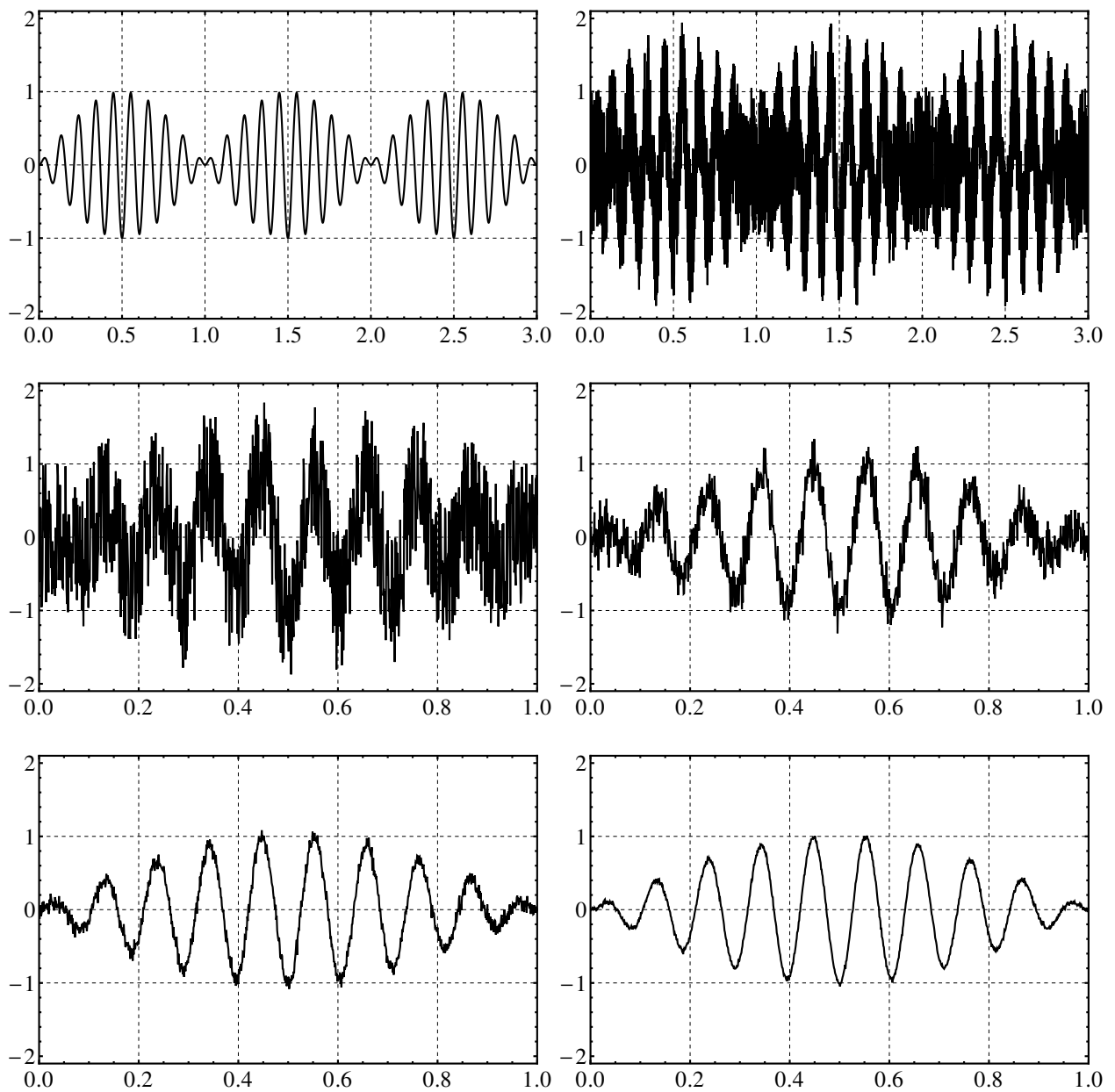
Galimas eksponentinio slenkančio vidurkio algoritmo apibendrinimas – kai formulėje (10) imamas priklausomas nuo einamojo indekso n parametras $\alpha = \alpha_n$. Dažniausiai kol apdorojamos signalo reikšmės numeris n yra pakankamai mažas (dar turime nedaug vidurkinamų duomenų) imamos α reikšmės artimos vienetui (vidurkinimo efektas silpnas), vėliau, didėjant n (pasirodant vis daugiau ir daugiau duomenų) – tolygiai mažiname α (stipriname vidurkinimo efektą), kol pasiekiamo tam tikrą ribinę reikšmę $\alpha > 0$. Dydžių α mažinimo mechanizmas ir ribinė reikšmė vėlgi parenkami eksperimentiškai konkrečiam signalui ar signalų klasei.

2.2 Sinchroninis triukšmų filtravimas periodiniams signalams

Labai dažnai eksperimentiniai signalai turi registravimo prietaiso triukšmą, kuris gali paslėpti svarbias signalo detales. Kaip sumažinti signale esančią triukšmo komponentę? Būtina pasakyti, kad pradėti reikėtų ne nuo algoritminio signalo apdorojimo. Pirmiausiai verta surasti triukšmo atsiradimo priežastis ir jas pašalinti. Pavyzdžiui, triukšmo priežastimi gali būti elektrinio signalo perdavimas ilgu kabeliu. Patartina registravimo aparatūroje patikrinti įtampas, kabelio būklę ir radus trūkumų – juos pašalinti. Tačiau galima ir tokia situacija, kai nieko pakeisti negalime (signalas jau eksperimentiškai užregistruotas ir tai pakartoti neįmanoma arba labai brangu) arba nepaisant visų pastangų nepavyksta pašalinti triukšmo atsiradimo registracijos prietaise. Tokiu atveju belieka mėginti eliminuoti triukšmus signalo apdorojimo algoritmais.

Jeigu triukšmas aukšto dažnio, o jo intensyvumas (amplitudė) santykinai (palyginus su signalo reikšmių dydžiu) nedidelis, galima jį eliminuoti iš signalo taikant vieną iš signalo glodinimo slenkančio vidurkio metodu algoritimų variantų, išdėstytų 2.1.1, 2.1.2 ir 2.1.3 skyreliuose. Tačiau šis signalo glodinimo būdas neefektyvus jeigu triukšmas intensyvus arba žemo dažnio. Vis dėlto, su sąlyga, kad turime *periodinį* signalą (arba daug vieno ir to pačio signalo registracijų skirtingais prietaisais, kurias

išrikiavę vieną po kito, simuliuotume vieną ilgą periodinį signalą), egzistuoja efektyvus metodas sumažinti jame bet kokius triukšmus. Šis metodas vadinamas *synchroniniu triukšmų filtravimu*.



20 pav.: Synchroninio triukšmų filtravimo algoritmo (11) iliustracija. Iliustracijos viršuje: periodinis signalas be triukšmų, pavaizduoti trys periodai (kairėje); užtriukšmintas signalas, pavaizduoti trys periodai (dešinėje). Iliustracijos viduryje: užtriukšmintas signalas, vienas periodas (kairėje); filtravime panaudojome $M = 10$ periodo ilgio laiko intervalų, rezultatas – filtruotas signalas $f_{filtruotas}(t)$, vienas periodas (dešinėje). Iliustracijos apačioje: $M = 100$, filtruotas signalas $f_{filtruotas}(t)$ (kairėje); $M = 1000$, filtruotas signalas $f_{filtruotas}(t)$ (dešinėje).

Metodo idėja paremta signalo sudaliniu į jo periodo ilgio laiko intervalus. Skirtingiems laiko intervalams priklausančios signalo reikšmės, paimitos tame pačiame periodo taške (t. y. turinčios tą pačią fazę), suvidurkinamos. Kadangi triukšmai turi atsitiktinį pobūdį, vidurkinant jie tarpusavyje kompensuojasi – sumažėja, o pats signalas išryškėja.

Tarkime, nagrinėjame signalą $f(t)$. Jis sudarytas iš naudingos (informatyvios) periodinės kom-

ponentės $s(t)$ ir apibrėžiančios triukšmą komponentės $n(t)$:

$$f(t) = s(t) + n(t).$$

Pažymėkime $f_k(t)$ signalo fragmentą, kai laiko kintamasis t priklauso k -tajam periodui. Triukšmo sudaromoji kiekviename laiko periode vis kitokia (priklauso nuo periodo numerio k), taigi pažymėkime ją $n_k(t)$. Tačiau signalo komponentė $s(t)$ yra periodinė, vadinasi jos reikšmės identiškai atsikartoja kiekviename periode. Taigi, kai t priklauso k -tajam periodui, turime

$$f_k(t) = s(t) + n_k(t).$$

Sakykime, signalo stebėjimo trukmė sudaryta iš M periodų. Suvidurkinkime signalo reikšmes pagal fragmentus atskiruose perioduose. Rezultatą, reprezentuojantį signalo $f(t)$ filtruotus duomenis (apibrėžtus viename laiko periode), pažymėkime $f_{\text{filtruotas}}(t)$:

$$f_{\text{filtruotas}}(t) = \frac{1}{M} \sum_{k=1}^M f_k(t) = \frac{1}{M} \sum_{k=1}^M s(t) + \frac{1}{M} \sum_{k=1}^M n_k(t) = s(t) + \frac{1}{M} \sum_{k=1}^M n_k(t). \quad (11)$$

20 pav. pateikti sinchroninio triukšmų filtravimo algoritmo rezultatai konkrečiam signalui. Iš jų darome išvadą, kad kuo daugiau periodų telpa duoto signalo registravimo trukmėje, tuo efektyviau veikia filtravimas, pagrįstas triukšmų vidurkinimo formule (11).

Kadangi praktikoje triukšmų komponentės $n_k(t)$ reikšmės su lygia tikimybe įgyja tiek teigiamas, tiek neigiamas vertes, vidurkinant per daugelį periodų jos viena kitą naikina. Tokiu būdu galima sumažinti net labai intensyvius triukšmus. Matematiškai tai reiškia, kad

$$\frac{1}{M} \sum_{k=1}^M n_k(t) \rightarrow 0, \quad M \rightarrow \infty,$$

todėl $f_{\text{filtruotas}}(t) \rightarrow s(t)$, kai $M \rightarrow \infty$.

Žinoma, praktiškai imamas ne begalinis, bet pakankamai didelis signalo periodų skaičius M .

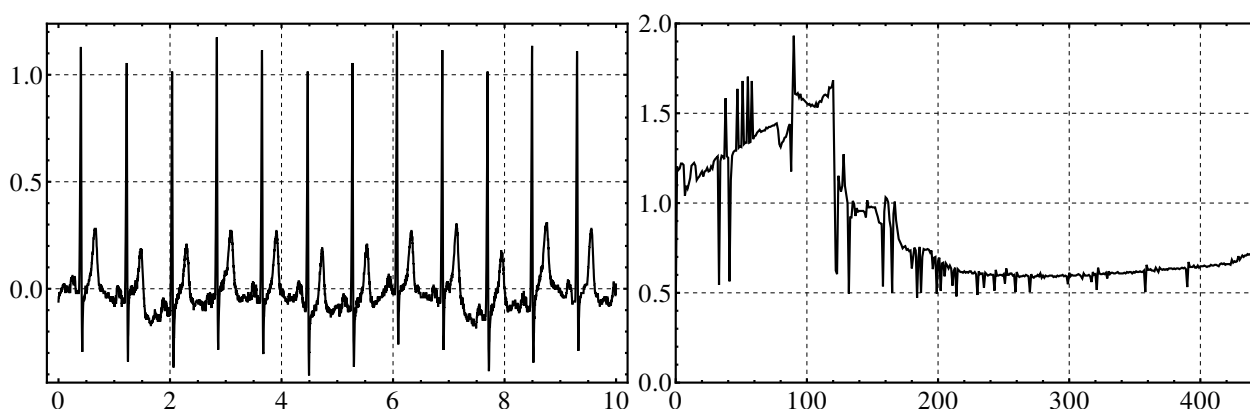
Kaip jau buvo minėta, sinchroninio triukšmų filtravimo algoritmą galima taikyti ir ne tik periodiniams signalams, tačiau ir tokioms duomenų serijoms, iš kurių galima imituoti periodiškumą. Pavyzdžiui, registruojame vieną ir tą patį signalą, naudodami skirtingus prietaisus. Surenkame daug duomenų porcijų, besiskiriančių tik triukšmine komponente. Nuosekliai išdėstę visas porcijas vieną po kito, tarsi gauname vieną periodinę signalą (duomenų porcijų kiekis atitinka periodų skaičių). Taigi, galime taikyti šiame skyrelyje išdėstytą filtravimo metodą.

3 Signalų rekurentinės diagramos

3.1 Kiek sudėtingas yra tiriamas procesas?

Vienas įdomiausių klausimų, kylančių tiriant signalus – įvertinti juos skleidžiančio šaltinio sudėtingumą (netrivialumą). Įvairiausios prigimtys vyksmai gali būti sąlygiškai klasifikuojami į paprastus (pakankamai trivalios prigimtys), sudėtingesnius ir labai sudėtingus. Paprasto proceso pavyzdžiu gali būti svyruojanti švytuoklė, sudėtingesnio – vandens bangavimas, įmetus į jį akmenį, labai sudėtingo – akcijų biržos indekso svyravimas. Žinoma yra daug tarpinių sudėtingumo lygių.

Panašius klausimus apie sudėtingų procesų elgsenos dėsningumus, jų klasifikavimą ir teorinį paaiškinimą kelia mokslo sritis, vadinama sinergetika. Ką visa tai turi bendro su signalų apdorojimu ir praktiniais taikymais?



21 pav.: Iliustracija kairėje: elektrokardiogramos V4 kanalo fragmentas (ordinačių ašyje įtampa [mV], abscisų ašyje laikas [s]), pagal [9]. Iliustracija dešinėje: iš elektrokardiogramos gautų RR intervalų (laiko tarpų tarp dviejų elektrokardiogramoje iš eilės einančių R bangų, kitaip tariant tarp dviejų širdies “suplakimų”) signalas (ordinačių ašyje RR intervalų trukmės [s], abscisų ašyje RR intervalų eilės numeriai), pagal [9].

Tarkime, nagrinėjame elektrokardiogramą – žmogaus širdies veiklą atspindintį signalą. Net ir nespecialistas elektrokardiogramoje gali pastebėti daugybę panašių (atsikartojančių) fragmentų (žr. 21 pav., iliustraciją kairėje). Tačiau vis dėlto jie nėra identiški. Pakanka išmatuoti laiko tarpus tarp dviejų iš eilės einančių “smailių” (kardiologijoje vadinamų R bangomis) ir matysime, kad jie bėgant laike elgiasi toli gražu netrivialiai (žr. 21 pav., iliustraciją dešinėje). Šie laiko tarpai, vadinami RR intervalais, praktiškai reiškia laiko trukmę tarp dviejų širdies dūžių (pulso). Kartais (dėl fizinio krūvio ar nervinio streso) pulsas padažnėja, kartais (miegant, ilsintis) tampa retesnis, taip pat egzistuoja laiko ruožai kada RR intervalų signalas atrodo kaip konstanta (širdis plaka kaip laikrodis – periodiškai).

Sveiko žmogaus širdis pasižymi itin sudėtinga elgsena, turi keletą savireguliacinių mechanizmų. Pasirodo, kad tai kaip ji po patirto streso pereina į režimą artimą periodiniam, kiek apytikslių periodų stebima ilgą laiką registruojant jos darbą ir koku būdu tie periodai keičia vienas kitą (visa tai telpa po anglišku terminu “heart rate variability”) – turi daug informacijos apie širdies galimybes savireguliuotis, tuo pačiu – apie žmogaus sveikatos būklę.

Šia prasme – pagal tai kiek skirtingų periodų stebima, kaip dažnai kiekvienas iš jų registruojamas, ar galimi nukrypimai nuo lokaliai laike periodinio elgesio ir perėjimas į neperiodinį ar netgi pseudoatsitiktinį bei vėl sugrįžimas (galbūt jau prie kito) periodo, gali būti tiriamos ir klasifikuojamos daugelis sistemų, ne tik medicinoje.

Pavyzdžiui, paprasta švytuoklė turi vienintelį periodą – taigi jos elgesys trivialus (primityvus), savireguliacinių savybių nėra. Taip pat primityvi (be savireguliacijos) yra visiškai atsitiktinai besielgianti sistema (atsitiktinis triukšmas). Bet kuri sistema turinti keletą periodų, bet nerodanti neperiodinio elgesio tarp jų irgi gana paprasta. Iš neperiodinio elgesio į periodinį (be išorinio poveikio) sugeba pereiti tik savireguliacija pasižymintys reiškiniai, o jie paprastai yra labai sudėtingai priklausantys nuo didžiulio kiekio faktorių. Tokių reiškinių ir sistemų pavyzdžiai: apytiksliai periodinis (atsirandantis maždaug kas penki metai Ramiajame vandenyne Pietų pusrutulyje) klimato darinys El Niño, daugelis meteorologinių dydžių, akcijų, naftos, aukso kainos, valiutų kursai, epidemijų intensyvumas, daugelis medicininės kilmės signalų, transporto srautų intensyvumas, kompiuterinių tinklų serverių apkrovimas.

Tačiau kiekviena sudėtinga sistema turi tik jai būdingą periodiškumo laipsnio ir nestacionarumo derinį, savotišką savo “piršto antspaudą”.

Kalbant signalų kalba, galima pasakyti, kad ne tik periodiniai signalai turi tikslaus ar bent apytikslio atsikartojimo (kvazi-periodiškumo) fragmentus. Šiuo atveju mus domina kiek šie kvazi-periodiškumo fragmentai yra atsitiktiniai, arba kiek reguliarūs.

Šiame skyriuje nagrinėsime skaitmeninių signalų analizės instrumentą, vadinamą *rekurentine diagrama* (angliškai “recurrence plot”). Rekurentinės diagramos būtent ir naudojamos grafiniu būdu perteikti signalo *nestacionarumo pobūdį* bei *periodiškumo laipsnį*. Jų pavadinimas kilęs iš angliško žodžio “recurrence”, reiškiančio pasikartojimą. Taigi rekurentinė diagrama yra grafinis pasikartojimų (esančių signale) vaizdas.

Svarbus rekurentinių diagramų privalumas yra tas, kad jos gali būti taikomos gana trumpiems signalams (surinkus sąlyginai nedaug stebėjimo duomenų). Klausimai į kuriuos galima atsakyti vizualiai susipažinus su konkreto signalo rekurentine diagrama yra šie:

- Ar signalas yra periodinis arba kvazi-periodinis? Ar egzistuoja keli periodai keičiantys vienas kitą? Jeigu taip, kam šie periodai lygūs ir kaip ilgai jie išsilaiko?
- Ar signalas stacionarus ar nestacionarus? Jeigu nestacionarus – kaip stipriai pasikeičia jo statistinės savybės bėgant laikui?
- Ar signalas turi tendų?
- Ar signalo reikšmių svyravimuose stebimas atsitiktinumas kai signalo reikšmės tampa nepriklausomomis nuo reikšmių artimoje praeityje? Galbūt signale yra ir periodinių laiko fragmentų, ir atsitiktinio elgesio laikotarpių?
- Ar yra laiko tarpų kada signalo reikšmės nesikeičia arba beveik nesikeičia?

Atsakymai į šiuos bei kitus klausimus randami 3.3 skyrelyje.

Daug teorinės ir praktinės medžiagos, susijusios su rekurentinių diagramų taikymais signalų analizėje pateikta [11, 12]

3.2 Rekurentinės diagramos apibrėžimas ir savybės

Tarkime, duotas skaitmeninis signalas (laiko eilutė), apibrėžtas $N+1$ reikšmėmis

$$f_0, f_1, \dots, f_N.$$

Sudalinkime duotas signalo reikšmes į poras

$$y_0 = (f_0, f_1), \quad y_1 = (f_1, f_2), \quad \dots, \quad y_{N-1} = (f_{N-1}, f_N).$$

Matome, kad gavome vektorių y_i (kurių dimensija $D = 2$) seką. Šie vektoriai vadinami *signalo būsenomis*, šį terminą paaiškinsime netrukus 3.2.1 skyrelyje.

Prisiminkime, kad mūsų tikslas – tirti ar yra pasikartojančių (apytikslė prasme) signalo fragmentų. Todėl palyginsime visas galimas vektorių y_i poras. Jeigu surasime du skirtingus vektorius (signalo būsenas)

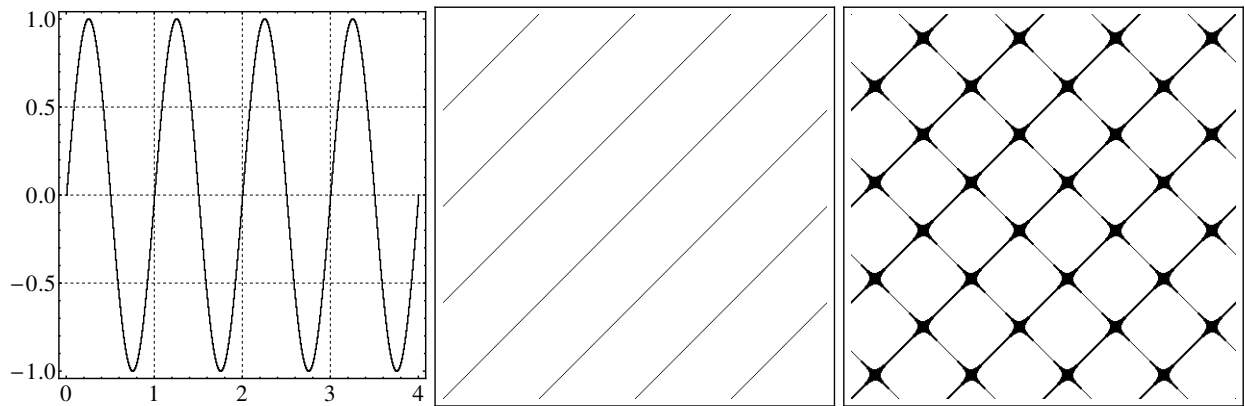
$$y_i = (f_i, f_{i+1}) \quad \text{ir} \quad y_j = (f_j, f_{j+1}),$$

kurie beveik lygūs tarpusavyje, laikysime, kad atitinkami diskretaus laiko indeksai i ir j žymi dviejų beveik lygių tarpusavyje (pasikartojančių) signalų fragmentus.

Rekurentinė diagrama yra iš taškų (pikselių) sudarytas taškinis paveikslėlis (angliškai “bitmap”). Kiekvienas taškas yra baltos arba juodos spalvos. Būtent, paveikslėlio taškas su koordinatėmis (i, j) yra juodos spalvos, jeigu vektorius y_i yra nutolęs nuo vektoriaus y_j mažiau nei per tam tikrą atstumą r :

$$\|y_i - y_j\| \leq r, \quad i, j = 0, 1, \dots, M, \quad r > 0 \quad \text{pasirinktas parametras}, \quad (12)$$

kitu atveju šis taškas rekurentinėje diagramoje yra baltos spalvos. Čia $M+1$ yra vektorių y_i kiekis (mūsų nagrinėjamu atveju $M = N - 1$), taigi gaunamas rekurentinės diagramos paveikslėlis yra kvadratas sudarytas iš $(M+1) \times (M+1)$ taškų. Juodi taškai žymi signalo būsenos pasikartojimus, žr. 22 pav.



22 pav.: Periodinis sinusinės bangos tipo signalas (ilustracija kairėje) ir jo rekurentinės diagramos su $r = 0.001$ (ilustracija centre) ir su $r = 0.1$ (ilustracija dešinėje). Signalų imčių skaičius $N = 400$. Signalų būsenų dimensija $D = 2$, vėlinimas $d = 1$ (žr. 3.2.1 skyrelį). Pastebimos rekurentinėse diagramose įstrižaininės linijos liudija apie signalo periodiškumą.

Rekurentinę diagramą apibrėžiančioje nelygybėje (12) dydis $r > 0$ yra vartotojo pasirenkamas algoritmo parametras, vadinamas *slenkstiniu atstumu*. Jeigu ieškotume laiko momentų, kada signale užregistruotos būsenos tiksliai pasikartoja, imtume $r = 0$, tačiau praktiškai domimės signalo kvazi-periodiškumu (apytiksliais būsenų pasikartojimais), taigi renkamės teigiamą $r > 0$. Šio dydžio prasmė – kokio didumo skirtumą tarp signalo būsenų toleruojame, kad du signalo fragmentus vis dar laikytume pasikartojančiais. Į kvazi-periodiškumo sąvoką įtraukiame tiek signalo paklaidas (dėl kurių jis nebus idealiai periodinis), tiek registruoto proceso įpatybes (gyvenime daugelis reiškinių nuolat atsikartoja, tačiau neidentišškai).

Apie tai kaip nelygybėje (12) skaičiuojamas atstumas tarp dviejų vektorių, kalbėsime 3.2.2 skyrelyje.

Didelio duomenų kiekio problema ir jos sprendimas

Pastebėkime, kad skaičiavimo pagal kriterijų (12) laikas auga kvadratiškai, didinant diskretizuoto signalo imčių skaičių N . Todėl turint pakankamai išsamius (ilgos trukmės ir/arba su aukštu diskretizavimo dažniu skaitmenizuotus) signalus galime nesulaukti skaičiavimų pabaigos. Be to kyla ir kita techninė problema – labai aukštos rezoliucijos paveikslėlio peržiūra ir rezultatų interpretavimas. Tokiu atveju tenka dirbtinai sutrumpinti signalą, kurio rekurentinę diagramą ruošiamės skaičiuoti: arba suskaidžius jį į fragmentus (galėsime paskaičiuoti seriją rekurentinių diagramų ir jas tarpusavyje palyginti), arba praretinus (angliškas terminas “downsampling”) skaitmeninio signalo reikšmes, paliekant, pavyzdžiui, tik kas dešimtą.

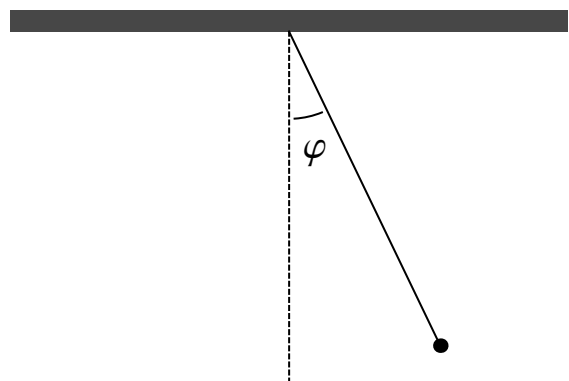
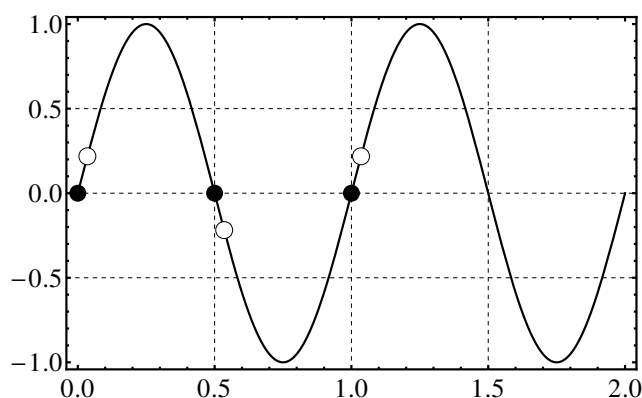
Taigi, rekurentinės diagramos algoritmas iš esmės praktiškai veikia tik su sąlyginai nedideliu N (taikymuose imama N tarp 100 ir 1000). Tai greičiau yra jo privalumas nei trūkumas: kaip taisyklė tokio duomenų kiekio pakanka dėsningumams išvelgti šiuo metodu. Be to, tai reiškia, kad algoritmą galima taikyti ir neturint eksperimentuose surinkto didelio duomenų kiekio.

3.2.1 Signalų būsenos sąvoka

Pastebėkime, kad vietoje signalo reikšmių f_i pradėjome vartoti terminą *signalų būseną*

$$y_i = (f_i, f_{i+1}), \quad i = 0, 1, \dots, M, \quad M = N-1. \quad (13)$$

Gali kilti klausimas: kodėl lyginame ne tiesiogiai signalo reikšmes f_i ir f_j , bet signalo būsenas $y_i = (f_i, f_{i+1})$ ir $y_j = (f_j, f_{j+1})$? Atsakymas į šį klausimą paprastas: tai kad sutaps dvi skirtingos signalo reikšmės f_i ir f_j , toli gražu negarantuoja, kad bus lygios tarpusavyje ir sekančios signalo vertės f_{i+1} ir f_{j+1} !



23 pav.: Iliustracija kairėje: demonstracija, kad norint surasti periodinius signalo fragmentus, nepakanka lyginti vien tik signalo reikšmes f_i ir f_j (pavaizduotos skrituliais juodu fonu), turi sutapti dar ir sekančios reikšmės f_{i+1} ir f_{j+1} (pavaizduotos skrituliais baltu fonu). Iliustracija dešinėje: švytuoklės schema (punkttyrine linija pažymėta švytuoklės pusiausvyros padėtis).

Panagrinėkime, pavyzdžiui signalą $f(t) = \sin(2\pi t)$, kurio periodas lygus vienetui. 23 pav. iliustracijoje kairėje skrituliais juodu fonu pažymėjime sutampančias signalo reikšmes f_i , o skrituliais baltu fonu – sekančias diskretizuoto signalo reikšmes f_{i+1} . Matome, kad reikšmės $f_i = f(0)$ ir $f_j = f(0.5)$ sutampa, tačiau tai nereiškia, kad signalas turi periodą, lygų 0.5! Juk viena sekanti

reikšmė f_{i+1} yra teigiama, o kita f_{j+1} – neigiama. Tačiau, jeigu palyginame kitą reikšmių porą $f_i = f(0)$ ir $f_j = f(1)$, tai matyti, kad tarpusavyje lygios ir sekančios signalo vertės f_{i+1} bei f_{j+1} . Vadinasi, signalas turi periodą 1, bet neturi periodo 0.5.

Analogiškai motyvuojantis pavyzdys: švytuoklė, kurios schema pateikta 23 pav. iliustracijoje dešinėje. Galima naiviai manyti, kad švytuoklės judėjimui aprašyti pakanka žinoti jos atsilenkimo kampo φ nuo pusiausvyros padėties reikšmę $\varphi(0)$ pradinio laiko momentu $t = 0$. Iš tiesų, jeigu švytuoklę atlenkę duotu kampu paleisime laisvai (be pradinio greičio) svyruoti, jos judėjimas bus vienoks, o jeigu dar suteiksime ir pradinį impulsą (stumtelsime), svyravimas bus pagal kitą scenarijų, nepaisant to, kad pradinis kampas $\varphi(0)$ abiem atvejais sutaps. Taigi tolesniam švytuoklės elgesiui nuspėti nepakanka žinoti vien tik jos atsilenkimo kampo reikšmę kažkokiu laiko momentu t . Reikia žinoti švytuoklės būseną

$$(\varphi(t), \varphi'(t)),$$

čia $\varphi'(t)$ yra kampo kitimo greitis, matematiškai apskaičiuojamas kaip kampo funkcijos išvestinė. Iš tiesų, išsprendus idealios (be trinties) tiesinės švytuoklės matematinį modelį, parodoma, kad ji svyruoja pagal trigonometrinį dėsnį, taigi abu 23 pav. pateikti pavyzdžiai turi bendrumų.

Šie pavyzdžiai paaiškina, kodėl ieškant signalo periodiškumo fragmentų (pasikartojimų) lyginamos ne signalo reikšmės, bet būsenos (13). Taip pat atkreipkime dėmesį, kad antroji būsenos komponentė f_{i+1} (tiksliau pokytis $f_{i+1} - f_i$) turi informaciją apie signalo kitimo greitį, taigi vietoje signalo reikšmės ir signalo svyravimo greičio konkrečiu laiko momentu, patogiau naudoti dviejų iš eilės einančių signalo reikšmių porą būsenos apibrėžime (13).

Signalų būsenos sąvokos apibendrinimas

Pagalvoję galėtume pateikti daug gyvenime stebimų procesų pavyzdžių, kurių evoliuciją (kitimą) lemia ne tik jų reikšmės pradinio atskaitos (stebėjimo) laiko momentu, bet ir pradinis greitis, o neretai ir pradinis pagreitis bei kiti faktoriai. Šie pastebėjimai veda prie minčių, kad galbūt reikėtų apibrėžti būseną ne kaip dvimatį, bet kaip aukštesnės dimensijos vektorių.

Todėl kartais skaičiuojant rekurentines diagramas signalo būsenos apibrėžiamos ne kaip dvimačiai vektoriai (13), bet kaip D -dimensiniai vektoriai, o pačios signalo reikšmės grupuojamos su diskretaus laiko *vėlinimo* parametru d :

$$y_i = (f_i, f_{i+d}, f_{i+2d}, \dots, f_{i+(D-1)d}), \quad i = 0, 1, \dots, M, \quad M = N - (D - 1)d, \quad (14)$$

čia natūriniai skaičiai $D \geq 2$, $d \geq 1$ yra algoritmo parametrai.

Pavyzdžiui, jeigu $D = 3$, $d = 2$, tai vektoriai y_i apibrėžiami taip:

$$y_0 = (f_0, f_2, f_4), \quad y_1 = (f_1, f_3, f_5), \quad \dots, \quad y_{N-4} = (f_{N-4}, f_{N-2}, f_N).$$

Vis dėlto, dažniausiai skaičiuojant rekurentines diagramas apsiribojama supaprastintu būsenų apibrėžimu su $D = 2$, $d = 1$. Visose šio skyriaus iliustracijose mes naudojome būtent šias parametrų vertes. Tačiau analizuojant konkretų signalą, verta paeksperimentuoti ir su kitomis reikšmėmis, pavyzdžiui $D = 3$ arba $D = 4$, taip pat su $d = 2$, $d = 3$, kiekvienu atveju palyginant, ar neatsiranda esminių skirtumų rekurentinėje diagramoje. Teoriškai geriausias D pasirinkimas būtų artima reikšmė signalo kilmės proceso *koreliacinei dimensijai* D_c : $D \approx D_c$, o vietoje d teisingiausia rinktis reikšmę, proporcingą tai su kuria signalo *autokoreliacijos funkcija* pirmą kartą tampa artima nuliui (jos modulis neviršija $2/\sqrt{N}$). Apie koreliacinę dimensiją ir jos nustatymą kalbėsime 4 skyriuje, o apie autokoreliacijos funkciją – 5 skyriuje. Sukurta su dydžių D ir d parinkimu susijusi matematinė teorija apie fazinės erdvės rekonstravimą, žr. [12] ir ten cituojamus literatūros šaltinius.

3.2.2 Atstumo tarp dviejų būsenų skaičiavimo būdai

Aptarsime kaip rekurentinę diagramą apibrėžiančioje nelygybėje (12) galima skaičiuoti atstumą tarp dviejų D -dimensinių vektorių (signalų būsenų).

Atstumas tarp vektorių y_i ir y_j , žymimas $\|y_i - y_j\|$, apibrėžiamas panaudojant funkciją, vadinamą *metrika* (sinonimas: *norma*).

Dažniausiai naudojama metrika yra Euklidinė, apskaičiuojanti tiesioginį atstumą tarp D -dimensinio taško $u = (u_{(1)}, u_{(2)}, \dots, u_{(D)})$ ir koordinačių pradžios taško:

$$\|u\| = \sqrt{u_{(1)}^2 + u_{(2)}^2 + \dots + u_{(D)}^2}.$$

Dvimatėje arba trimatėje erdvėje ši atstumo formulė išvedama remiantis Pitagoro teorema. Literatūroje ši metrika dar vadinama vektoriaus L_2 norma. Vietoje u įsistatę dviejų vektorių skirtumą $u = y_i - y_j$ ir apskaičiuojame Euklidinį atstumą tarp vektorių y_i, y_j .

Alternatyviai gali būti naudojamos ir kitos metrikos, pavyzdžiui Manheteno norma

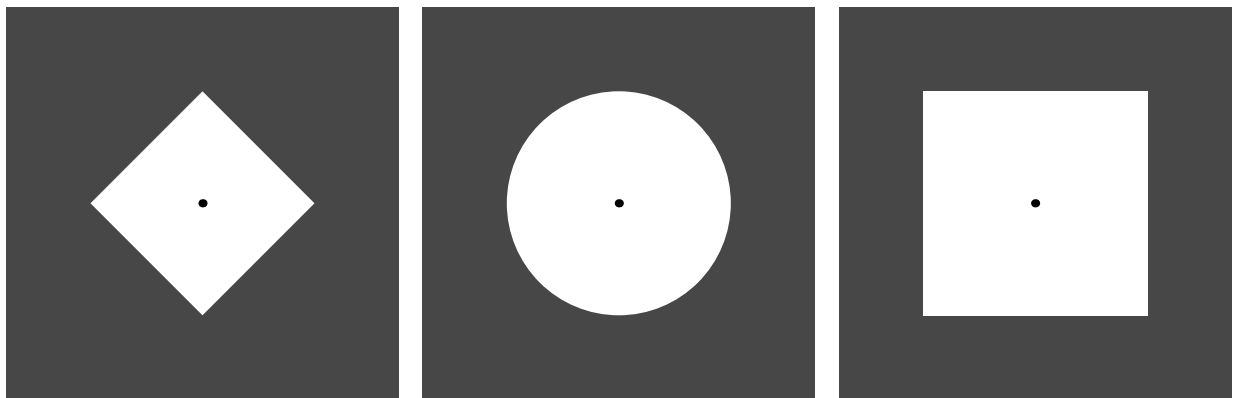
$$\|u\| = |u_{(1)}| + |u_{(2)}| + \dots + |u_{(D)}|,$$

dar vadinama vektoriaus L_1 norma ir maksimumo norma

$$\|u\| = \max(|u_{(1)}|, |u_{(2)}|, \dots, |u_{(D)}|),$$

taip pat vadinama vektoriaus L_∞ norma. Pastebėkime, kad maksimumo normos apskaičiavimas reikalauja mažiausiai kompiuterio procesoriaus laiko. Imliausia skaičiavimo trukmės požiūriu yra Euklidinė norma.

Šių trijų metrių veikimas dvimatėje ($D = 2$) erdvėje schematiškai iliustruotas 24 pav.



24 pav.: Trys dažnai naudojamos metrikos (normos), apibrėžiančios to paties spindulio aplinką (iliustruotą baltu fonu) aplink tašką, pažymėtą juodu tašku. Dėl vaizdumo iliustracijos pateiktos dvimatėje ($D = 2$) būsenų erdvėje. Iliustracija kairėje: L_1 metrika, dar vadinama Manheteno norma. Iliustracija centre: L_2 metrika, taip pat vadinama Euklidine norma. Iliustracija dešinėje: L_∞ metrika, taip pat vadinama maksimumo norma.

Verta pastebėti, kad nors parinkus pakankamai mažą r signalo rekurentinė diagrama esminiai nepriklauso nuo to kuri iš metrių naudojama nelygybėje (12), taikant maksimumo normą kriterijus (12) patenkinamas teoriškai šiek tiek dažniau (rekurentinėje diagramoje bus neženkliai daugiau juodų taškų), palyginus su kitomis dviem metrikomis. Taikant Manheteno normą gausime nežymiai mažiau juodų pikselių, o pagal Euklidinę normą – tarpinį jų skaičių. Ši išvada seka iš sričių baltu fonu plotų palyginimo 24 pav.

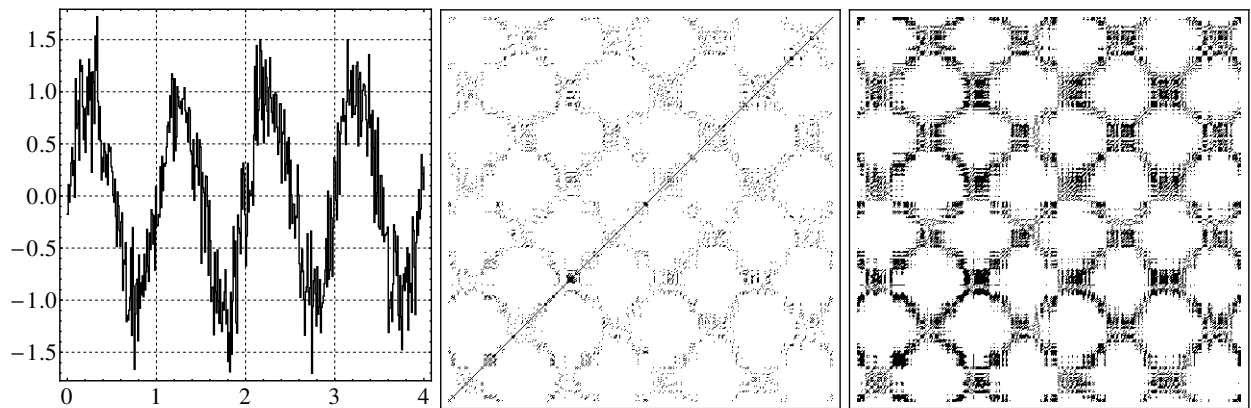
Šio skyriaus iliustracijoje mes naudosime Euklidinę metriką.

3.2.3 Rekurentinės diagramos savybės. Slenkstinio atstumo r parinkimas

Tiesiogiai iš rekurentinės diagramos apibrėžimo naudojant nelygybę (12), išplaukia šios pagrindinės savybės:

- Bet kokio signalo rekurentinėje diagramoje visi jos pagrindinėje įstrižainėje esantys taškai (su koordinatėmis (i, i)) yra juodi, koks bebūtų slenkstinis atstumas $r > 0$. Tai seka iš to, kad jeigu $i = j$, tai $y_i - y_i$ yra nulinis vektorius, kurio norma visada lygi nuliui.
- Kiekviena rekurentinė diagrama visada yra simetrinė pagrindinės įstrižainės atžvilgiu. Tai yra, jos taškų su koordinatėmis (i, j) ir (j, i) spalva visada sutampa. Taip yra todėl, kad $\|y_i - y_j\| = \|y_j - y_i\|$.
- Jeigu mažiname slenkstinio atstumo parametą r , tai juodų taškų rekurentinėje diagramoje kiekis mažėja (kadangi nelygybė (12) patenkinama vis rečiau). Galiausiai, jeigu $r \rightarrow 0$, tai juodi taškai lieka tik pagrindinėje įstrižainėje. Tiesa, idealiai periodiniam signalui teoriškai net ir su $r = 0$ turėtų išlikti juodos visos įstrižainės nutolusios nuo pagrindinės per periodo ilgį (matomos, pavyzdžiui, 22 pav., iliustracijoje centre), tačiau praktiškai dėl kompiuterinių skaičiavimų paklaidų ir jų gali nelikti.
- Didinant slenkstinio atstumo parametą r , juodų taškų rekurentinėje diagramoje kiekis didėja (nes nelygybė (12) patenkinama vis dažniau). Parametrui r viršijus tam tikrą skaičių (lygų $\max_{i,j} \|y_i - y_j\|$, kurį galima įvertinti per signalo reikšmių maksimalią amplitudę), visi rekurentinės diagramos pikseliai tampa juodais.

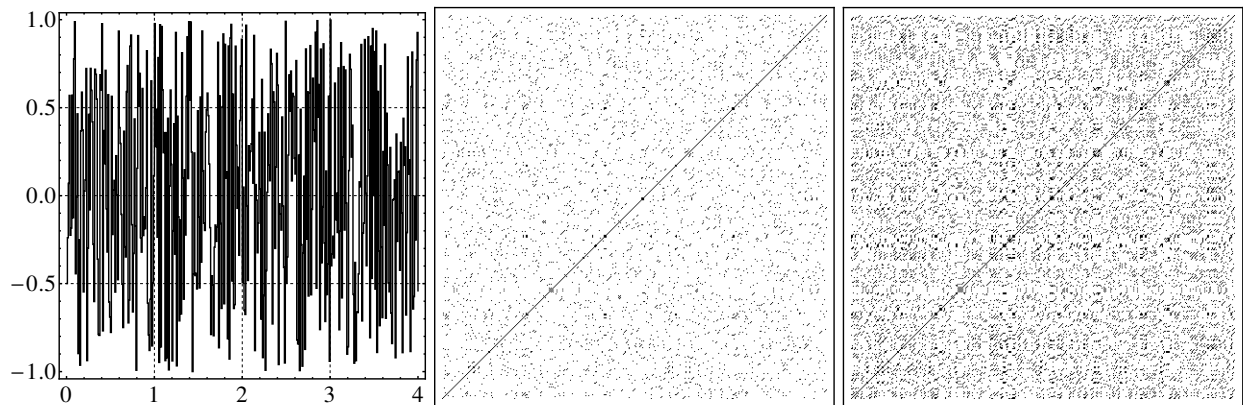
Šias savybes galima ir vizualiai pastebėti šiame skyriuje pateiktose iliustracijose, pavyzdžiui, 22 pav., 25 pav., 26 pav., 27 pav., ir kitose.



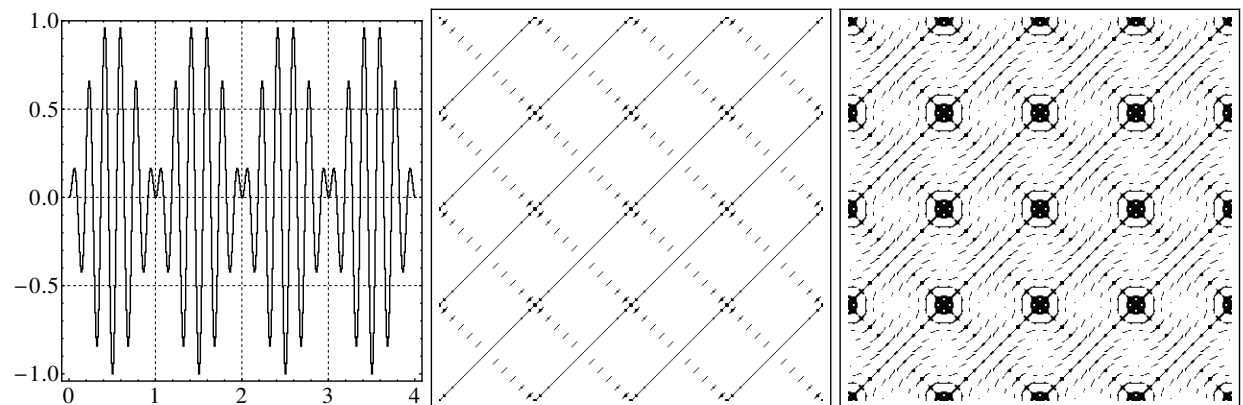
25 pav.: Užtriukšmintas sinusinės bangos tipo signalas (ilustracija kairėje) ir jo rekurentinės diagramos su $r = 0.2$ (ilustracija centre) ir su $r = 0.5$ (ilustracija dešinėje). Signalų imčių skaičius $N = 400$.

Vienas pagrindinių klausimų, kylančių vizualizuojant signalo savybes rekurentinės diagramos algoritmu: kaip parinkti slenkstinį atstumą r ?

Kaip matyti iš rekurentinės diagramos paskutinių dviejų savybių, per mažos arba per didelės parametro r reikšmės pasirinkimas padaro rekurentinę diagramą neinformatyvia. Jeigu r per mažas, rekurentinėje diagramoje neišryškėja signalo kvazi-periodiškumo savybės. Tačiau jeigu r per didelis, tai netgi diskrečiame laike iš eilės einančios signalo reikšmės detektuojamos kaip periodinės. Dėl



26 pav.: Atsitiktinio triukšmo signalas (ilustracija kairėje) ir jo rekurentinės diagramos su $r = 0.2$ (ilustracija centre) ir su $r = 0.4$ (ilustracija dešinėje). Signalo imčių skaičius $N = 400$. Netvarkingas bet vienodo tankio per visą rekurentinę diagramą juodų taškų išsibartymas liudija apie signalo reikšmių nereguliarumą.



27 pav.: Signalas su vienu periodu ir vienu kvazi-periodu (ilustracija kairėje) ir jo rekurentinės diagramos su $r = 0.05$ (ilustracija centre) ir su $r = 0.2$ (ilustracija dešinėje). Signalo imčių skaičius $N = 400$.

šio efekto su dideliais r apskaičiuotose rekurentinėse diagramose periodiškumo įstrižainės tampa storesnėmis ir ilgesnėmis nei yra iš tikrųjų (žr. 22 pav., 25 pav.). Vis dėlto, jeigu signalas užtriukšmintas, mes esame priversti rinktis didoką r vertę, kitaip dėl triukšmo sukeltų fluktuacijų neidentifikuosime daugelio periodinių darinių, esančių signale.

Taigi r parinkimas yra kompromisinis uždavinys.

Tam tikroms signalų klasėms literatūroje yra siūlomos įvairios sudėtingos automatinės r parinkimo strategijos, žr. [12] ir ten cituojamus literatūros šaltinius. Tačiau visada galima taikyti vieną iš paprastų metodų geriausios r reikšmės nustatymui konkrečiam signalui.

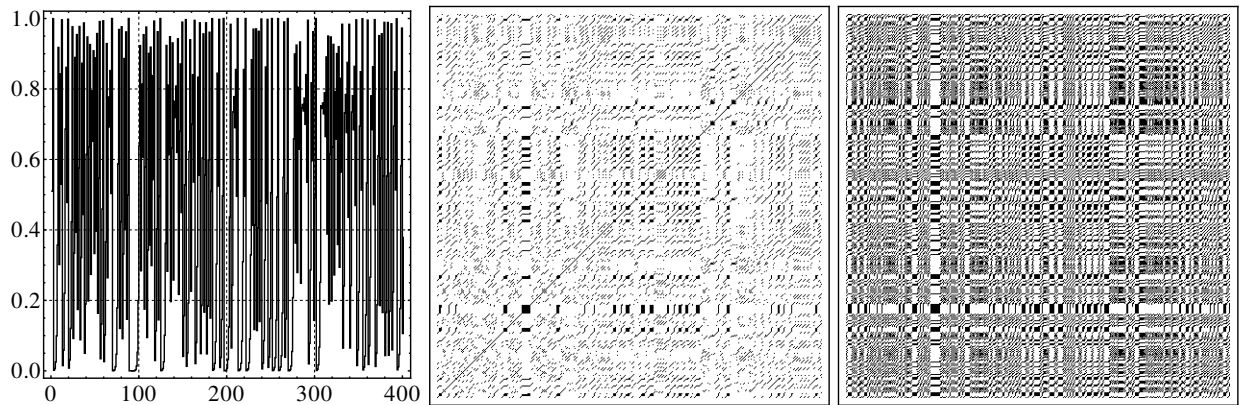
Vienas iš tokių būdų yra leisti vartotojui pasirinkti bet kokią r reikšmę, tada apskaičiuoti tris rekurentines diagramas, atitinkančias slenkstinius atstumus $r/2$, r ir $2r$. Gauti rezultatai dažniausiai leidžia nustatyti tendenciją – ar r turi būti mažinamas, ar didinamas, ar jau yra patenkinamas. Toliau kartojame procesą.

Kitas būdas yra pagal panašų metodą leisti kompiuteriui automatiškai didinti arba mažinti r reikšmes (pradedant nuo kažkokios fiksuotos), kol juodų taškų procentas rekurentinėje diagramoje pasiekia nurodytą proporciją, pavyzdžiui, 1%.

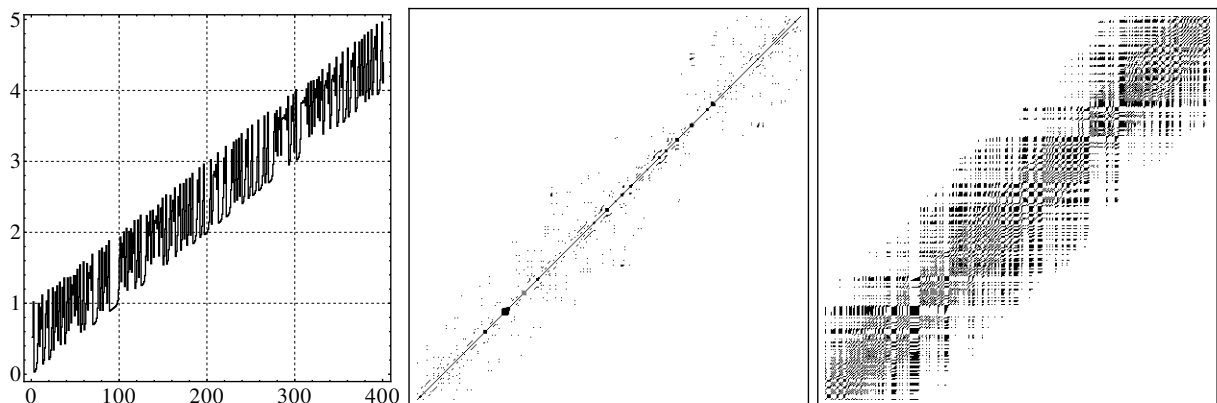
Kartais yra piešiamos spalvotos rekurentinės diagramos, skirtingomis spalvomis koduojant įvairias

r reikšmės, taip pavaizduojant visą spektrą rekurentinių diagramų. Tokiu atveju pasirenkama spalvų paletė, pavyzdžiui tokia, kurioje mėlyna spalva atitinka mažas r vertes, o raudona – dideles (žinoma, šį minimalios ir maksimalios r reikšmės intervalą reikia nustatyti eksperimentiškai), o tarpinius r dydžius – tarpinės paletės spalvos.

3.3 Rekurentinių diagramų interpretavimas. Pavyzdžiai



28 pav.: Iš logistinio atvaizdžio (žr. 1.2.2 skyrelį) $x_{n+1} = 4x_n(1 - x_n)$, $x_0 = 0.51$, gautas chaotinis signalas (ilustracija kairėje) ir jo rekurentinės diagramos su $r = 0.1$ (ilustracija centre) ir su $r = 0.5$ (ilustracija dešinėje). Signalų imčių skaičius $N = 400$. Iliustracijoje centre matomos trumpos įstrižaininės atkarpos, liudija apie trumpus kvazi-periodiškumo fragmentus signale. Nors vizualiai pats signalas atrodo panašus į atsitiktinį triukšmą, tačiau rekurentinėje diagramoje išryškėja tam tikros struktūros, skirtingai nuo 26 pav.

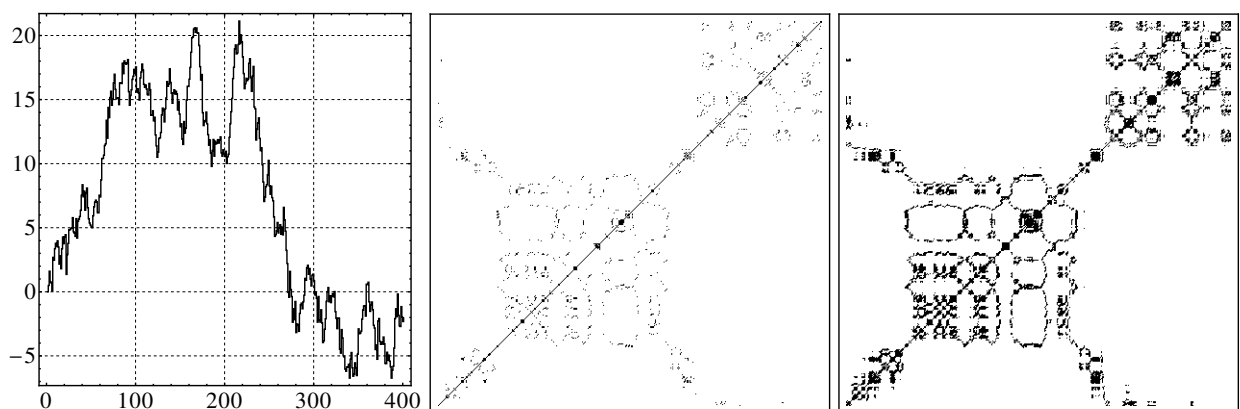


29 pav.: Signalas su trendu $y_n = x_n + 0.01n$ (ilustracija kairėje), čia x_n yra 28 pav. nagrinėtas signalas, ir jo rekurentinės diagramos su $r = 0.1$ (ilustracija centre) ir su $r = 0.5$ (ilustracija dešinėje). Signalų imčių skaičius $N = 400$. Tai, kad rekurentinės diagramos kraštai “išbąla” (juose nebelieka juodų taškų) liudija apie signalo trendą, tuo pačiu ir apie nestacionarumą (nes juodi taškai nėra pasiskirstę homogeniškai).

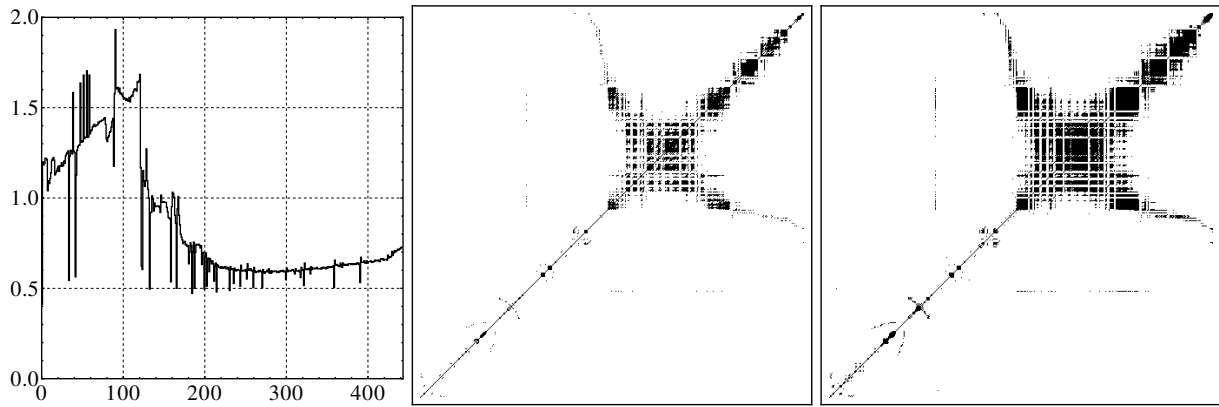
Rekurentinės diagramos gali ne tik nustatyti kvazi-periodiškumus signaluose, bet ir išryškinti daug kitų signalų savybių. Tam tikros juodų taškų struktūros (dariniai) rekurentinės diagramos paveikslėlyje liudija apie specifinius tiriamo signalo bruožus. Išvardinsime dažniausiai sutinkamas struktūras bei pateiksime jų interpretavimą signalo savybių terminais, iliustruodami pavyzdžiais.

- Įstrižaininės linijos sudarytos iš juodų taškų, tiek lygiagrečios pagrindinei rekurentinės diagramos diagonalei, tiek ir statmenos jai, taip pat šachmatinės struktūros įstrižainių kryptimi: signalas turi *periodų* arba *kvazi-periodų*. Periodinio arba kvazi-periodinio elgesio trukmę apsprendžia įstrižaininių linijų ilgiai, kuo jos ilgesnės tuo ilgiau išsilaiko signale periodiškumas. Atstumas (laiko skalėje) tarp dviejų įstrižaininių linijų paveikslėlyje lygus atitinkamo periodo ilgiui signale. Jeigu egzistuoja ilgos įstrižaininės linijos su skirtingais atstumais tarp jų, vadinasi sprendinys turi keletą kvazi-periodų. Pavyzdžiai: 22 pav., 25 pav., 27 pav.
- Juodi taškai išsidėstę daugmaž tolygiai (homogeniškai) per visą rekurentinės diagramos plotą, nors gali būti ir įstrižaininių linijų (liudijančių apie signalo kvazi-periodiškumą): tiriamas signalas *stacionarus*. Pavyzdžiai: 25 pav., 26 pav., 27 pav., 28 pav., 33 pav.
- Jeigu juodų taškų išsidėstymas homogeniškas, bet be pastebimos tvarkos (nereguliarus): signalo reikšmėse vyrauja atsitiktinis *triukšmas*. Tai yra, signalo reikšmių svyravimuose stebimas atsitiktinumas kai signalo reikšmės tampa nepriklausomomis nuo reikšmių artimoje praeityje. Jeigu rekurentinėje diagramoje yra tiek reguliarių, tiek nereguliarių sričių, vadinasi signalas atskiruose laiko intervaluose pasižymi analogiška elgsena. Pavyzdys: 26 pav.
- Juodi taškai pasiskirstę nehomogeniškai: tiriamas signalas *nestacionarus*. Kuo netolygesnis taškų išsibastymas, tuo nestacionaresnis signalas. Pavyzdžiai: 29 pav., 30 pav., 31 pav., 32 pav.
- Du simetriniai rekurentinės diagramos kampai “išbalę” (juose nėra juodų taškų): signalas turi *trendą* arba keletą trendų. Taip pat, kadangi yra trendas, signalas yra *nestacionarus*. Pavyzdžiai: 29 pav., 30 pav., 31 pav., 32 pav.
- Vertikalios ir horizontalios juodų taškų linijos ir telkiniai: signalo būseną tam tikrais laiko tarpais beveik arba visiškai *nekinta*. kuo ilgesnės šios linijos, tuo platesni signalo nusistovėjimo laikotarpiai. Pavyzdžiai: 30 pav., 31 pav., 32 pav.

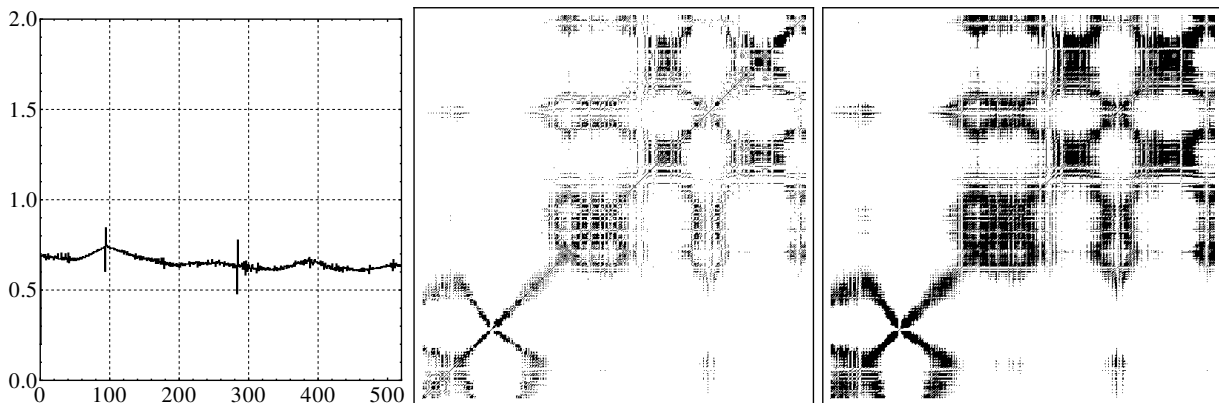
Daugiau sudėtingesnių rekurentinių diagramų interpretacijų pateikta [12].



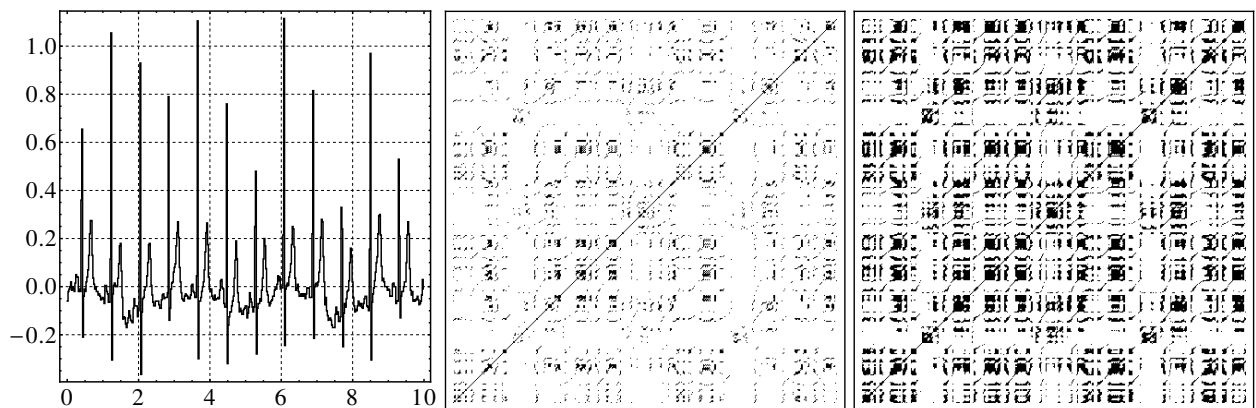
30 pav.: Brauno judėjimo (atsitiktinai klaidžiojančios dalelės) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.5$ (ilustracija centre) ir su $r = 1$ (ilustracija dešinėje). Signalio imčių skaičius $N = 400$. Juodų taškų nehomogeniškumas rodo signalo nestacionarumą.



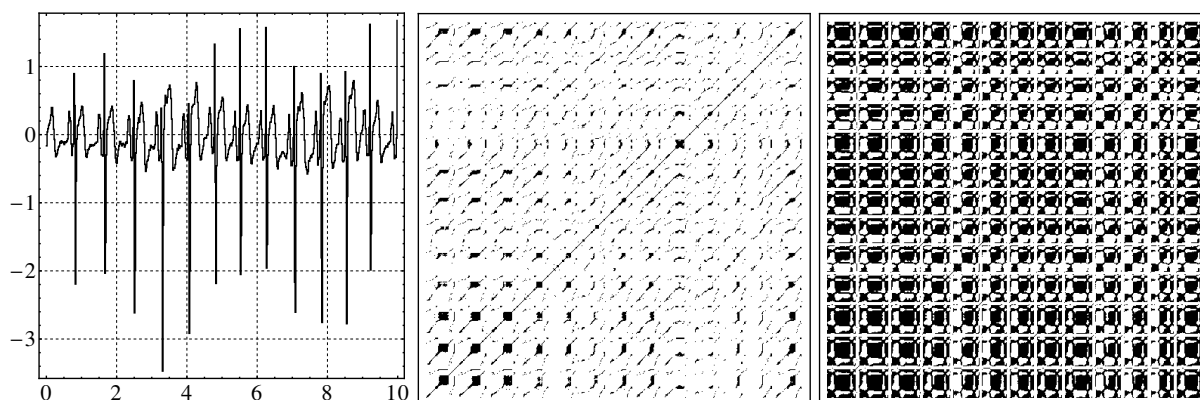
31 pav.: Iš elektrokardiogramos gautų RR intervalų (žr. 21 pav., grafiką dešinėje) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.01$ (ilustracija centre) ir su $r = 0.02$ (ilustracija dešinėje). Signalo imčių skaičius $N = 442$. Juodų taškų nehomogeniškumas rodo signalo nestacionarumą. Taip pat ryškėja vertikalios ir horizontalios juodos linijos, rodančios, kad signale yra lėtai kintančių būsenų.



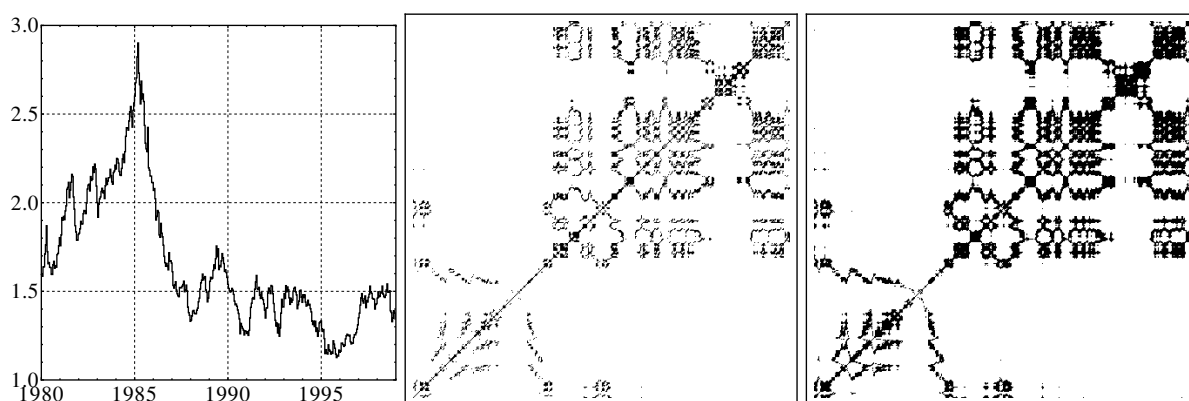
32 pav.: Iš kitos elektrokardiogramos (palyginus su 31 pav.) gautų RR intervalų signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.01$ (ilustracija centre) ir su $r = 0.02$ (ilustracija dešinėje). Signalo imčių skaičius $N = 519$.



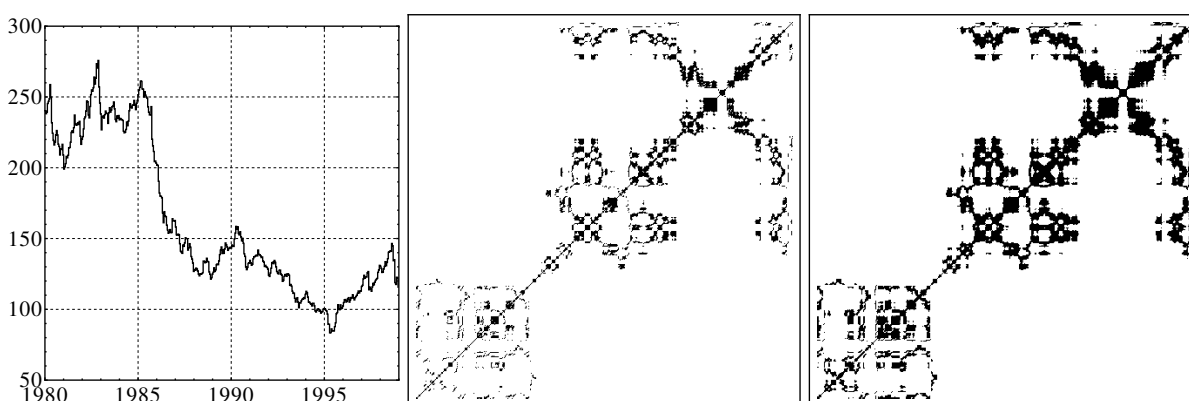
33 pav.: Elektrokardiogramos V4 kanalo fragmento (žr. 21 pav., grafiką kairėje) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.02$ (ilustracija centre) ir su $r = 0.05$ (ilustracija dešinėje). Signalo imčių skaičius buvo dirbtinai sumažintas (atlikus “downsampling”) iki $N = 416$.



34 pav.: Vieno iš encefalogramos kanalų (žr. 1 pav., grafiką dešinėje) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.1$ (ilustracija centre) ir su $r = 0.5$ (ilustracija dešinėje). Signalo imčių skaičius buvo dirbtinai sumažintas (atlikus “downsampling”) iki $N = 499$.



35 pav.: JAV dolerių keitimo kurso į Šveicarijos frankus (žr. 15 pav.) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 0.05$ (ilustracija centre) ir su $r = 0.1$ (ilustracija dešinėje). Signalo imčių skaičius buvo dirbtinai sumažintas (atlikus “downsampling”) iki $N = 433$.



36 pav.: JAV dolerių keitimo kurso į Japonijos jenas (žr. 8 pav., grafiką kairėje) signalas (ilustracija kairėje), ir jo rekurentinės diagramos su $r = 5$ (ilustracija centre) ir su $r = 10$ (ilustracija dešinėje). Signalo imčių skaičius buvo dirbtinai sumažintas (atlikus “downsampling”) iki $N = 433$. Atkreipkime dėmesį, kad dėl signalo reikšmių mastelio atitinkamai turėjo būti padidintos parametro r reikšmės.

4 Koreliacinis integralas ir koreliacinė dimensija

4.1 Laisvės laipsnių skaičiaus sąvoka

Ne tik pats rekurentinės diagramos (žr. 3 skyrių) paveikslėlis gali suteikti svarbios informacijos apie apdorojamą signalą. Pasirodo, įdomi informacija taip pat gaunama nagrinėjant juodų taškų (pikselių) tankį rekurentinėje diagramoje, tiksliau – tendenciją, kaip greitai tankis nyksta, mažinant slenkstinio atstumo parametą r .

Prisiminkime, kad juodi taškai atitinka diskretaus laiko indeksus i ir j , kada signalo būsenos (14) beveik (su toleruojama paklaida $r > 0$) atsikartoja (žr. kriterijų (12)). Kaip atsimename iš rekurentinių diagramų savybių (žr. 3.2.3 skyrelį) bei pateiktų pavyzdžių konkretiems signalams, paveikslėliai “pajuoduoja” padidinus r ir atvirkščiai – “pabąla” sumažinus r . Taigi, mažinant r , juodų taškų kiekis mažėja. Mus domins šio mažėjimo greitis.

Kodėl tai įdomu ir svarbu iš taikomosios pusės? Bet kuris eksperimentiškai registruotas signalas yra tam tikro sudėtingo proceso vyksmo protokolas. Iš dalies, kaip jau buvo rašyta (žr. 3.1 ir 3.3 skyrelius), apie šio (signalų kilmės) proceso sudėtingumą (kvazi-periodiškumus, stacionarumą, trendus ir t.t.) galime spręsti vizualiai nagrinėdami rekurentinę diagramą. Dabar mus papildomai domins konkretus signalo šaltinį charakterizuojantis dydis: *laisvės laipsnių skaičius* (angliškai “degree of freedom”), taip pat vadinamas *dimensija*.

Paaikškinsime šį terminą. Prisiminkime pavyzdį apie švytuoklę, pateiktą skyrelyje 3.2.1. Kaip nustatėme, švytuoklės svyravimą nulemia du (ir ne mažiau) skaičiai: jos atsilenkimo kampas φ ir impulsas (kampo kitimo greitis) φ' pradinio laiko momentu. Kitaip sakant, jeigu norime kontroliuoti švytuoklę, turime mokėti valdyti abu parametrus. Jeigu valdysime tik vieną iš jų, o kitą valdys kažkas kitas, švytuoklė bus nekontroliuojama nei mūsų, nei to kito asmens. Taip yra todėl, kad švytuoklė turi 2 laisvės laipsnius, kitaip tariant jos elgsena yra pilnai charakterizuojama grafiku (trajektorija) dvimatėje (dimensijos 2) erdvėje, kurios koordinatės yra φ ir φ' . Taip pat šis skaičius 2 yra susijęs su faktu, kad matematinis švytuoklės modelis yra 2-ojo laipsnio diferencialinė lygtis

$$\varphi'' + \frac{g}{L} \sin \varphi = 0,$$

čia $g \approx 9.807 \text{ m/s}^2$ yra laisvojo kritimo pagreitis, o L – švytuoklės ilgis.

Kiek laisvės laipsnių turi važiavimo dviračiu procesas? Atrodytų, dviratis rieda žemės paviršiumi (jeigu neskaitysime pakilimų į orą, įsibėgėjus nuo tramplino), kuris yra dvimatis objektas. Tačiau, ar pakanka kontroliuoti tik du kažkokius faktorius tam, kad valdyti dviratį? Tarkime, kad dviratis pats paprasčiausias, be pavarų. Suskaičiuokime kiek galime nepriklausomų judesių daryti, nuo kiekvieno iš kurių priklausys dviračio riedėjimo trajektorija: pedalų mynimas (nors pedalai du, bet kartu su švaistikliais sukasi abu kartu, taigi skaičiuojame kaip vieną judesį), vairo sukiojimas, stabdžių rankenėlės spaudymas (daugumoje dviračių yra du nepriklausomi stabdžiai priekiniam ir užpakaliniam ratui, bet dažniausiai naudojamės tik vienu, taigi skaičiuosime kaip vieną judesį), kūno judinimas, perkeliame masės centrą (kūno judėjimo vertikalia kryptimi, pakylant nuo balnelio, dėl paprastumo neįskaitysime; taigi galima pasilenkti dviem nepriklausomomis kryptimis: į šonus ir priekin-atgal, todėl priskaičiuojame du nepriklausomus judesius). Iš viso gauname penkis nepriklausomus dviratininko kontroliuojamus judesius. Kaip matėme iš pavyzdžio su švytuokle, kiekvienas judesys savo ruožtu turi 2 laisvės laipsnius, taigi šitaip skaičiuodami įvertiname, kad dviračio valdymas yra 10 laisvės laipsnių turintis procesas.

Vadinasi, mūsų matoma proceso dalis – švytuoklės svyravimas į šonus (vienmatėje erdvėje) arba dviratininko judėjimas žemės paviršiumi (dvimatėje erdvėje) tėra projekcija iš nepriklausomų

procesą įtakančių faktorių erdvės (kurios dimensija lygi proceso laisvės laipsnių skaičiui) į nedidelės dimensijos (1 arba 2) erdvę, kurią sugebame suvokti rega. Aukštesnės nei 3 dimensijos erdves mums vizualiai sunku įsivaizduoti, tačiau tai nereiškia, kad visi matomi procesai teturi 3 laisvės laipsnius.

Tuo tarpu vienmačiai (matematiškai priklausantys nuo vieno kintamojo t) signalai tėra procesų projekcijos į vienmatę erdvę. Mūsų tikslas – iš turimų duomenų pabandyti įvertinti tikrąją registruoto proceso dimensiją.

Svarbu pasakyti, kad teoriškai nėra paprasta tiksliai apskaičiuoti netgi santykinai nesudėtingų procesų realią dimensiją. Reikėtų išskirti visus procesą įtakančius veiksniai (faktorius), paliekant tik tarpusavyje nepriklausomus (nekoreliuotus) iš jų. Tačiau, pavyzdžiui, dviračio atveju galima ginčytis: ar pedalų mynimas ir stabdžių naudojimas yra tarpusavyje nepriklausomi veiksniai? Formaliai žiūrint, galime vienu metu spausti tiek pedalus, tiek stabdžius ir abu faktoriai atskirai susisumuos bendrai įtakodami dviratininko judėjimą. Tačiau praktiškai taip elgiamės labai retai – jeigu jau stabdome, paprastai nepamirštame nebeminti dviračio! Taip samprotaudami, prieiname išvados, kad, tarkime, vairo sukiojimas ir pedalų mynimas yra kur labiau nesusiję tarpusavyje faktoriai, nei pedalų mynimas ir dviračio stabdymas – kurie nors ir nėra visiškai tarpusavyje priklausomi (koreliuoti) veiksniai, bet ir nepriklausomi tik iš dalies.

Taigi, analizuodami realius procesus, turime suskaičiuoti juos įtakančius veiksniai, atsižvelgdami į tai, kad kai kurių iš jų tarpusavio nepriklausomumas yra tik dalinis – skaičiuojant išreiškiamas ne pilnu vienetu, bet trupmenine vieneto dalimi. Gaunama skaičių suma vadinama proceso *koreliacine dimensija* (angliškai “correlation dimension”), kuri, kaip matome gali būti ne tik natūrinis, bet ir *trupmeniniu* skaičiumi.

Beje, trupmeninės dimensijos taip pat pasirodo tiriant fraktalus. Pavyzdžiui, Sierpinskio trikampio fraktalinė dimensija (angliškai terminai “fractal dimension”, “Hausdorff dimension”) apytiksliai lygi 1.585, kas reiškia, kad šis darinys nėra nei kreivė (1-dimensinis objektas) nei paviršius (2-dimensinis objektas), o kažkas tarp jų (susiraitęs labiau už bet kurią kreivę ir tarsi pretenduoja užimti plokštumos dalį, vis dėlto per daug kiauras, kad taptų paviršiaus fragmentu). Susipažinus su fraktalų teorija paaiškėja, kad mūsų tradicinis supratimas, jog dimensija būtinai turi būti natūrinis skaičius iš esmės yra inertinio mąstymo pasekmė. Objektai ir reiškiniai su fraktalinėmis (atskiras objekto fragmentas panašus į jį visą) savybėmis sutinkami ir mūsų aplinkoje (debesys, medžiai, sniegas, pakrantės linija, kalnai, vandens ir oro sūkuriai, galaktikos, gyvi organizmai, kurių bet kuriame fragmente yra DNR su genais – informacija apie patį organizmą), o ne tik kaip matematinės abstrakcijos ar įstabūs kompiuterinės grafikos kūriniai. Fraktališkumas, atsiradęs dėl daugelio įtakančių faktorių netiesinės tarpusavio sąveikos, aptinkamas ir daugybėje netrivialios prigimties signalų (medicininės, finansinės, meteorologinės kilmės, transporto kamščiai ir t. t.)

Verta pridurti, kad sudėtingų sistemų ir chaotinių reiškinų matematinėje teorijoje parodoma, kad fraktalinė dimensija ir koreliacinė dimensija yra ekvivalenčios sąvokos.

Liko neatsakytas vienas klausimas: kokį ryšį turi signalo kilmės proceso dimensija su juodų taškų tankio rekurentinėje diagramoje kitimo greičiu, priklausomai nuo r ? Intuityvus atsakymas paremtas paprasta geometrija: prie taško, esančio aukštesnės dimensijos erdvėje yra daugiau priėjimo variantų. Pavyzdžiui, prie taško esančio ant 1-dimensinės tiesės galima priartėti tik dviem būdais – iš kairės arba iš dešinės. Jeigu tas pats taškas atsiduria 2-dimensinėje plokštumoje, jo kaimynu galima tapti ne tik ateinant iš šonų, bet ir iš viršaus bei apačios, taipogi daugeliu įstrižų krypčių. Taigi, didinant taško aplinkos spindulį r , patenkančių į šią aplinką taškų kiekis turėtų didėti tuo sparčiau, kuo aukštesnė proceso dimensija. Vadinasi, jeigu pats signalą pagimdęs procesas bus aukštesnės dimensijos – signalo būsenų, nutolusių mažiau nei per r viena nuo kitos, porų skaičius augs greičiau (didinant r). Kitais žodžiais, juodų taškų rekurentinėje diagramoje skaičiaus (o tuo pačiu ir tankio) kitimo greitis, didinant

(arba mažinant) r , susijęs su signalo kilmės proceso dimensija.

Signalų koreliacinės dimensijos tyrimai reikšmingi analizuojant elektrokardiogramas, encefalogramas (pabandykite paieškoti informacijos internete pagal “ECG correlation dimension”, “EEG correlation dimension”) ir daugelį kitų medicininių, finansinių, meteorologinių (ir ne tik) signalų.

4.2 Signalų koreliacinis integralas

Taip pat kaip 3 skyriuje, pradiniai duomenys yra diskretizuotos signalo reikšmės

$$f_0, f_1, \dots, f_N.$$

Pasirinkę tam tikras parametrų (natūrinių skaičių) $D \geq 2$, $d \geq 1$ reikšmes (pradžioje galima imti $D = 2$, $d = 1$), apibrėžiame signalo būsenas (vektorius y_i) pagal formulę (14):

$$y_i = (f_i, f_{i+d}, f_{i+2d}, \dots, f_{i+(D-1)d}), \quad i = 0, 1, \dots, M, \quad M = N - (D - 1)d.$$

Juodų taškų (pikselių) tankiu rekurentinėje diagramoje (kuri kaip pamename yra paveikslėlis, sudarytas iš juodų ir baltų taškų), apibrėžtoje kriterijum (12), laikomas juodų taškų skaičiaus ir bendro paveikslėlio taškų skaičiaus (kuris lygus $(M+1)^2$) santykis.

Tačiau skaičiuodami juodų taškų tankį neįskaitysime trivialių juodų taškų, esančių pagrindinėje rekurentinės diagramos įstrižainėje (t. y. taškų gaunamų kai $i = j$), žr. 3.2.3 skyrelį. Tokių taškų yra $M+1$. Vadinasi, likęs bendrasis taškų kiekis paveikslėlyje yra

$$(M + 1)^2 - (M + 1) = M(M + 1).$$

Programuodami juodų pikselių kiekio skaičiavimą (su fiksuotu slenkstiniu parametru $r > 0$) tiesiog suskaičiuojame kiek kartų tenkinama nelygybė (12):

$$\|y_i - y_j\| \leq r, \quad i, j = 0, 1, \dots, M, \quad i \neq j.$$

Gautą skaitliuko reikšmę padalinę iš bendro pikselių skaičiaus $M(M + 1)$, gauname juodų pikselių tankio reikšmę. Ši reikšmė taip pat vadinama signalo *koreliaciniu integralu* (angliškas terminas “correlation integral”) ir žymima $C(r)$ (pabrėžiant, kad tai yra funkcija, priklausanti nuo argumento r).

Matematiškai koreliacinis integralas yra apibrėžiamas panaudojant Hevisaido funkciją (angliškai “Heaviside step function” arba “unit step function”) $H(x)$:

$$H(x) = \begin{cases} 0, & x < 0, \\ 1, & x \geq 0. \end{cases}$$

Ji (kartu su sumavimo ženklu \sum) yra patogi formalia matematine kalba užrašyti algoritmuose naudojamus skaitliukus, kurių reikšmės didinamos patenkinus tam tikrą sąlygą.

Taigi, signalo koreliacinis integralas gali būti išreiškiamas formule

$$C(r) = \frac{1}{M(M + 1)} \sum_{\substack{i,j=0 \\ i \neq j}}^M H(r - \|y_i - y_j\|). \quad (15)$$

Pastaba. Griežtai teoriškai koreliaciniu integralu vadinama išraiškos (15) riba, kai signalo būsenų skaičius neapbrėžtai auga:

$$\lim_{M \rightarrow \infty} C(r),$$

o pats dydis (15) laikomas tik koreliacinio integralo įverčiu. Tačiau kompiuteriniuose skaičiavimuose apsiribojama vien tik įverčiu. Todėl, paprastumo tikslu mes vadinsime koreliaciniu integralu išraišką (15).

4.2.1 Koreliacinio integralo savybės

Remiantis tuo, kad signalo koreliacinis integralas $C(r)$, apibrėžtas formule (15), yra rekurentinės diagramos juodų taškų (išmetus pagrindinę įstrižainę) tankis ir prisiminus rekurentinės diagramos savybes (žr. 3.2.3 skyrelį), konstatuojame šias savybes:

- Kadangi mažinant r juodų taškų rekurentinėje diagramoje kiekis mažėja ir kai $r \rightarrow 0$, juodi taškai lieka tik pagrindinėje įstrižainėje, tai

$$C(r) \rightarrow 0, \quad r \rightarrow 0. \quad (16)$$

Kaip jau rašėme 3.2.3 skyrelyje, išimtis įmanoma tik idealiai periodiniam signalui. Teoriškai tokiam signalui net ir su $r = 0$ rekurentinėje diagramoje išlieka juodos įstrižainės nutolusios nuo pagrindinės per periodo ilgį. Taigi, periodiniam signalui įmanoma, kad $C(r) \rightarrow \text{const} > 0$, kai $r \rightarrow 0$. Ši išimtis iliustruota toliau pateiktame grafike 37 pav.

- Kadangi didinant r juodų taškų rekurentinėje diagramoje kiekis nemažėja ir viršijus tam tikrą skaičių (kurį galima įvertinti per signalo reikšmių maksimalią amplitudę), visi rekurentinės diagramos pikseliai tampa juodais, tai

$$C(r) \rightarrow 1, \quad r \rightarrow \infty. \quad (17)$$

Realiai vienetas pasiekiamas esant baigtiniam pakankamai dideliu r (žr. toliau pateikiamus 37 pav., 39 pav.).

- Iš pastarosios savybės taip pat išplaukia, kad $C(r)$ yra *nemažėjanti* funkcija.

4.2.2 Koreliacinio integralo skaičiavimo optimizavimas

Kaip matysime toliau, signalo koreliacinio integralo reikšmės naudojamos koreliacinės dimensijos įverčio skaičiavimuose. Kadangi tenka skaičiuoti funkcijos $C(r)$ reikšmes prie daug įvairių r verčių, o M gali būti pakankamai didelis, skaičiavimai gali pareikalaus nemažai kompiuterio procesoriaus laiko.

Pasinaudojant savybe $\|y_i - y_j\| = \|y_j - y_i\|$ (rekurentinės diagramos simetrija), nesunku dvigubai sumažinti aritmetinių operacijų kiekį koreliacinio integralo skaičiavime:

$$C(r) = \frac{2}{M(M+1)} \sum_{i=0}^M \sum_{j=i+1}^M H(r - \|y_i - y_j\|). \quad (18)$$

Norint išlošti dar daugiau greičio $C(r)$ skaičiavime, galima paaukoti šiek tiek skaičiavimo tikslumo. Idėja tame, kad vietoje tikslaus juodų pikselių tankio rekurentinėje diagramoje skaičiavimo, galime įvertinti tą tankį pagal atsitiktinai parinktus paveikslėlio taškus. Jeigu parinksime sąlyginai ženklų pikselių procentą (tarkime, 10% visų taškų, kai M pakankamai didelis), statistiškai galime tikėtis, kad juodų taškų tankis (dalis) tarp parinktųjų bus apytiksliai lygus tikrajam tankiui, paskaičiuotam patikrinus visus rekurentinės diagramos pikselius.

Tarkime, nusprendėme pagreitinti koreliacinio integralo skaičiavimo algoritmą $opt = 10$ kartų (su konkrečia opt reikšme reikia paeksperimentuoti, lyginant gaunamą $C(r)$ reikšmę su tikrąja). Realizuodami šią idėją, vietoje sumavimo pagal $M(M + 1)/2$ taškų, sumuosime tik pagal

$$M_{opt} = \text{round} \left(\frac{M(M + 1)}{2 \cdot opt} \right),$$

taškų, čia “round” reiškia slankaus kablelio skaičiaus suapvalinimą iki sveikojo.

Atsitiktinai (naudojame atsitiktinių skaičių generatorių) renkamės *nelygias tarpusavyje* (nesikartojančias) indeksų poras

$$(i_k, j_k), \quad \text{tenkinančias sąlygas} \quad 0 \leq i_k \leq M, \quad i_k + 1 \leq j_k \leq M,$$

kol prirenkame skaičių M_{opt} indeksų nesikartojančių porų. Jeigu ateityje teks skaičiuoti funkcijos $C(r)$ reikšmes daug kartų, su įvairiais r – taupant skaičiavimų laiką galima šį atsitiktinių indeksų porų (i_k, j_k) , $k = 1, 2, \dots, M_{opt}$ generavimą vykdyti tik vieną kartą, pirmojo $C(r)$ skaičiavimo metu.

Tuomet apytikslė koreliacinio integralo reikšmė skaičiuojama taip:

$$C(r) \approx \frac{1}{M_{opt}} \sum_{k=1}^{M_{opt}} H(r - \|y_{i_k} - y_{j_k}\|). \quad (19)$$

Pateiktos formulės (15), (18) ar (19) naudoja Hevisaido funkciją tik dėl lakoniškos matematinės užrašymo formos. Praktiškai, programuojant koreliacinio integralo skaičiavimo algoritmą pagal bet kurią iš šių formulių, žymiai racionaliau naudoti tiesiogiai sąlygos (12) patikrinimo sakinius bei skaitliuką.

Taip pat verta atkreipti dėmesį, kad naudojant Euklidinę metriką (žr. 3.2.2 skyrelį), racionaliau sąlygą (12) programuojant pakeisti jai ekvivalenčia

$$\|y_i - y_j\|^2 \leq r^2,$$

o nepriklausančius nuo r dydžius $\|y_i - y_j\|^2$ (arba $\|y_i - y_j\|$, jeigu naudojama ne Euklidinė metrika) skaičiuoti tik vieną kartą.

4.3 Signalų koreliacinė dimensija

Kaip jau buvo rašyta 4.1 skyrelio pabaigoje, iš koreliacinio integralo $C(r)$ priklausomybės nuo r galima gauti informaciją apie apytikslę analizuojamo signalo *koreliacinės dimensijos*, kurią žymėsime D_c , reikšmę.

Jeigu procesas (kurio vienmatė projekcija yra mūsų turimos signalo reikšmės) iš tiesų gyvena D_c -dimensinėje erdvėje, tai bet kurios proceso būsenos r -spindulio aplinka yra D_c -dimensinis rutulys (jeigu naudojame Euklidinę metriką, žr. 3.2.2 skyrelį).

Rutulio tūris proporcingas r^{D_c} , tuo galime įsitikinti nagrinėdami paprasčiausius atvejus. Kai $D_c = 1$, “rutulys” yra $2r$ ilgio (vienmačiu atveju ilgis yra tūrio sinonimas) atkarpa. Kai $D_c = 2$, turime πr^2 ploto (dvimačiu atveju plotas yra tūrio sinonimas) skritulį. Kai $D_c = 3$, rutulio tūris lygus $\frac{4}{3}\pi r^3$.

Ta pati išvada galioja ir kitoms metrikoms (Manheteno arba maksimumo), tik skirsis proporcingumo konstanta.

Todėl (nepriklausomai nuo metrikos pasirinkimo), proceso būsenų kiekis bėgant laikui patenkančių į bet kurios vienos būsenos r -spindulio aplinką (kitais žodžiais, beveik pasikartojančių) turėtų būti proporcingas r^{D_c} .

Vadinasi, jeigu signalo būsenų (vektorių y_i) dimensija (algoritmo parametras D) nedaug skiriasi nuo tikrosios (bet mums nežinomos) proceso dimensijos D_c , tai pagal formulę (15) arba (18) apskaičiuoto koreliacinio integralo $C(r)$ priklausomybė taip pat bus proporcinga (su kažkokia proporcingumo konstanta β) r^{D_c} :

$$C(r) = \beta r^{D_c},$$

arba

$$\log C(r) = \log \beta + D_c \log r. \quad (20)$$

Kaip nustatyti ar algoritme naudojama D reikšmė artima ieškomai D_c , ir ką daryti priešingu atveju, aptarsime toliau – 4.3.2 skyrelyje.

Iš paskutinės formulės išplaukia teorinis koreliacinės dimensijos apibrėžimas:

$$D_c = \lim_{r \rightarrow 0} \frac{\log C(r)}{\log r}. \quad (21)$$

Riboje (21) galima imti bet kurio pagrindo logaritmus (pavyzdžiui, dešimtainius arba natūrinius, nuo jų pagrindo logaritmų santykis nepriklauso), tačiau būtinai vienodo pagrindo trupmenos vardiklyje ir skaitiklyje.

4.3.1 Praktinės algoritmo detalės

Kaip praktiškai, remiantis ribine priklausomybe (21), apskaičiuoti koreliacinės dimensijos D_c artinį?

Dėl konkretumo, tarkime, kad algoritme naudosis dešimtainio pagrindo logaritmus. Perrašome (20):

$$\log_{10} C(r) = D_c \log_{10} r + \log_{10} \beta.$$

Įvedus pažymėjimus $x = \log_{10} r$, $y = \log_{10} C(r)$, $m = \log_{10} \beta$, tai yra tiesės lygtis

$$y = D_c x + m, \quad (22)$$

o ieškomas dydis D_c yra šios tiesės *krypties koeficientas*.

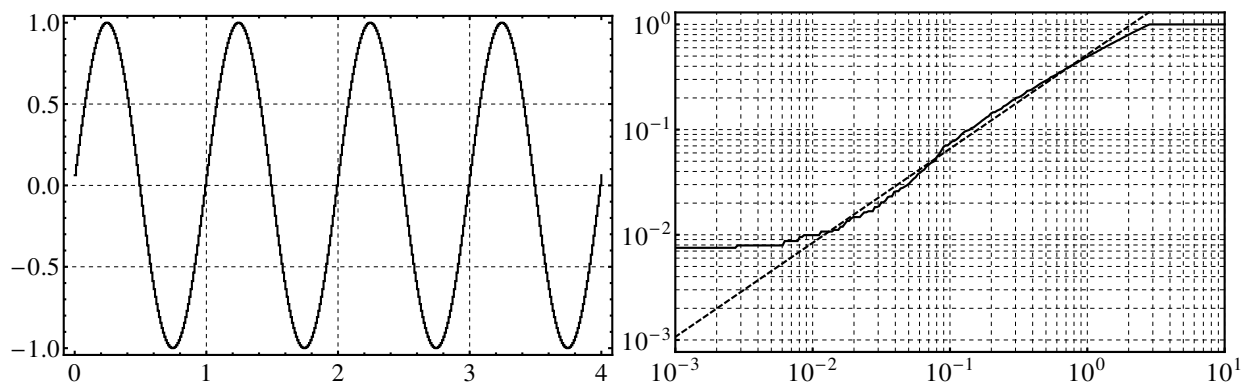
Telieka nusibraižyti funkcijos y grafiką (priklausomybę nuo kintamojo x) ir iš jo įvertinti krypties koeficientą D_c . Kitaip sakant, atsižvelgus į pažymėjimus, turime nubraižyti funkcijos $\log_{10} C(r)$ priklausomybę nuo $\log_{10} r$. Toks grafikas dar vadinamas tiesiog funkcijos $C(r)$ grafiku atžvilgiu r *log-log mastelyje* (skalėje), angliškas terminas “log-log plot” arba “log-log graph”.

Rezultatas pavaizduotas, pavyzdžiui, 37 pav., iliustracijoje dešinėje.

Grafiko braižymas. Argumento r diskretizavimas logaritminiame mastelyje

Programuojant funkcijos grafiko braižymą log-log mastelyje, svarbu suprasti techninį niuansą: pasirinktame argumento r kitimo intervale turi būti tolygiai diskretizuojamos ne r reikšmės (kaip yra įprasta braižant grafikus tiesiniame, o ne logaritminiame mastelyje), bet $\log_{10} r$ (arba kito pagrindo logaritmo) reikšmės. Ta pati pastaba galioja ir tik generuojant funkcijos reikšmes, kurios vėliau bus vizualizuojamos kaip grafikas, naudojant tam skirtus kompiuterinius įrankius.

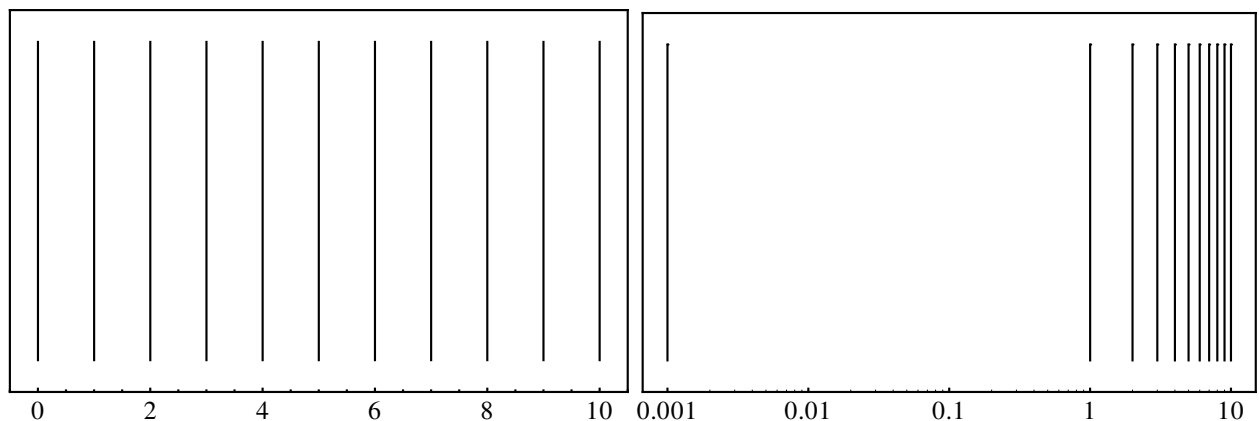
Paaiškinsime šią ir kitas detales konkrečiu pavyzdžiu, pateiktu 37 pav., iliustracijoje dešinėje, ištisinę kreivę. Pirmiausiai pasirenkame argumento r kitimo intervalą: $r_{\min} \leq r \leq r_{\max}$. Skaičiai



37 pav.: Periodinis sinusinės bangos tipo signalas (ilustracija kairėje) ir jo koreliacinio integralo $C(r)$ priklausomybė nuo r log-log mastelyje (ilustracija dešinėje, ištisinė kreivė). Iš priderintos tiesės (ilustracija dešinėje, punktyrinė kreivė) krypties koeficiento įvertinta apytikslė koreliacinės dimensijos reikšmė $D_c = 0.90$ (teoriškai šiam trivialiam signalui turėtume gauti $D_c = 1$). Signalo imčių skaičius $N = 400$. Signalo būsenų dimensija $D = 2$, vėlinimas $d = 1$. Atlikus skaičiavimus su $D = 3$, gaunama artima, bet šiek tikslesnė reikšmė $D_c = 0.94$.

r_{min} ir r_{max} parenkami eksperimentiniu keliu. Ties r_{max} (dešiniajame grafiko šone) funkcijos $C(r)$ reikšmės turėtų įsisotinti ties verte $C(r) = 1$ (remiantis savybe (17)), o r_{min} turėtų būti keleta eilių mažesnis už r_{max} : toks, kad nubraižius $C(r)$ grafiką log-log mastelyje išryškėtų tiesinė priklausomybė – kaip 37 pav. (ilustracija dešinėje, ištisinė kreivė), arba 39 pav. (ilustracija dešinėje, ištisinė kreivė).

Mūsų nagrinėjamu atveju pasirinkome $r_{min} = 10^{-3} = 0.001$, $r_{max} = 10$. Kadangi abscisių ašyje turime $\log_{10} r$, natūralu rinktis r_{min} ir r_{max} kaip kažkokius 10-ties laipsnius.



38 pav.: Intervalo $[0.001, 10]$ padalinimas į 10 vienodo ilgio intervaliukų. Ilustracija kairėje: tiesinis mastelis. Ilustracija dešinėje: logaritminis mastelis. Matyti, kad logaritminiame mastelyje ilgame fragmente tarp 0.001 ir 1 neturėsime diskretizuotų taškų grafiko braižymui.

Dabar jau galime generuoti funkcijos $C(r)$ reikšmes, iš kurių suformuosime grafiką. Tradiciškai elgdamiesi sudalintume intervalą $[r_{min}, r_{max}]$ į pasirinktą kiekį vienodo ilgio intervaliukų ir paskaičiuotume $C(r)$ reikšmes šių intervaliukų kraštuose – tada šias reikšmes perduotumėme grafiką piešančiai paprogramei. Bėda tame, kad perėjus prie logaritminio mastelio beveik visos r reikšmės (prie kurių skaičiuosime $C(r)$) atsidurs dešiniajame intervalo $[r_{min}, r_{max}]$ krašte, žr. 38 pav. Tai gi, 38 pav. atveju logaritminiame mastelyje ilgame argumento r kitimo fragmente $0.001 < r < 1$ neturėsime diskretizuotų r taškų, su kuriais skaičiuotume $C(r)$ – grafikas bus nekokybiškas.

Šios problemos sprendimas – tolygiai (vienodo ilgio intervaliukais) diskretizuoti $\log_{10} r$ (o ne

r) kitimo intervalą $\log_{10} r_{\min} \leq \log_{10} r \leq \log_{10} r_{\max}$. Mūsų atveju $r_{\min} = 10^{-3}$, $r_{\max} = 10$, taigi nagrinėsime intervalą $-3 \leq \log_{10} r \leq 1$.

Pasirenkame grafiko rezoliucijos parametą rez (intervaliukų skaičių), pavyzdžiui $rez = 100$. Kuo didesnė rez reikšmė, tuo glodesnį grafiką gausime, tačiau reikės skaičiuoti atitinkamai daugiau kartų koreliacinį integralą $C(r)$, taigi užtruks ilgiau. Apskaičiuojame kintamojo $\log_{10} r$ diskretizavimo žingsnį (vieno intervaliuko ilgį)

$$h = \frac{\log_{10} r_{\max} - \log_{10} r_{\min}}{rez}.$$

Su mūsų pasirinktomis parametų vertėmis gauname $h = 0.04$.

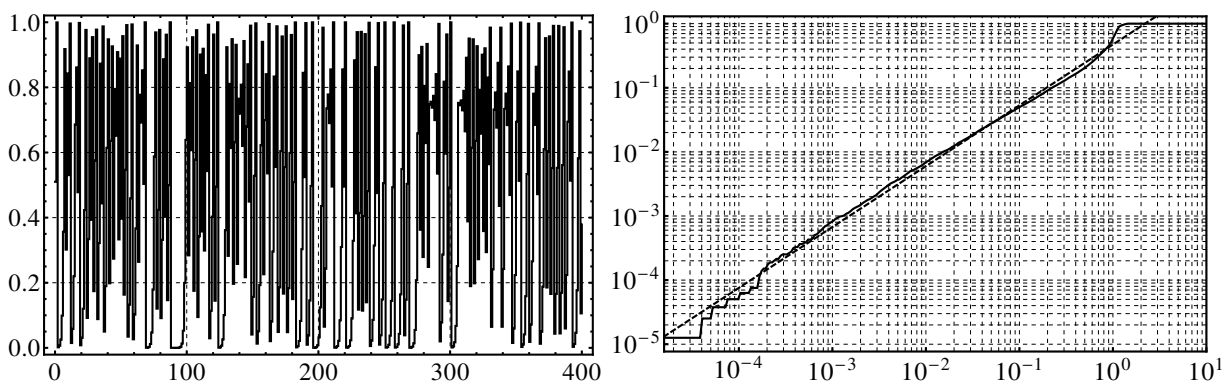
Diskretizuotas kintamojo r reikšmės žymime r_k . Kadangi tolygaus skaidinio žingsnis h buvo paskaičiuotas $\log_{10} r$ mastelyje, turime

$$\log_{10} r_k = \log_{10} r_{\min} + kh, \quad k = 0, 1, \dots, rez.$$

Iš čia išreiškiame

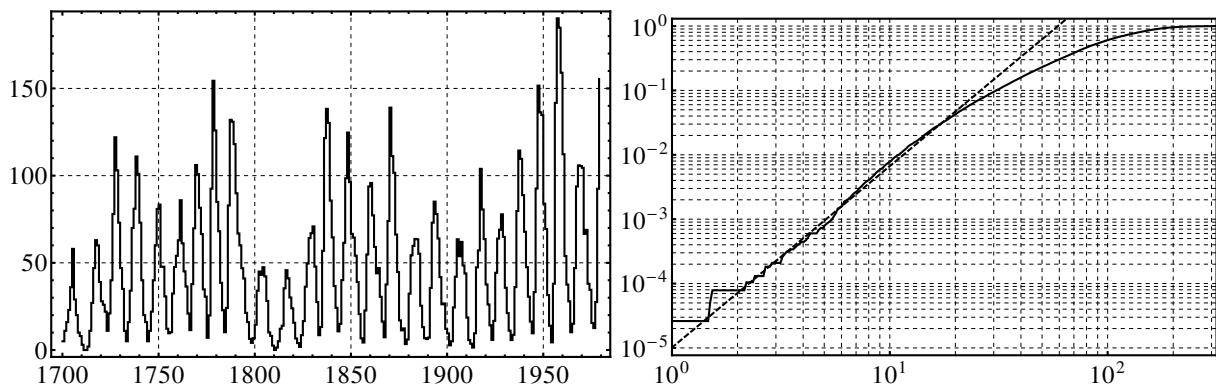
$$r_k = r_{\min} 10^{kh}, \quad k = 0, 1, \dots, rez. \quad (23)$$

Diskrečios reikšmės r_k , paskaičiuotos pagal formulę (23), bus tolygiai išsidėsčiusios logaritminiam mastelyje.



39 pav.: Iš logistinio atvaizdžio (žr. 1.2.2 skyrelį) $x_{n+1} = 4x_n(1 - x_n)$, $x_0 = 0.51$, gautas chaotinis signalas (ilustracija kairėje) ir jo koreliacinio integralo $C(r)$ priklausomybė nuo r log-log mastelyje (ilustracija dešinėje, ištisinė kreivė). Iš priderintos tiesės (ilustracija dešinėje, punktyrinė kreivė) krypties koeficiento įvertinta apytikslė koreliacinės dimensijos reikšmė $D_c = 0.95$. Signalų imčių skaičius $N = 400$. Signalų būsenų dimensija $D = 3$, vėlinimas $d = 1$. Atlikus skaičiavimus su įvairiomis parametro D reikšmėmis gauname: kai $D = 1$, $D_c = 0.78$; kai $D = 2$, $D_c = 0.89$; kai $D = 3$, $D_c = 0.95$; kai $D = 4$, $D_c = 0.97$. Galima pastebėti, kad $D_c \rightarrow 1$.

Galiausiai, apskaičiuojame koreliacinio integralo reikšmes $C(r_k)$, $k = 0, 1, \dots, rez$, ir naudodami jas nubraižome grafiką log-log mastelyje. Naudojant specializuotus kompiuterinius įrankius grafikų braižymui, papildomai skaičiuoti šių reikšmių logaritmus $\log_{10} C(r_k)$ nereikia. Užtenka perduoti įrankiui taškus $(r_k, C(r_k))$, nurodant, kad norime grafiką gauti log-log mastelyje – įrankis pats paskaičiuos logaritmus ir teisingai sužymės abscisių ir ordinačių ašis. Programuojant grafikų braižymą patiems, reikės taip pat pasirūpinti $\log_{10} C(r_k)$ skaičiavimu bei teisingu abscisių ir ordinačių ašių sužymėjimu.



40 pav.: Saulės dėmių skaičiaus per metus signalas (abscisių ašyje atidėti metai), pagal [7], ir jo koreliacinio integralo $C(r)$ priklausomybė nuo r log-log mastelyje (ilustracija dešinėje, ištisinė kreivė). Iš priderintos tiesės (ilustracija dešinėje, punktyrinė kreivė) krypties koeficiento įvertinta apytikslė koreliacinės dimensijos reikšmė $D_c = 2.82$. Signalio imčių skaičius $N = 279$. Signalio būsenų dimensija $D = 4$, vėlinimas $d = 1$. Atlikus skaičiavimus su įvairiomis parametro D reikšmėmis gauname: kai $D = 2$, $D_c = 1.69$; kai $D = 3$, $D_c = 2.28$.

Koreliacinės dimensijos D_c įvertinimas iš $C(r)$ grafiko

Sugeneravę diskrečias reikšmes r_k pagal formulę (23) bei apskaičiavę $C(r_k)$, nubraižome koreliacinio integralo $C(r)$ grafiką log-log mastelyje, intervale $r_{min} \leq r \leq r_{max}$. Kaip jau buvo minėta, intervalo kraštai r_{min} ir r_{max} turi būti parinkti taip, kad dešiniajame grafiko šone funkcijos $C(r)$ reikšmės įsisotintų ties vertę $C(r) = 1$, o intervalo viduryje stebėtume tiesinį grafiko pavidalą, taip kaip 37 pav. (ilustracija dešinėje, ištisinė kreivė), arba 39 pav. (ilustracija dešinėje, ištisinė kreivė).

Išskirsime tą intervalo $r_{min} \leq r \leq r_{max}$ dalį, kurioje grafikas turi tiesinį pavidalą.

Pirmiausiai tarp diskretizuotų reikšmių r_k surandame tokią, ties kuria prasideda kreivės tiesinis augimas. Iš principo tai galima padaryti vizualiu būdu, bet geriau taikyti automatizuotą kriterijų. Pavyzdžiui, pradžioje suskaičiuojame kiek funkcijos $C(r)$ logaritmas užauga per visą intervalą:

$$\Delta = \log_{10} C(r_{max}) - \log_{10} C(r_{min}) = -\log_{10} C(r_{min}).$$

(čia pasinaudojome tuo, kad, remiantis 4.2.1 skyrelyje gautomis koreliacinio integralo savybėmis, funkcija $C(r)$ yra nemažėjanti, be to, turėtų būti $C(r_{max}) = 1$). Tada pasirenkame pirmąją r_k reikšmę, tokią, ties kuria $C(r)$ logaritmas yra jau užaugęs 25% viso prieauglio (empiriškai manome, kad čia grafikas jau turėtų būti įgavęs tiesinį pavidalą):

$$\min r_k : \log_{10} C(r_k) > \log_{10} C(r_{min}) + 0.25\Delta = 0.75 \log_{10} C(r_{min}).$$

Šios r_k reikšmės logaritmą pažymime $x_1 = \log_{10} r_k$, o atitinkamos $C(r_k)$ reikšmės logaritmą pažymime $y_1 = \log_{10} C(r_k)$.

Panašiai pažymime dešiniąjį tiesinio augimo intervalo kraštą. Pavyzdžiui, iš visų r_k reikšmių išrenkame pirmąją tokią, ties kuria $C(r)$ logaritmas yra jau užaugęs 75% viso prieauglio

$$\min r_k : \log_{10} C(r_k) > \log_{10} C(r_{min}) + 0.75\Delta = 0.25 \log_{10} C(r_{min}).$$

Šios r_k reikšmės logaritmą pažymime $x_2 = \log_{10} r_k$, o atitinkamos $C(r_k)$ reikšmės logaritmą pažymime $y_2 = \log_{10} C(r_k)$.

Pagaliam, galime įvertinti tiesės (22) krypties koeficientą – ieškomą koreliacinės dimensijos reikšmę D_c . Tai galima padaryti išskirtoje intervalo dalyje parinkus optimalią tiesę *mažiausių kvadratų* metodu

(angliški terminai “least squares”, “linear fit”), tačiau dažniausiai pakanka paprasto įvertio, paremto tiesės (22) koeficientų D_c ir m radimu iš sąlygų, kad ji eina per taškus (x_1, y_1) ir (x_2, y_2) :

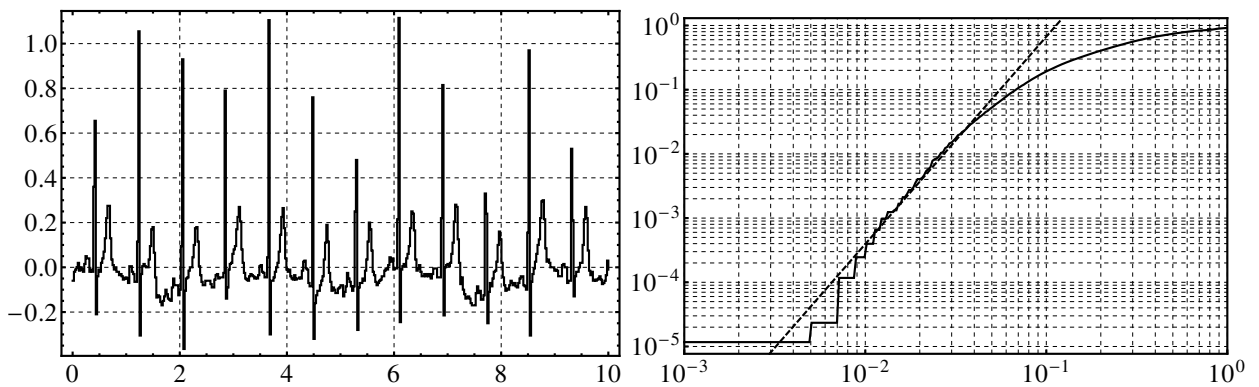
$$D_c = \frac{y_2 - y_1}{x_2 - x_1}, \quad m = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}.$$

Iš esmės mus domino tik D_c reikšmė, tačiau žinoti koeficientą m taip pat naudinga – galima nubraižyti tiesės (22) grafiką kartu su koreliacinio integralo $C(r)$ grafiku log-log skalėje (kaip tai padaryta 37 pav. ir 39 pav., tiesė nubraižyta punktyrine linija) ir vizualiai patikrinti, kad koreliacinę dimensiją D_c apskaičiavome teisingai.

4.3.2 Algoritmo parametrų D ir d pasirinkimas

Nagrinėdami signalo būsenas (vektorius y_i) mes naudojame dimensiją D , kuri greičiausiai *skiriasi* nuo mums nežinomos D_c . Jeigu skaičiavimuose naudojama parametro D vertė ženkliai mažesnė nei tikroji proceso dimensija D_c , skaičiuodami galime gauti D_c įvertį su nemaža paklaida.

Taip pat signalo būsenų apibrėžimui reikalingas vėlinimo parametras d . Remiantis matematine teorija, geriausiu parametro d pasirinkimu yra reikšmė, proporcinga tai, su kuria signalo *autokoreliacijos funkcija* (žr. 5 skyrių) pirmą kartą tampa artima nuliui (jos modulis neviršija $2/\sqrt{N}$).

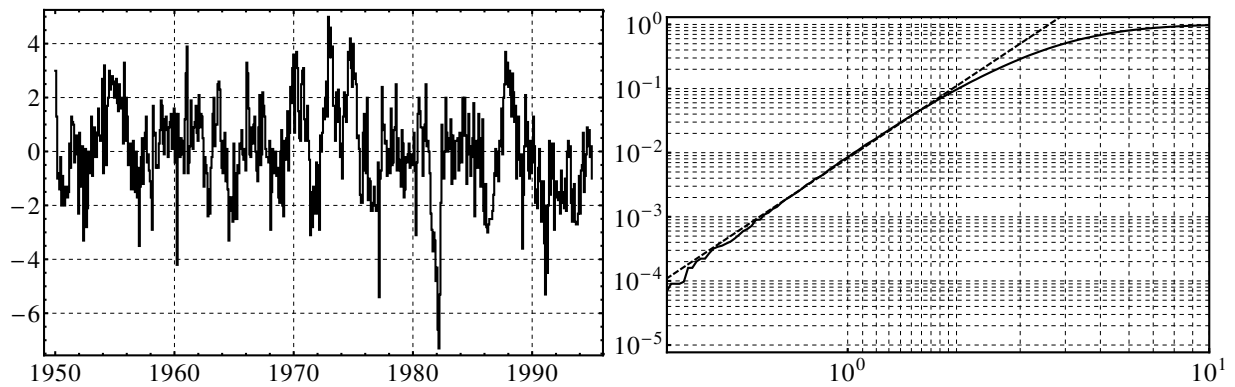


41 pav.: Elektrokardiogramos V4 kanalo fragmento (žr. 21 pav. ir 33 pav.) signalas (ilustracija kairėje), ir jo koreliacinio integralo $C(r)$ priklausomybė nuo r log-log mastelyje (ilustracija dešinėje, ištisinė kreivė). Iš priderintos tiesės (ilustracija dešinėje, punktyrinė kreivė) krypties koeficiento įvertinta apytikslė koreliacinės dimensijos reikšmė $D_c = 3.23$. Signalio imčių skaičius buvo dirbtinai sumažintas (atlikus “downsampling”) iki $N = 416$. Signalio būsenų dimensija $D = 4$, vėlinimas $d = 1$. Gautas D_c įvertis dar nėra nusistovėjęs – auga didinant D . Norint D_c įvertinti patikimiau, reikėtų atlikti daug eksperimentų didinant algoritmo parametrų D ir d reikšmes.

Norint patikimai įvertinti D_c , reikia kartoti koreliacinės dimensijos apskaičiavimo algoritmą, vis didinant metodo parametro D reikšmę. Tęsiame tol, kol algoritmu apskaičiuojama D_c vertė beveik nebesikeis – įsisotins. Gauta D_c vertė ir bus patikimiausias koreliacinės dimensijos artinys.

Sudėtingos kilmės (medicininės, meteorologinės, finansinės, ir kt., žr. pvz. 41 pav., 42 pav.) signalai dažnai pasižymi didelėmis D_c reikšmėmis. Joms patikimai įvertinti reikia atlikti daug eksperimentų pagal šiame skyriuje išdėstytą algoritmą, tolygiai didinant parametro D vertę ir, jeigu optimalios vėlinimo d reikšmės nesame nustatę minėtu signalo autokoreliacijos metodu, taip pat ir d vertę.

Kaip buvo minėta 3.2 skyrelyje, rekurentinės diagramos skaičiavimo algoritmo, tuo pačiu ir koreliacinės dimensijos įvertinimo metodo vykdymo laikas auga kvadratiškai, didinant diskretizuoto signalo imčių skaičių N . Tokiu atveju tenka priimti analogiškus siūlomiems 3.2 skyrelyje sprendimus:



42 pav.: Nuokrypių nuo vidutinio okeano lygio signalas (susijęs su klimato dariniu El Niño), registruotas Ramiojo vandenyno pietuose (abscisių ašyje atidėti metai), pagal [7], ir jo koreliacinio integralo $C(r)$ priklausomybė nuo r log-log mastelyje (ilustracija dešinėje, ištisinė kreivė). Iš priderintos tiesės (ilustracija dešinėje, punktyrinė kreivė) krypties koeficiento įvertinta apytikslė koreliacinės dimensijos reikšmė $D_c = 3.75$. Signalų imčių skaičius $N = 539$. Signalų būsenų dimensija $D = 4$, vėlinimas $d = 1$. Atlikus skaičiavimus su įvairiomis parametro D reikšmėmis gauname: kai $D = 3$, $D_c = 2.82$; kai $D = 5$, $D_c = 4.33$. Gautas D_c įvertis dar nėra nusistovėjęs – auga didinant D . Norint D_c įvertinti patikimiau, reikėtų atlikti daug eksperimentų didinant algoritmo parametrų D ir d reikšmes.

dirbtinai sutrumpinti skaitmeninį signalą, arba suskaidžius jį į fragmentus (kiekvienam fragmentui galėsime atskirai įvertinti koreliacinę dimensiją ir tarpusavyje palyginti gautus rezultatus), arba praretinus (angliškas terminas “downsampling”) skaitmeninio signalo reikšmes, paliekant, pavyzdžiui, tik kas dešimtą.

Beje, jeigu turime įvertinę koreliacinę dimensiją D_c , tai skaičiuojant rekurentines diagramas (apie kurias kalbėjome 3 skyriuje) reikėtų pasirinkti parametrą D kaip artimiausią sveiką skaičių $D \approx D_c$.

5 Signalų koreliacija ir autokoreliacija

Tarptautinis žodis *koreliacija* reiškia nagrinėjamų objektų tarpusavio ryšius, panašumą, priklausomybę vienas nuo kito. Praeitame skyriuje susipažinome su koreliacinės dimensijos sąvoka ir skaičiavimo algoritmu. Kaip paminėjome, koreliacinė dimensija įvertina proceso, iš kurio kilęs signalas, vyksmą įtakojančių tarpusavyje nepriklausomų arba dalinai priklausomų faktorių kiekį.

Šiame skyriuje naudosime terminą “koreliacija” kiek kitu kampu. Būtent, domėsime dviejų signalų tarpusavio priklausomybės (koreliacijos) laipsniu bei vieno signalo reikšmių savitarpio priklausomybe (autokoreliacija) nuo reikšmių praeityje. Kaip matysime, signalų tarpusavio koreliacija ne tik nustato arba paneigia ryši tarp dviejų vyksmų, bet ir įvertina laiko trukmę, per kurią vieno vyksmo efektas pasireiškia kitam. Galimi taikymai būtų, remiantis turimais ilgalaikiais stebėjimais, išsiaiškinti kaip greitai ir kaip ženkliai sumažinti arba padidinti mokesčiai įtakoja ekonominius rodiklius, reklamos apimtis – įmonės apyvartą, ir pan. Arba, pavyzdžiui, naftos kainos inercija – kainos priklausomybė nuo to kokia ji buvo prieš dieną, prieš dvi dienas ir taip toliau; kaip greitai gęsta ši priklausomybė, didinant dienų skaičių į praeitį, t. y. kaip ilgai išlieka sistemoje atmintis? Galimi pavyzdžiai ne tik finansuose, tarkime kardiologus domina kaip ilgai išlieka patirto streso (fizinio arba emocinio) “aidas” elektrokardiogramoje.

Prisiminsime tokias paprasčiausias sąvokas: atsitiktinis dydis, atsitiktinio dydžio vidurkis, dispersija, dviejų atsitiktinių dydžių koreliacijos koeficientas, empiriniai vidurkis, dispersija bei koreliacijos koeficientas.

5.1 Vidurkis, dispersija ir koreliacijos koeficientas

Atsitiktinio dydžio sąvoka

Atsitiktiniu dydžiu vadiname taisyklę (funkciją), pagal kurią atsitiktinumams priskiriame skaičius. Pavyzdžiui, mėtant monetą galimos dvi atsitiktinės baigtys: atsiverčia monetos pusė su skaitmeniu arba su herbu. Pirmajai baigmei priskiriame skaičių 0, antrajai skaičių 1. Taigi, apibrėžėme atsitiktinį dydį, kuris su tikimybėmis $1/2$ ir $1/2$ (simetrinės monetos atveju) įgyja reikšmes 0 ir 1.

Atsitiktinio dydžio sąvoka būtina nagrinėjant atsitiktinumus matematiškai ir rašant kompiuterines programas. Naudodami ją, vietoje tokių sąvokų kaip, pavyzdžiui, “herbas” ir “skaitmuo”, operuojame vien tik skaičiais.

Mes nagrinėsime tik *diskrečius* atsitiktinius dydžius, įgyjančius baigtinį galimų reikšmių kiekį. Pavyzdžiui, su moneta susietas atsitiktinis dydis diskretus (tik dvi galimos reikšmės), bet oro temperatūros vertė nėra diskretus atsitiktinis dydis – formaliai, jeigu matuojame ją kiek norima tiksliai, gausime begalybę (kontinuumą) galimų temperatūros reikšmių. Tačiau eksperimentuose ir programuodami dirbame tik su suapvalintomis signalų reikšmėmis ir baigtiniame jų kitimo intervale. Tarkime, oro temperatūrą registruojame 0.1 laipsnio tikslumu ir tik intervale nuo -50°C iki $+50^\circ\text{C}$. Tokiu atveju temperatūros signalo reikšmės tampa diskrečiu atsitiktiniu dydžiu, galinčiu įgyti tik 1001 galimą reikšmę.

Vidurkio apibrėžimas

Diskretaus atsitiktinio dydžio X , įgyjančio m skirtingų reikšmių $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m$ su atitinkamomis tikimybėmis p_1, p_2, \dots, p_m *vidurkiu* (angliškai “mean”, “expectation”) vadinamas skaičius

$$MX = \tilde{x}_1 p_1 + \tilde{x}_2 p_2 + \dots + \tilde{x}_m p_m = \sum_{k=1}^m \tilde{x}_k p_k. \quad (24)$$

Kaip matome, šis skaičius yra visų X galimų reikšmių vidutinė vertė, vidurkinant su svoriais, lygiais atitinkamoms tikimybėms. MX yra skaičius, apie kurį statistiškai svyruoja X reikšmės.

Vidurkio savybės

Prisiminkime vidurkio operacijos savybes, kurios nesunkiai įrodomos, remiantis vidurkio apibrėžimu (24) ir elementariomis tikimybių savybėmis:

1. Konstantos c vidurkis lygus pačiai konstantai: $Mc = c$.
2. Konstanta c išsikelia prieš vidurkio operaciją: $M(cX) = cMX$.
3. Tarkime, X ir Y yra atsitiktiniai dydžiai. Tuomet $M(X + Y) = MX + MY$.
4. $M(X \cdot Y) = MX \cdot MY$ tada ir tik tada, kai X ir Y yra tarpusavyje nepriklausomi (nekoreliuoti) atsitiktiniai dydžiai.

Dispersijos apibrėžimas

Atsitiktinio dydžio X dispersija (angliškai “variance”) vadinamas skaičius

$$DX = M\left((X - MX)^2\right). \quad (25)$$

Dispersija nusako kiek atsitiktinės reikšmės yra išsibarsčiusios atžvilgiu savo vidutinės vertės. Pavyzdžiui, sverdami daiktą svarstyklėmis, kiekvieno svėrimo metu gausime atsitiktinius skaičius, daugiau ar mažiau nutolusius nuo tikrojo svorio. Bet kokių svarstyklių (brangių elektroninių ar pigaus bezmėno) rodomas svoris bus atsitiktinis dydis, kurio vidurkis lygus tiksliai daikto masei. Tačiau brangus prietaisas pasižymės maža dispersija, o pigus – greičiausiai ženkliai didesne.

Dispersijos savybės

Iš dispersijos apibrėžimo (25) ir vidurkio savybių išplaukia (atlikus elementarius algebrinius pertvarkymus) šios dispersijos savybės:

1. Dispersiją galima skaičiuoti ir pagal formulę $DX = M(X^2) - (MX)^2$.
2. Dispersija visada neneigiama: $DX \geq 0$.
3. Konstantos c dispersija lygi nuliui: $Dc = 0$.
4. Konstanta c išsikelia prieš dispersijos operaciją, pakelta kvadratu: $D(cX) = c^2DX$.
5. Tarkime, X ir Y yra atsitiktiniai dydžiai. Tuomet

$$D(X + Y) = DX + DY + 2M\left((X - MX)(Y - MY)\right).$$

Remiantis 4-ąja vidurkio savybe, bei algebriniais pertvarkymais įrodoma formulė

$$M\left((X - MX)(Y - MY)\right) = M(XY) - MXMY,$$

konstatuojame, kad

$$M\left((X - MX)(Y - MY)\right) = 0 \quad \Leftrightarrow \quad X \text{ ir } Y \text{ tarpusavyje nepriklausomi.} \quad (26)$$

Skaičius $M\left((X - MX)(Y - MY)\right)$ vadinamas atsitiktinių dydžių X ir Y kovariacija (angliškai “covariance”). Taigi, $D(X + Y) = DX + DY$ tada ir tik tada, kai X ir Y yra tarpusavyje nepriklausomi (nekoreliuoti) atsitiktiniai dydžiai.

Dažnai vietoje dispersijos naudojama charakteristika $\sigma(X) = \sqrt{DX}$, vadinama *standartiniu nuokrypiu* (angliškai “standard deviation”). Skirtingai nuo dispersijos, konstanta išsikelia prieš $\sigma(X)$ be kvadrato: $\sigma(cX) = c\sigma(X)$, taigi, ši operacija išlaiko atsitiktinio dydžio dimensiją.

Koreliacijos koeficiento apibrėžimas

Atsitiktinių dydžių X ir Y *koreliacijos koeficientu* (angliškai “correlation coefficient”) vadinamas skaičius

$$\rho(X, Y) = \frac{M((X - MX)(Y - MY))}{\sqrt{DX}\sqrt{DY}} = \frac{M((X - MX)(Y - MY))}{\sqrt{M((X - MX)^2)M((Y - MY)^2)}}. \quad (27)$$

Koreliacijos koeficiento savybės

Koreliacijos koeficiento savybės atskleidžia galimybę naudoti šį dydį duomenų tarpusavio priklausomybės laipsniui nustatyti:

1. Koreliacijos koeficientas simetrinis atsitiktinių dydžių X ir Y atžvilgiu: $\rho(X, Y) = \rho(Y, X)$.
2. Koreliacijos koeficientas lygus nuliui: $\rho(X, Y) = 0$ tada ir tik tada, kai X ir Y yra tarpusavyje nepriklausomi (nekoreliuoti) atsitiktiniai dydžiai. Ši išvada išplaukia iš kovariacijos savybės (26).
3. Tarkime, kad atsitiktiniai dydžiai X ir Y susiję (priklausomi) tarpusavyje *tiesiškai*. Tai yra, $Y = aX + b$ su tam tikromis konstantomis $a \neq 0$ ir b . Ši prielaida reiškia, kad abu dydžiai svyruoja atsitiktinai, bet sinchroniškai (panašiai kaip svyruoja oro temperatūra dvejose šalia esančiose vietovėse). Tuomet

$$\rho(X, Y) = 1, \quad \text{jeigu } a > 0 \quad \text{ir} \quad \rho(X, Y) = -1, \quad \text{jeigu } a < 0.$$

Šis teiginys lengvai įrodomas įsistačius Y išraišką į koreliacijos koeficiento apibrėžimą (27) ir pasinaudojus vidurkio bei dispersijos savybėmis.

4. Bet kokiems atsitiktiniams dydžiams X ir Y

$$-1 \leq \rho(X, Y) \leq 1, \quad \text{kitaip tariant,} \quad |\rho(X, Y)| \leq 1. \quad (28)$$

Savybė įrodoma naudojantis Koši–Švarzo nelygybe (angliškoje literatūroje dar žinoma kaip “Cauchy–Bunyakovsky–Schwarz inequality”).

Remiantis koreliacijos koeficiento savybėmis, skaičiaus $|\rho(X, Y)|$ reikšmės tarp 0 ir 1 parodo statistinio ryšio tarp atsitiktinių dydžių X ir Y laipsnį (stiprumą). Jeigu norima šį laipsnį išreikšti procentais, galima nagrinėti dydį $100|\rho(X, Y)|$.

$|\rho(X, Y)|$ reikšmių intervalas nuo 0 iki 0.2 rodo labai silpną ryšį, nuo 0.2 iki 0.5 – silpną ryšį. Reikšmės tarp 0.5 ir 0.7 byloja apie vidutinį ryšį, o ryšys laikomas stipriu, jeigu gauta reikšmė yra tarp 0.7 ir 1.

Statistinis ryšys ir priežastingumas. Jeigu atsitiktiniai dydžiai tarpusavyje susiję priežastingumo ryšiais, tai atspindės koreliacijos koeficiento reikšmėje. Tačiau būtina pastebėti, kad atvirkštinis teiginys nebūtinai teisingas! Konkretiems atsitiktiniams dydžiams gavę pakankamai didelę jų koreliacijos koeficiento reikšmę, turime teisę teigti tik tiek, kad nustatytas *statistinis ryšys*, o dėl tarpusavio priežastingumo garantijos duoti negalime.

Pateiksime tai iliustruojančius gerai žinomus pavyzdžius. Tirdami ledų pardavimo duomenis ir skenduolių skaičiaus įrašus, gausime aukštą koreliaciją. Vis dėlto, būtų klaidinga teigti, kad prisivalgę ledų žmonės dažniau skęsta. Paprasčiausiai, tiek ledų suvartojimas, tiek besimaudančiųjų skaičius, yra susiję su atšilusiais orais. Tokiu būdu šie procesai tampa statistiškai susiję, nors ir nėra susiję priežastingumu.

Kitas pavyzdys atskleidžia esant įmanomą dar labiau netikėtą variantą: koreliacija gali parodyti ryšį, kai tikrasis priežastingumas būna visiškai atvirkščias. Prieš daugiau nei 100 metų buvo nustatyta, kad sveiki žmonės dažnai turėdavo blusų, tuo tarpu ligoniai – ne. Ar tai reiškia, kad blusos suteikia sveikatą? Iš tiesų yra priešingai: jos pačios dažnai perneša bacilas, bet žmogui susirgus ir pakilus temperatūrai, blusoms tampa per karšta ir jos išnyksta.

Atvejis, kada gaunama koreliacijos koeficiento reikšmė yra klaidinanti, vadinama *klaidingąja koreliacija* (angliškai “spurious correlation”).

Vis dėlto, klaidinga koreliacija pasireiškia gana retai ir tai neturėtų atbaidyti nuo jos taikymo.

Empirinis koreliacijos koeficientas

Tarkime, stebėdami atsitiktinį dydį X (pavyzdžiui, registruodami tam tikro signalo reikšmes), gauname reikšmes

$$x_1, x_2, \dots, x_n.$$

Dalis gautų skaičių gali kartotis, o kai kurios iš teoriškai įmanomų X reikšmių gali iš viso nepasitaikyti.

Atsitiktinio dydžio X vidurkį galime įvertinti statistiškai (empiriškai), remiantis turimais stebėjimo duomenimis:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Skaičių $\bar{x} \approx MX$ vadinsime atsitiktinio dydžio X *empiriniu* vidurkiu (angliškai “sample mean”, “empirical mean”, “arithmetic mean”). Apytikslis vidurkio įvertis tuo tikslesnis, kuo didesnis n (stebėjimo duomenų kiekis). Dažnai empirinis vidurkis taip pat vadinamas *aritmetiniu* vidurkiu.

Aritmetinio vidurkinimo pavojai. Naudojant aritmetinį (empirinį) vidurkį reikia būti itin atidiems, jeigu duomenys susiję su kampais, fazėmis ir panašiai. Pavyzdžiui, tiesiogiai apskaičiavę dviejų kampų $x_1 = 1^\circ$ ir $x_2 = 359^\circ$ aritmetinį vidurkį, gausime nekorektišką įvertį $\bar{x} = 180^\circ$, kai tikrasis vidurkis lygus 0° .

Kadangi dispersija (25) apibrėžiama per vidurkio operaciją, ją taip pat galima empiriškai įvertinti.

Pakeitę išraiškoje (27) vidurkio operacijas empiriniais analogais, gauname formulę *empirinio koreliacijos koeficiento* (dar vadinamo Pirsono koreliacijos koeficientu, angliškai “Pearson’s correlation coefficient”) skaičiavimui:

$$r_{xy} = \frac{\sum_{j=1}^n (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2 \sum_{j=1}^n (y_j - \bar{y})^2}}, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i, \quad (29)$$

čia y_1, y_2, \dots, y_n yra atsitiktinio dydžio Y stebėjimo duomenys. Formulė (29) naudojama norint praktiškai įvertinti (apytiksliai) teorinį koreliacijos koeficientą (27).

5.2 Dviejų signalų tarpusavio koreliacija

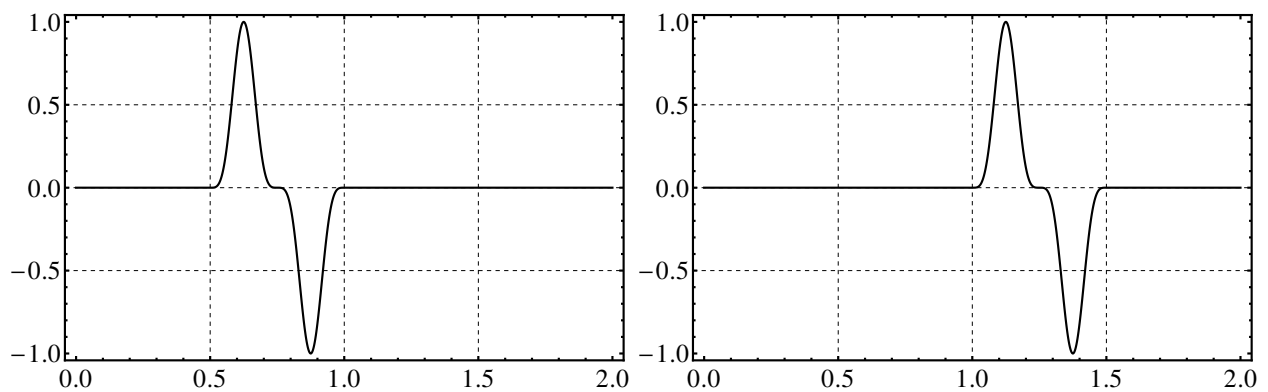
5.2.1 Signalų tarpusavio padėties laike įtaka koreliacijai

Pirmiausiai aptarsime *dviejų signalų tarpusavio koreliaciją* ir jos įvertinimo būdą.

Sakykime, turime signalų $f(t)$ ir $g(t)$, registruotų tą pačią laiko t trukmę ir su tuo pačiu diskretizavimo žingsniu Δt (laiko tarpu, skiriančiu du iš eilės einančius signalo reikšmių matavimus) eksperimentinio stebėjimo duomenis

$$f_0, f_1, \dots, f_N, \quad \text{ir} \quad g_0, g_1, \dots, g_N.$$

Norėtume įvertinti jų tarpusavio koreliaciją, tikėdamiesi identifikuoti esantį ar nesantį panašumą, priklausomybės vienas nuo kito laipsnį.



43 pav.: Iliustracija kairėje: signalas $f(t)$. Iliustracija dešinėje: signalas $g(t)$, kuris kaip matome iš grafiko yra tas pats signalas $f(t)$, tik paslinktas laike.

Nagrinėkime du konkrečius signalus, pavaizduotus 43 pav. Iš tiesų, abiejuose grafikuose pavaizduotas vienas ir tas pats signalas, tik paslinktas laike (pavyzdžiui, ta pati banga, užfiksuota dviem skirtingais laiko momentais).

Apskaičiuokime šių signalų reikšmių empirinį koreliacijos koeficientą, pagal formulę (29). Kaip matyti iš grafikų, abiejų signalų vidutinės reikšmės lygios nuliui, todėl išraiška (29) atrodo taip:

$$r_{fg} = \frac{\sum_{j=0}^N f_j g_j}{\sqrt{\sum_{j=0}^N f_j^2 \sum_{j=0}^N g_j^2}}.$$

Bet mūsų pavyzdyje signalai f ir g tokie, kad ten kur reikšmės $f_j \neq 0$, turime $g_j = 0$ ir atvirkščiai – jeigu $g_j \neq 0$, tai $f_j = 0$. Vadinasi, visada sandauga $f_j g_j = 0$, o tai reiškia, kad šiems (iš tiesų labai panašioms vienas į kitą signalams) koreliacijos koeficientas lygus nuliui: $r_{fg} = 0$.

Tačiau užtenka pastumti laike nagrinėjamų signalų reikšmes taip, kad $f_j = g_j$ (jos sutaptų) ir koreliacija taps lygia vienetui: $r_{fg} = 1$. Stumdami signalus (bet nekeisdami jų formos) galime gauti ir tarpinės koreliacijos koeficiento reikšmes.

Išvada. Dviejų signalų koreliacijos koeficiento reikšmė priklauso ne tik nuo signalų formos, bet ir nuo jų tarpusavio padėties laike.

Todėl, siekiant įvertinti signalų tarpusavio ryšį, vieno koeficiento nepakanka.

5.2.2 Signalų tarpusavio koreliacijos funkcija

Ištirsime kaip keičiasi koreliacijos koeficiento dydis, priklausomai nuo besikeičiančios dviejų signalų f ir g tarpusavio padėties laike. Kadangi nagrinėjame skaitmeninius (diskrečius) signalus, laiko rolę atlieka signalų reikšmių sekų indeksai.

Sveikuoju skaičiumi d žymėsime postūmį diskrečiame laike – per kiek indeksų (diskretaus laiko žingsnių) pastumsime diskrečias signalo g reikšmes atžvilgiu kito signalo f reikšmių. Kartais d vadinamas *vėlinimu* (angliškai “delay”).

Būtent, su įvairiais d skaičiais apskaičiuosime koreliacijos koeficientą (29), kuriame

$$x = (f_0, f_1, f_2, \dots, f_{N-d}), \quad y = (g_d, g_{d+1}, g_{d+2}, \dots, g_N), \quad d = 0, 1, \dots, [N/2]. \quad (30)$$

Abi sekos x ir y išraiškoje (30) turi po vienodą kiekį $N-d+1$ elementų, o $[N/2]$ žymi sveikąją skaičiaus $N/2$ dalį. Iš principo galima nagrinėti ir atvejus kai $d = [N/2]+1, [N/2]+2, \dots, N$, tačiau su tokiais d sekose x ir y nebebus bendrų laiko indeksų, be to, kai d priartės prie N , šios sekos turės vos po keletą elementų. Praktiškai pakanka ištirti koreliacijos priklausomybę laiko intervale, lygiame pusei signalų stebėjimo trukmės, t. y. su d reikšmėmis, apibrėžtomis (30).

Taigi, remiantis išraiškomis (29) ir (30), diskrečių signalų f ir g tarpusavio koreliacijos funkcija (angliškai “cross-correlation function”), priklausanti nuo argumento d , apibrėžiama kaip

$$r_{fg}(d) = \frac{\sum_{j=0}^{N-d} (f_j - \bar{f})(g_{d+j} - \bar{g})}{\sqrt{\sum_{j=0}^{N-d} (f_j - \bar{f})^2 \sum_{j=0}^{N-d} (g_{d+j} - \bar{g})^2}}, \quad d = 0, 1, \dots, [N/2], \quad (31)$$

čia

$$\bar{f} = \frac{1}{N-d+1} \sum_{i=0}^{N-d} f_i, \quad \bar{g} = \frac{1}{N-d+1} \sum_{i=0}^{N-d} g_{d+i}.$$

Reikšmė $r_{fg}(0)$, paskaičiuota kai $d = 0$, atitinka originalių (nepaslinktų laike) signalų duomenų koreliacijos koeficiento reikšmę (ją mes nagrinėjome 5.2.1 skyrelyje).

Alternatyvus tarpusavio koreliacijos funkcijos apibrėžimas

Kartais, naudojant periodinį signalo reikšmių pratęsimą

$$g_{N+1} = g_0, \quad g_{N+2} = g_1, \quad \dots,$$

sutinkamas alternatyvus tarpusavio koreliacijos funkcijos apibrėžimas:

$$r_{fg}^{alternative}(d) = \frac{\sum_{j=0}^N (f_j - \bar{f})(g_{d+j} - \bar{g})}{\sqrt{\sum_{j=0}^N (f_j - \bar{f})^2 \sum_{j=0}^N (g_{d+j} - \bar{g})^2}}, \quad \bar{f} = \frac{1}{N+1} \sum_{i=0}^N f_i, \quad \bar{g} = \frac{1}{N+1} \sum_{i=0}^N g_i.$$

Tuomet, daugeliui realių signalų (patikrinkite savarankiškai!),

$$r_{fg}^{alternative}(d) \approx r_{fg}(d),$$

jeigu N pakankamai didelis, o indeksas d įgyja sąlyginai artimas nuliui reikšmes.

Tarpusavio koreliacijos funkcija neigiamiems d

Pastebėjime, kad skirtingai nuo koreliacijos koeficiento, tarpusavio koreliacijos funkcija *nėra* simetrinė signalų f ir g atžvilgiu, tai yra, bendru atveju $r_{fg}(d) \neq r_{gf}(d)$.

Pagal apibrėžimą (30), vektorių x , sudaryta iš signalo f reikšmių galima vertinti kaip reikšmes “dabartyje”, o vektorių y , sudaryta iš signalo g reikšmių – kaip reikšmes “ateityje” (su “ateities” gyliu d).

Signalų f ir g sukeitimas vietomis reiškia, kad (jeigu f reikšmes vis dar laikome “dabartimi”) tarpusavio koreliacijos funkcijos argumentas d , vietoje to, kad žymėtų nuotolį iki “ateities”, pradeda žymėti nuotolį iki “praeities”. Todėl, funkciją $r_{gf}(d)$ su $d = 0, 1, \dots, [N/2]$ galima interpretuoti kaip funkcijos $r_{fg}(d)$ funkcijos pratęsimą į neigiamas argumento reikšmes $d = 0, -1, \dots, -[N/2]$:

$$r_{fg}(d) = r_{gf}(|d|), \quad d = 0, -1, \dots, -[N/2]. \quad (32)$$

Tarpusavio koreliacijos funkcijos apibrėžimas tiek teigiamiems (formulė (31)), tiek neigiamiems (formulė (32)) d , leidžia ištirti dviejų signalų panašumą, vieną iš jų stumdant laike tiek į kairę, tiek į dešinę kito signalo atžvilgiu.

Tarpusavio koreliacijos funkcija realaus laiko t mastelyje

Signalų f ir g tarpusavio koreliacijos funkcijos $r_{fg}(d)$, apibrėžtos formulėmis (30), (31) ir (32), argumentas d dažniausiai neinformatyvus. Todėl braižant tarpusavio koreliacijos funkcijos grafiką geriau naudoti priklausomybę nuo realaus laiko t .

Signalų f ir g tarpusavio koreliacijos funkciją, priklausančią nuo t , žymėsime $R_{fg}(t)$:

$$R_{fg}(t) = r_{fg}(d), \quad t = d \Delta t, \quad d = -[N/2], -[N/2]+1, \dots, 0, 1, \dots, [N/2]. \quad (33)$$

čia Δt yra signalų f ir g diskretizavimo žingsnis (abu signalai turi būti diskretizuoti su tuo pačiu žingsniu), t. y. realaus laiko tarpas skiriantis du iš eilės einančius signalo reikšmių matavimus eksperimente.

Skaičiavimų optimizavimo galimybė, naudojant greitosios Furje transformacijos algoritmą

Matyti, kad tarpusavio koreliacijos funkcijos reikšmėms suskaičiuoti reikia atlikti proporcingą N^2 skaičių aritmetinių veiksmų. Jeigu apdorojamų duomenų kiekis N yra pakankamai didelis skaičius, galima modifikuoti tarpusavio koreliacijos funkcijos apibrėžimą (žr. “Alternatyvus tarpusavio koreliacijos funkcijos apibrėžimas” aukščiau) taip, kad jis būtų suderinamas su 7.1.4 skyrelyje apibrėžta išraiška (panašia į konvoliuciją suma). Tokiu būdu skaičiavimuose bus galima taikyti itin efektyvų (kompiuterio procesoriaus laiko požiūriu) greitosios Furje transformacijos algoritmą, aprašytą 7.2 skyrelyje.

Ši pastaba teisinga ir vieno signalo autokoreliacijos funkcijai, kurią nagrinėsime 5.3 skyrelyje.

5.2.3 Tarpusavio koreliacijos grafiko interpretavimas. Pavyzdžiai

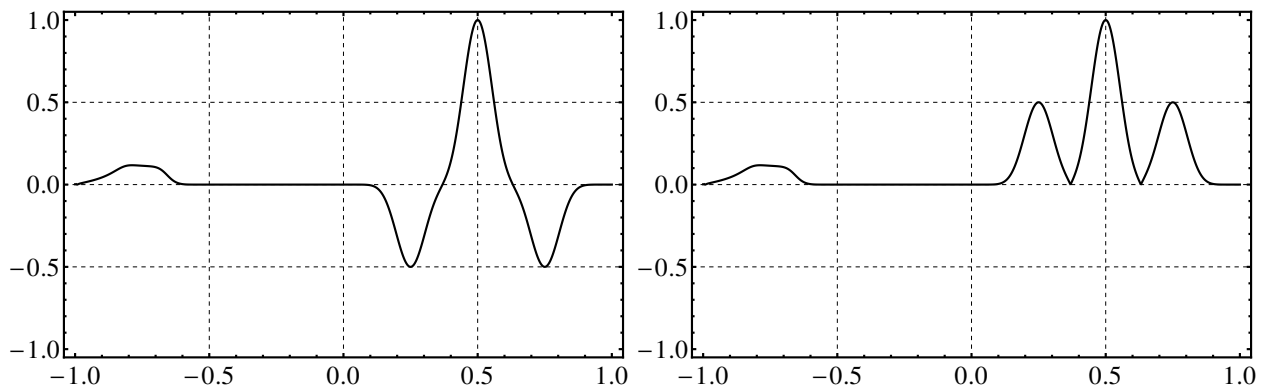
Nusibraižę tarpusavio koreliacijos funkcijos $R_{fg}(t)$, apibrėžtos išraiška (33), grafiką (žr., pvz., 44 pav., 45 pav.), konstatuojame (žr. 5.1 skyrelyje nustatytas koreliacijos koeficiento savybes) galimus variantus:

- Visiems t , $|R_{fg}(t)| \in [0, 0.2]$: tarp signalų *nėra* statistinio panašumo arba jis labai silpnas.

- Tam tikriems t , $|R_{fg}(t)| \in (0.2, 0.5]$: tarp signalų yra *silpnas* statistinis ryšys.
- Tam tikriems t , $|R_{fg}(t)| \in (0.5, 0.7]$: tarp signalų yra *vidutinis* statistinis ryšys.
- Tam tikriems t , $|R_{fg}(t)| \in (0.7, 1]$: tarp signalų yra *stiprus* statistinis ryšys.

Ta funkcijos $R_{fg}(t)$ argumento t vertė, su kuria funkcijos modulis pasiekia maksimumą, parodo kaip reikėtų pastumti laike antrąjį signalą atžvilgiu pirmojo, kad jie taptų labiausiai panašūs.

Jeigu $|R_{fg}(t)|$ įgyja maksimalią reikšmę su $t = 0$, vadinasi duomenys neturi savitarpio vėlinimo nei į ateitį, nei į praeitį – jie yra labiausiai panašūs tokie, kokie yra (be stumdymų laike).



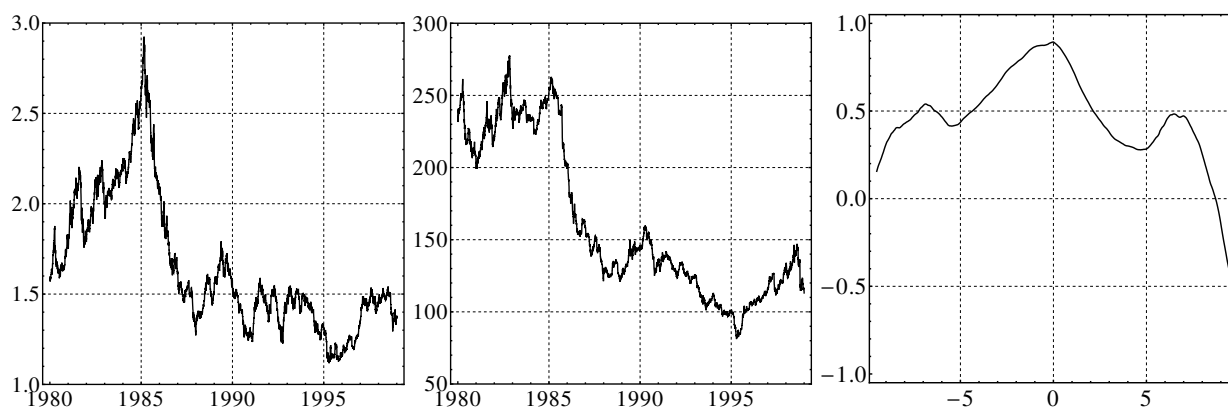
44 pav.: Signalų $f(t)$ ir $g(t)$, pavaizduotų 43 pav., tarpusavio koreliacijos funkcijos $R_{fg}(t)$, apibrėžtos išraiška (33), priklausomybė nuo laiko t . Iliustracija kairėje: funkcija $R_{fg}(t)$. Iliustracija dešinėje: jos modulis $|R_{fg}(t)|$. Rezultatai parodo, kad abu signalai iš tiesų labai panašūs (koreliacija pasiekia vienetą), bet turimi antrojo signalo duomenys pastumti laike per 0.5 sekundės į priekį, atžvilgiu pirmojo signalo duomenų.

Pavyzdžiui, interpretuodami 44 pav. pavaizduotą tarpusavio koreliacijos funkcijos grafiką, iš karto pastebime, kad funkcijos $|R_{fg}(t)|$ maksimumas (be to, labai ryškus, išreikštas smaile) yra ties $t = 0.5$. Maksimumo vertė yra 1, taigi analizuojami signalai itin panašūs tarpusavyje. Iš to, net nematydami pačių signalų grafikų, galime daryti išvadą, kad antrąjį signalą $g(t)$ reikėtų pastumti per 0.5 sekundės atgal, kad panašumas taptų maksimalus.

Kitame pavyzdyje, lygindami dviejų skirtingų valiutų porų kursų signalus (žr. 45 pav.), konstatuojame, kad signalai nors ir ne identiški, bet labai statistiškai panašūs ($\max |R_{fg}(t)| \approx 0.9$). Be to, laikas abiejuose signaluose yra jau sinchronizuotas (nes tarpusavio koreliacijos funkcijos maksimumas yra ties $t = 0$).

Gali kilti klausimas: ar galima tarpusavio koreliacijos metodu tirti signalus, kurie turi skirtingas dimensijas, nekonvertuojant jų į vieną matavimo sistemą? Pavyzdžiui, viename mieste oro temperatūra matuojama laipsniais Celsijaus skalėje, kitame – Farenheito. Atsakymas paprastas: taip, galima – visiškai tuo nesirūpinant. Tarpusavio koreliacijos funkcijos reikšmės nepasikeis, jeigu vieną iš signalų tiesiškai transformuosime (padauginsime iš bet kokios nenulinės konstantos ir pridėsime arba atimsime bet kokį skaičių). 45 pav. pavyzdyje Japonijos jenų vertė apie šimtą kartų mažesnė už Šveicarijos frankų (nagrinėjamų signalų masteliai skiriasi apie šimtą kartų), bet tai neįtakoja funkcijos $R_{fg}(t)$ reikšmių.

Taigi, šiuo metodu be papildomų rūpesčių galime lyginti net ir visiškai skirtingos dimensijos signalus, pavyzdžiui iškritusio kritulių kiekio signalą ir sergančiųjų infekcinėmis ligomis skaičiaus signalą.



45 pav.: Iliustracija kairėje: JAV dolerių keitimo kurso į Šveicarijos frankus (žr. 15 pav.) signalas. Iliustracija centre: JAV dolerių keitimo kurso į Japonijos jenas (žr. 8 pav., grafiką kairėje) signalas tuo pačiu laiko tarpu. Iliustracija dešinėje: šių signalų tarpusavio koreliacijos funkcijos $R_{fg}(t)$, apibrėžtos išraiška (33), priklausomybė nuo laiko t .

Signalų tarpusavio koreliacija ne tik nustato arba paneigia ryši tarp dviejų vyksmų, bet ir įvertina laiko trukmę, per kurią vieno vyksmo efektas pasireiškia kitam. Pavyzdžiui, kaip greitai (ir kiek ženkliai) reklama išaugina apyvartą, medikamentai įtakoja sveikatos būklę, sumažinti arba padidinti mokesčiai paveikia ekonominius rodiklius ir pan.

Kadangi tarpusavio koreliacijos metodu galima išmatuoti vėlavimą laike tarp dviejų signalų, jis taip pat gali būti taikomas ir greičio, su kuriuo sklinda mus dominantis procesas, įvertinimui (jeigu žinome atstumą tarp dviejų registravimo punktų). Pavyzdžiui, taršalų sklaidimas dirvožemyje, registruojant jų intensyvumą įvairiame gylyje.

Apibendrinant galima pasakyti, kad nors lyginti paprasčiausius signalus ir ieškoti juose panašių fragmentų galima ir vizualiai (tiesiog lyginant jų grafikus), tarpusavio koreliacijos algoritmas ne tik automatizuoja šį procesą, bet ir gali surasti panašumus, kurie nematomi plika akimi, be to panašumo laipsnį ir signalų tarpusavio vėlinimą įvertina konkrečiais skaičiais.

Tačiau programuojant šį algoritmą ir darant išvadas iš gautų rezultatų reikia būti itin atidiems: žr. 5.1 skyrelyje išdėstytas pastabas apie *aritmetinio vidurkinimo pavojus* bei apie *statistinę ryšį ir priežastingumą*.

5.3 Signalų autokoreliacijos funkcija

Šiame skyrelyje kalbėsime apie *vieno signalo autokoreliaciją*, kuri įvertina signalo panašumą į savo paties fragmentus, esančius įvairiais laiko nuotoliais.

Kitaip sakant, signalo *autokoreliacijos* (angliškai “autocorrelation”) funkcija yra ne kas kita kaip tarpusavio koreliacijos funkcija (apie ją kalbėjome 5.2.2 skyrelyje), paskaičiuota tarp signalo su pačiu savimi.

Matematiškai signalo $f(t)$ autokoreliacijos funkcija $r(d)$, kurios argumentas d turi tą pačią prasmę, kaip tarpusavio koreliacijos funkcijos $r_{fg}(d)$ išraiškoje (31), apibrėžiama kaip

$$r(d) = r_{ff}(d).$$

Ji įvairiems d įvertina koreliaciją tarp vektorių

$$x = (f_0, f_1, f_2, \dots, f_{N-d}), \quad y = (f_d, f_{d+1}, f_{d+2}, \dots, f_N), \quad d = 0, 1, \dots, [N/2].$$

Vektorius x gali būti traktuojamas kaip signalo “dabartis”, o y – kaip signalo “ateitis”, nutolusi nuo “dabarties” per d indeksų (diskretaus laiko žingsnių). Arba kitaip: x “praeitis”, tuomet y “dabartis”.

Paskaičiavę koreliacijas tarp x ir y įvairiems d susidarome vaizdą kaip ilgai “dabartis atsimena praeitį” ir kaip greitai ši “atmintis” gęsta (arba ne) didinant “praeities gylį” d , kitaip sakant kiek inertiškas laike yra signalas.

Tai gali būti įdomu, pavyzdžiui, tiriant finansų rinkų atsistatymo greitį ir pobūdį po patirto streso (tarkime, krizinio laikotarpio), ligonio sveikimą po ligos, ir pan., tačiau autokoreliacijos taikymai neapsiriboja vien tokio pobūdžio uždaviniais (žr. toliau 5.3.1 skyrelį).

Autokoreliacijos funkcijos reikšmių seka (grafikas), taip pat vadinama *korelograma* (angliškai “correlogram”).

Užrašykime tiesioginę autokoreliacijos funkcijos apibrėžimą, nesinaudodami tarpusavio koreliacijos funkcija (31). Taigi, diskretaus signalo f , apibrėžto eksperimentiniais duomenimis

$$f_0, f_1, \dots, f_N,$$

autokoreliacijos funkcija, priklausanti nuo argumento d , apibrėžiama kaip

$$r(d) = \frac{\sum_{j=0}^{N-d} (f_j - \bar{f})(f_{d+j} - \bar{F})}{\sqrt{\sum_{j=0}^{N-d} (f_j - \bar{f})^2 \sum_{j=0}^{N-d} (f_{d+j} - \bar{F})^2}}, \quad d = 0, 1, \dots, [N/2], \quad (34)$$

čia

$$\bar{f} = \frac{1}{N-d+1} \sum_{i=0}^{N-d} f_i, \quad \bar{F} = \frac{1}{N-d+1} \sum_{i=0}^{N-d} f_{d+i}.$$

Analogiškai kaip tarpusavio koreliacijos funkcijai (žr. 5.2.2 skyrelį), nėra prasmės skaičiuoti autokoreliaciją kai $d > [N/2]$, o kai kurie autoriai netgi siūlo apsiriboti tik $d = 0, 1, \dots, [N/4]$.

Alternatyvus autokoreliacijos funkcijos apibrėžimas

Analogiškai kaip tarpusavio koreliacijai, kartais, naudojant periodinę signalo reikšmių pratęsimą

$$f_{N+1} = f_0, \quad f_{N+2} = f_1, \quad \dots,$$

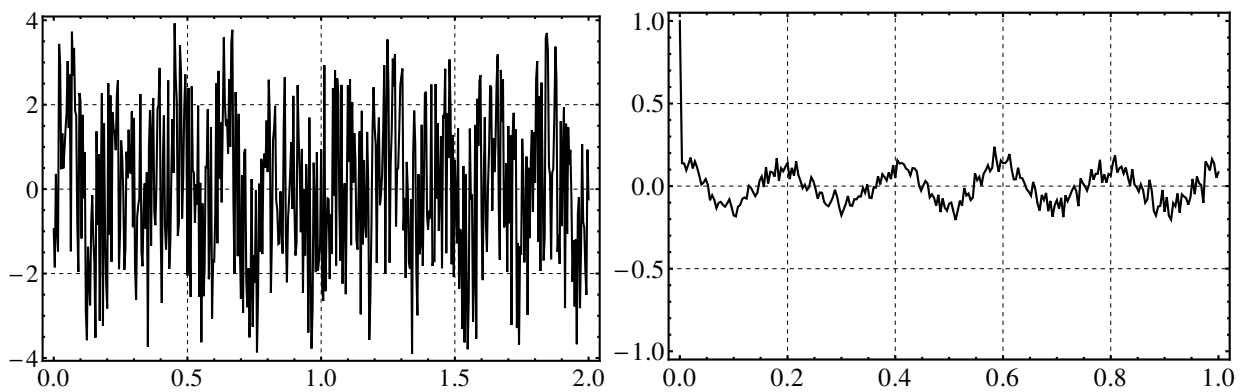
sutinkamas alternatyvus autokoreliacijos funkcijos apibrėžimas:

$$r^{alternative}(d) = \frac{\sum_{j=0}^N (f_j - \bar{f})(f_{d+j} - \bar{f})}{\sum_{j=0}^N (f_j - \bar{f})^2}, \quad \bar{f} = \frac{1}{N+1} \sum_{i=0}^N f_i.$$

Tuomet, daugeliui realių signalų (patikrinkite savarankiškai!), $r^{alternative}(d) \approx r(d)$, jeigu N pakankamai didelis, o indeksas d įgyja sąlyginai artimas nuliui reikšmes.

Autokoreliacijos funkcijos savybės

Tiesiog iš apibrėžimo seka pora svarbių autokoreliacijos savybių:



46 pav.: Iliustracija kairėje: intensyvaus triukšmo paveiktas periodinis signalas. Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$, apibrėžtos išraiška (35), priklausomybė nuo laiko t . Matyti, kad autokoreliacija išryškina plika akimi nematomą signalo periodiškumą.

- Bet kokiam signalui f , kai $d = 0$, turime $r(0) = 1$. Taigi, autokoreliacinė funkcija visada prasideda nuo 1 ir, didėjant d , dažniausiai turi tendenciją mažėti. Žr., pvz., 46 pav. ir kitas šio skyrelio iliustracijas.
- Kadangi tarpusavio koreliacija neigiamiems d apibrėžiama kaip $r_{fg}(d) = r_{gf}(|d|)$, $d < 0$ (žr. formulę (32)), tai autokoreliacijai gauname $r(-d) = r(d)$. Vadinasi, autokoreliacijos funkcija yra *simetrinė* taško $d = 0$ atžvilgiu – pakanka nagrinėti jos priklausomybę tik neneigiamiems $d = 0, 1, \dots, [N/2]$.

Autokoreliacijos funkcija realaus laiko t mastelyje

Analogiškai kaip tarpusavio koreliacijos atveju, vietoje neinformatyvaus kintamojo d , realiuose taikymuose prasmingiau naudoti autokoreliacijos funkciją su argumentu t (realiuoju laiku).

Signalas f autokoreliacijos funkciją, priklausančią nuo argumento t , žymėsime $R(t)$:

$$R(t) = r(d), \quad t = d \Delta t, \quad d = 0, 1, \dots, [N/2]. \quad (35)$$

čia Δt yra signalo f diskretizavimo žingsnis, t. y. realaus laiko tarpas skiriantis du iš eilės einančius signalo reikšmių matavimus eksperimente.

Formulė (35) yra analogiška (33).

5.3.1 Autokoreliacijos grafiko interpretavimas. Pavyzdžiai

Autokoreliacijos funkcijos $R(t)$, apibrėžtos išraiška (35), grafikas (korelograma) gali suteikti šios informacijos apie analizuojamą signalą $f(t)$:

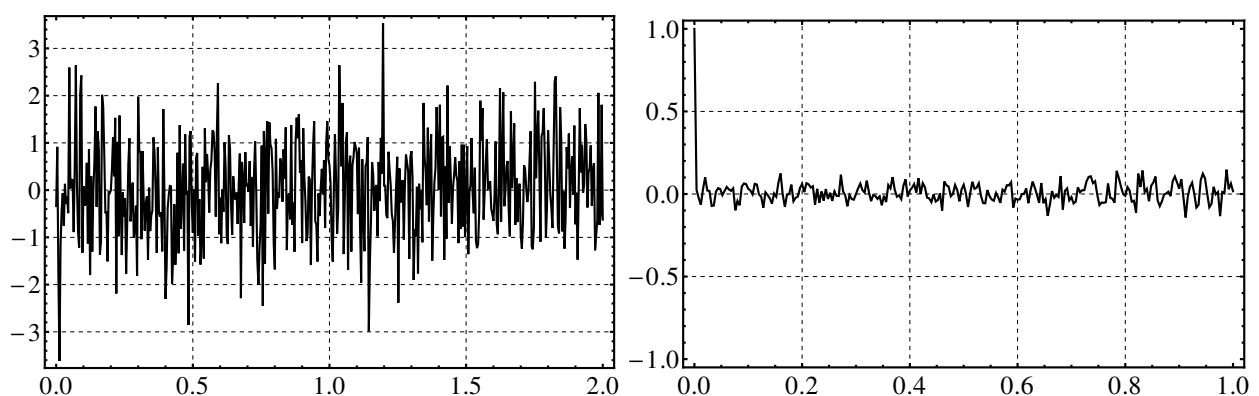
- Signalas *kvazi-periodiškas*. Jeigu signalo reikšmėse yra kvazi-periodinių darinių, tai korelograme pamatysime to paties dažnio svyravimus, žr. 46 pav., 49 pav., 50 pav., 52 pav. Be to, autokoreliacija paryškina apytikslus signalo reikšmių pasikartojimus, kurie gali būti užgožti triukšmo ir nepastebimi plika akimi (46 pav.).
- *Triukšmas, atsitiktinio pobūdžio, nereguliarūs* signalai. Jeigu autokoreliacija labai greitai krenta iki artimų nuliui reikšmių bei išsilaiko artima nuliui visame grafiko intervale, vadinasi signalo reikšmės nesusijusios arba labai silpnai susijusios su reikšmėmis praityje. Artimomis nuliui autokoreliacijos reikšmėmis laikomos tokios, kurių modulis neviršija $2/\sqrt{N}$. Tokia autokoreliacija būdinga triukšmams, nereguliariems signalams. Pavyzdys: 47 pav.

- *Stacionarus* signalas, pasižymintis tam tikros trukmės *atmintimi* (inercija). Daugeliui sudėtingos kilmės stacionarių signalų būdinga tam tikrą laiką išsilaikanti priklausomybė nuo buvusių signalo reikšmių. Tai yra, signalas “atsimena” prieš tam tikrą laiką buvusias savo paties reikšmes, yra jų įtakotas. Inercija pasižymi ciklonai ir kiti dariniai meteorologijoje, ligonio sveikatos būklės progresas medicinoje ir daugelis kitų signalų. Tokio signalo koreliacija priartėja prie nulio ne iš karto (kaip atsitiktinio pobūdžio signaluose), bet per tam tikrą laiką tarpą, nusakantį *inercijos trukmę*. Priartėjęs prie nulio, autokoreliacijos grafikas toks (artimas nuliui) ir išlieka (skirtingai nuo kvazi-periodinių signalų). Pavyzdys: 48 pav.
- *Nestacionarus* signalas, turintis *trendus*. Kadangi trendas dominuoja (užgožia) signalo reikšmes, pats signalas autokoreliacijos požiūriu tampa panašus į tiesinį, taigi koreliacija išsilaiko labai ilgai. Tokiems signalams autokoreliacijos grafikas labai ilgai nepriartėja prie nulio. Norint ištirti subtilesnes signalo savybes, prieš tai reikėtų eliminuoti iš jo trendus (pvz., nustatyti trendus glodinant signalą 2 skyriuje aptartais algoritmais ir išminusuoti juos iš signalo reikšmių). Pavyzdžiai: 54 pav., 55 pav., 51 pav.

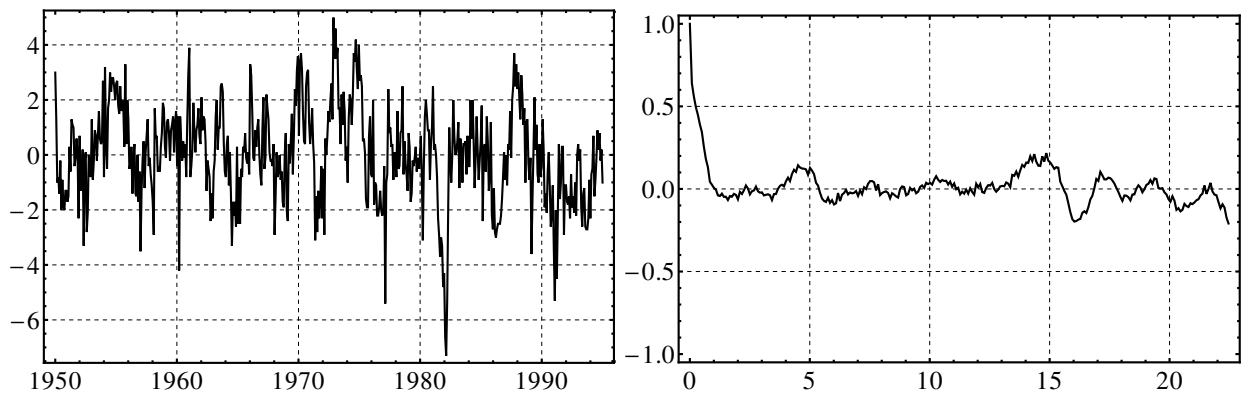
Prieš pradedant skaičiuoti signalo autokoreliaciją, pageidautina patikrinti ar signalo reikšmėse nėra artefaktų (beprasmių reikšmių, atsiradusių, pavyzdžiui, dėl registravimo aparatūros broko). Atskiros “išsišokusios” reikšmės, kurių dydis ženkliai viršija visas kitas, gali iškreipti korelogramą. Taigi, esant galimybei, tokios reikšmės turi būti pašalintos iš duomenų pradiniam signalo apdorojimo etape.

Autokoreliacija gali praversti ir tiriant signalo *rekurentinę diagramą* (žr. 3 skyrių) bei skaičiuojant signalo *koreliacinę dimensiją* (žr. 4 skyrių). Būtent, apibrėžiant signalo būsenas minėtuose algoritmuose, parametrai d reikėtų rinktis proporcingą šiame skyriuje nagrinėto (ir neatsitiktinai tuo pačiu simboliu pažymėto) indekso d reikšmei, su kuria signalo autokoreliacijos funkcija pirmą kartą tampa artima nuliui (jos modulis neviršija $2/\sqrt{N}$).

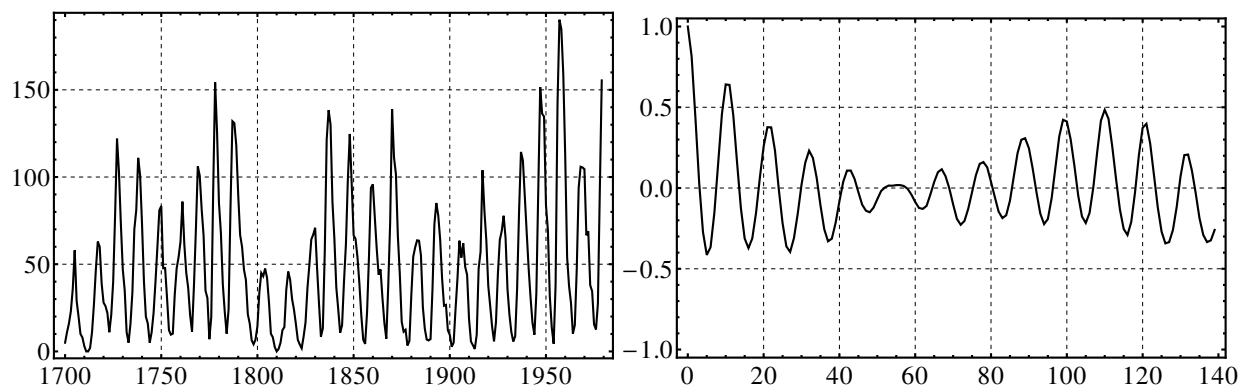
Siekiant ženkliai pagreitinti algoritmą, galima modifikuoti autokoreliacijos apibrėžimą. Analogiškas pastebėjimas buvo išsakytas tarpusavio koreliacijai, žr. “Skaičiavimų optimizavimo galimybė, naudojant greitosios Furje transformacijos algoritmą” 5.2.2 skyrelio pabaigoje.



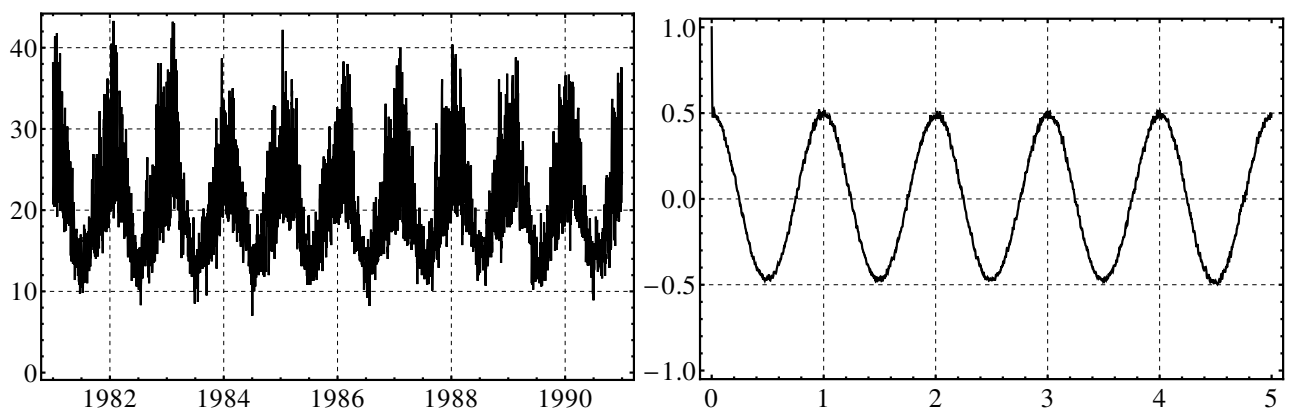
47 pav.: Iliustracija kairėje: atsitiktinis signalas, vadinamas *baltas triukšmas* (angliškai “white noise”), t. y. visos signalo reikšmės – tarpusavyje nepriklausomi atsitiktiniai skaičiai iš standartinio normaliojo pasiskirstymo. Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$, apibrėžtos išraiška (35), priklausomybė nuo laiko t .



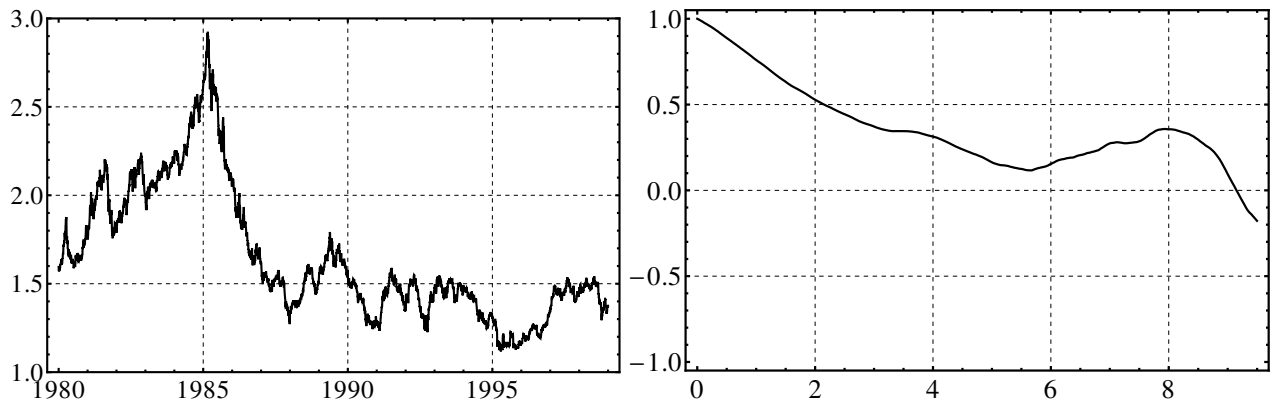
48 pav.: Iliustracija kairėje: nuokrypių nuo vidutinio okeano lygio signalas (susijęs su klimato dariniu El Niño), registruotas Ramiojo vandenyno pietuose (žr. 42 pav.). Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$ priklausomybė nuo laiko t . Pastebėkime, kad autokoreliacija nukrinta iki nulio ne iš karto, bet ties $t = 1$ metai. Taigi, signalas turi tam tikrą inerciją (atmintį) – signalo reikšmės beveik metus įtakojaamos buvusių signalo reikšmių (signalų praeities).



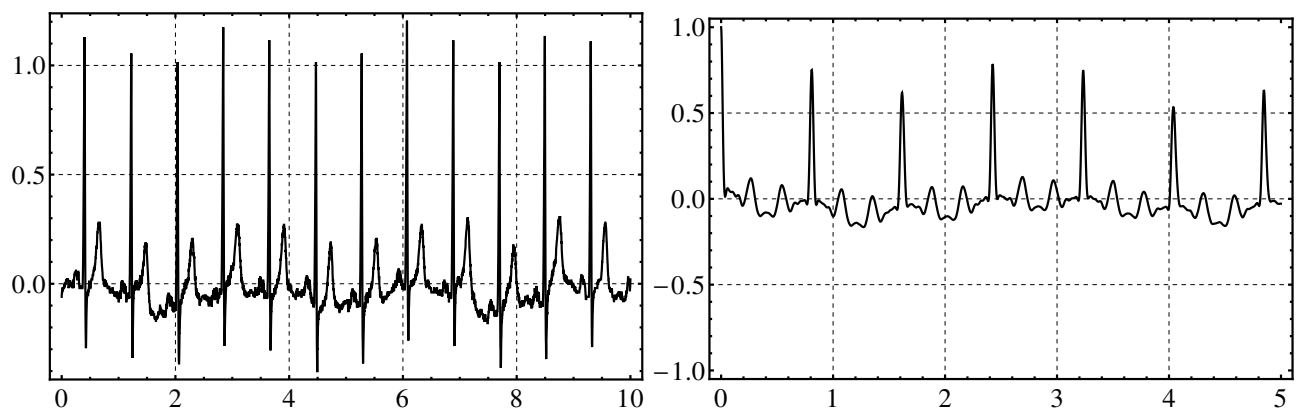
49 pav.: Iliustracija kairėje: Saulės dėmių skaičiaus per metus (žr. 40 pav.) signalas. Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$ priklausomybė nuo laiko t .



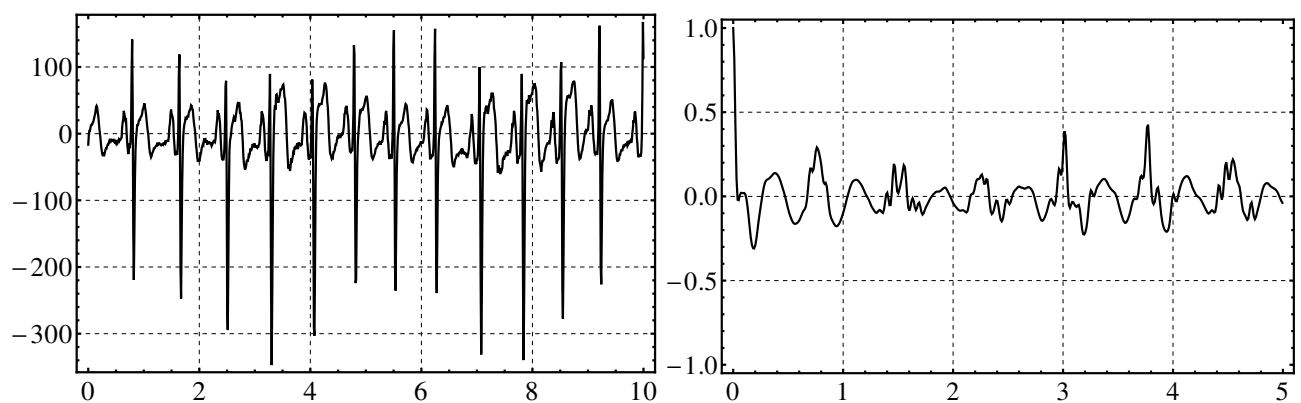
50 pav.: Iliustracija kairėje: dienos maksimalios oro temperatūros Melburne (Australija) svyravimo (žr. 12 pav., grafiką kairėje) signalas. Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$ priklausomybė nuo laiko t .



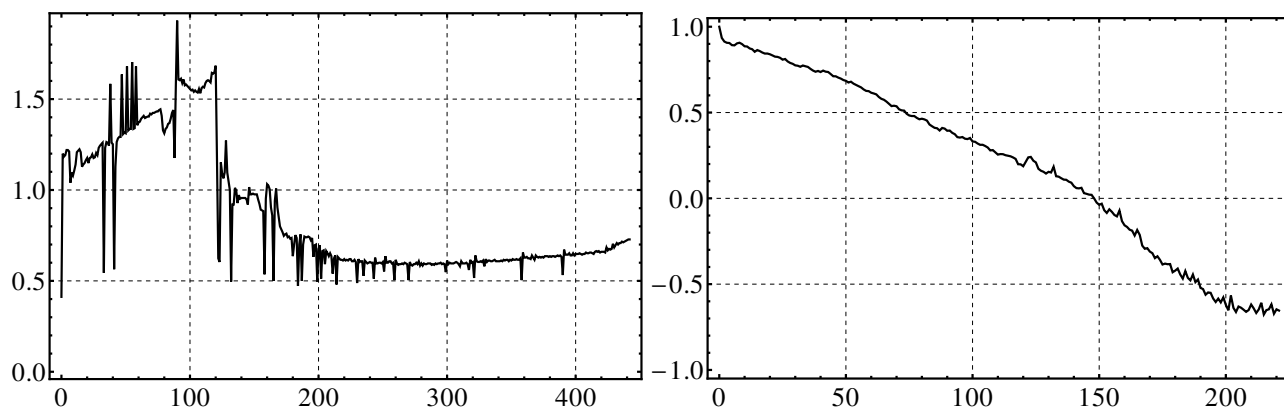
51 pav.: Iliustracija kairėje: JAV dolerių keitimo kurso į Šveicarijos frankus (žr. 15 pav.) signalas. Iliustracija dešinėje: signalo autokoreliacijos funkcijos $R(t)$ priklausomybė nuo laiko t .



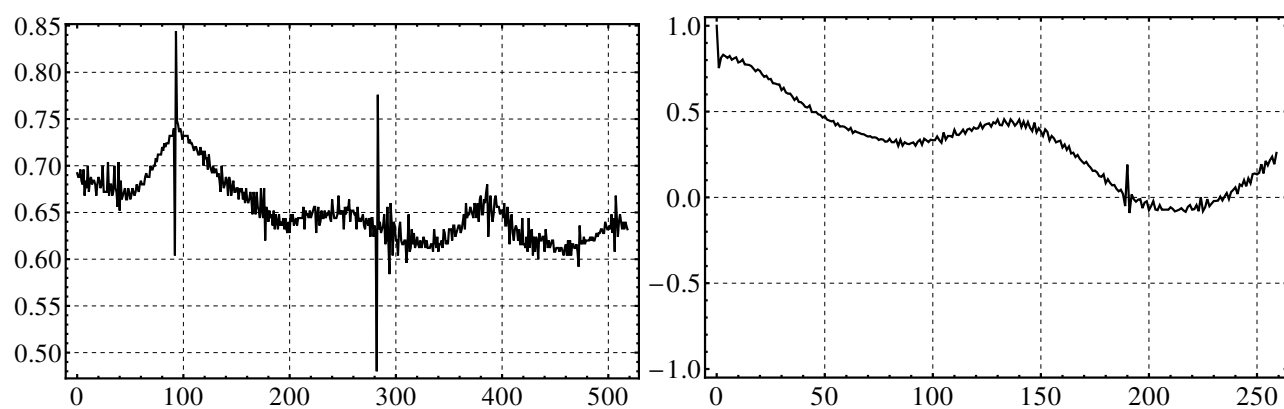
52 pav.: Elektrokardiograma (EKG) (kairėje), žr. 21 pav., ir jos autokoreliacija $R(t)$ (dešinėje).



53 pav.: Encefalograma (kairėje), žr. 1 pav., ir jos autokoreliacija $R(t)$ (dešinėje).



54 pav.: EKG RR intervalai (kairėje), žr. 21 pav., ir jų autokoreliacija $R(t)$ (dešinėje).



55 pav.: Kitos EKG RR intervalai (kairėje) ir jų autokoreliacija $R(t)$ (dešinėje).

6 Signalų komponentų išskyrimas. Furje skleidiniai

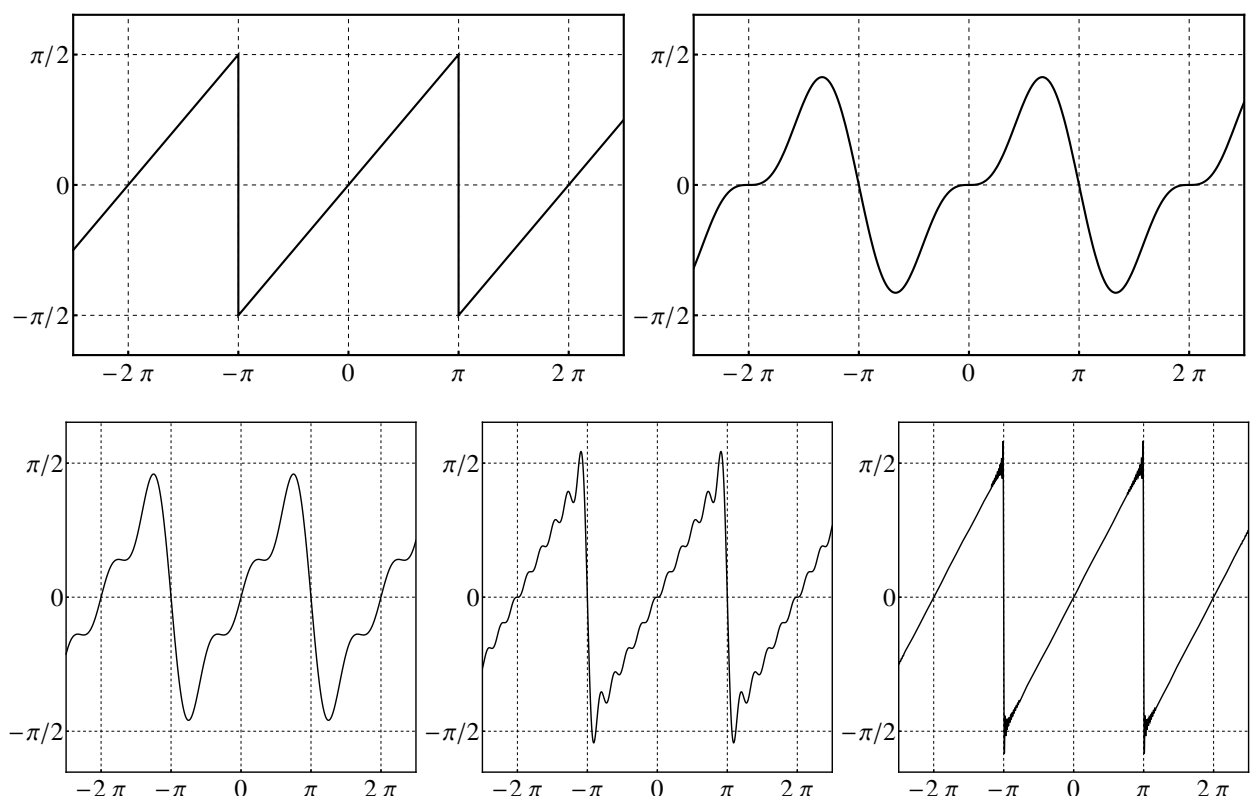
6.1 Furje skleidinio sąvoka ir pavyzdžiai

6.1.1 Furje hipotezė ir jos eksperimentinis patikrinimas

Prancūzų matematikas ir fizikas Ž. B. Ž. Furje (Jean Baptiste Joseph Fourier, 1768–1830) 1807 metais iškėlė hipotezę, kad bet kokios formos signalą (funkciją) $f(t)$ intervale $-\pi \leq t \leq \pi$ galima išreikšti (išskaidyti) elementariųjų trigonometrinių funkcijų – sinusų ir kosinusų, pasižyminčių įvairiais dažniais, suma.

Jeigu Furje hipotezė teisinga, akivaizdu, kad jos vaidmuo signalų apdorojime gali būti labai didelis. Pirma, bet kurį signalą nesudėtingos laiko t transformacijos dėka galime suvesti į intervalą $-\pi \leq t \leq \pi$. Antra, tai reiškia, kad sumuodami skirtingų dažnių paprastas trigonometrines funkcijas, galime išgauti kokios tik norime formos signalą ir atvirkščiai – bet kokią signalą galime tarsi “išardyti”: išdėlioti į standartines sudaromasias komponentes (apibrėžtas nesudėtingomis trigonometrinėmis funkcijomis).

Kadangi pats Furje nepateikė savo iškelto teiginio įrodymo, praėjo nemažai laiko, kol jo hipotezė sulaukė rimto dėmesio. Ir tai suprantama, nes remiantis vien tik intuicija sunku ją patikėti! Juk sinusų ir kosinusų grafikai – glodžios kreivės, o signalas gali būti visiškai nepanašus į jas, pavyzdžiui – pjūklo formos (žr. 56 pav., iliustracija viršuje kairėje).



56 pav.: Pjūklo pavidalo signalas (iliustracija viršuje kairėje) ir trigonometrinis skleidinys (suma) (36) kai $M = 2$ (iliustracija viršuje dešinėje), kai $M = 3$ (iliustracija apačioje kairėje), kai $M = 10$ (iliustracija apačioje centre) ir kai $M = 100$ (iliustracija apačioje dešinėje).

Patikrinkime Furje hipotezę šiam pjūklo formos signalui.

Nagrinėkime funkciją, kurios grafikas pavaizduotas 56 pav. viršuje dešinėje:

$$f(t) = \sin t - \frac{1}{2} \sin 2t,$$

kuri, kaip matome, yra dviejų trigonometrinių funkcijų suma. Kitais žodžiais, ši funkcija yra trigonometrinis *skleidinys* (suma), turintis du narius.

Pridėkime dar vieną narį, sudarydami naują skleidinį iš trijų narių (jo grafiką matome 56 pav. apačioje kairėje):

$$f(t) = \sin t - \frac{1}{2} \sin 2t + \frac{1}{3} \sin 3t.$$

Tęsdami toliau ir išlaikydami pradėtą dėsningumą, sudarytume skleidinį iš M narių:

$$f(t) = \sin t - \frac{1}{2} \sin 2t + \dots + \frac{(-1)^{M+1}}{M} \sin Mt = \sum_{k=1}^M b_k \sin kt, \quad b_k = \frac{(-1)^{k+1}}{k}. \quad (36)$$

Trigonometrinio skleidinio (36) koeficientai b_k nusako sinusinių bangų su *kampiniu dažniu* k *amplitudes*. Kitaip sakant, funkciją $f(t)$ išskleidėme trigonometrinio skleidiniu, sudarytu iš skirtingo dažnio komponentių, o šių komponentių dydžius apibrėžia koeficientai b_k .

56 pav. apačioje centre pavaizduotas skleidinio, sudaryto iš dešimties narių ($M = 10$), grafikas, o apačioje dešinėje – grafikas, kai $M = 100$. Taigi, didindami trigonometrinio skleidinio narių skaičių, vis artėjame prie pjūklo pavidalo signalo.

Atkreipkime dėmesį, kad pjūklo pavidalo signalas nėra glodus: jis turi trūkius taškuose $t = -\pi$ ir $t = \pi$. Tai, kad sumuodami trigonometrines funkcijas sugebėjome aproksimuoti net tokį signalą, turėtų ženkliai sustiprinti tikėjimą Furje hipoteze! Tiesa, tam tikrą nerimą gali kelti tai, kad arti trūkio taškų išlieka smulkūs svyravimai (apie šį reiškinį, vadinamą *Gibbs'o efektu*, pakalbėsime šio skyriaus pabaigoje – 6.3.4 skyrelyje).

Spektrinės analizės sąvoka. Analogija su šviesos spektru

Mus supančioje aplinkoje yra daugybė reiškinių, kuriuos galima išreikšti įvairaus dažnio virpesių sumomis. Tipiškas pavyzdys – šviesa. Žmogaus matoma šviesa yra ne kas kita kaip įvairaus dažnio, apytiksliai nuo 405 THz (raudonos spalvos šviesa) iki 790 THz (violetinės spalvos šviesa) elektromagnetinių bangų suma. Puikiai žinome, kad baltos spalvos šviesai perėjus per stiklinę prizmę, išvystame jos *spektrą*, sudarytą iš septynių spalvų, tai yra, prizmė išskaido baltą šviesą (sudarytą iš visų spalvų mišinio – elektromagnetinių bangų sumos) į atskiras komponentes, pagal kiekvienos iš jų dažnį. Taip yra todėl, kad stiklo (iš kurio pagaminta prizmė) lūžio rodiklis keičiasi priklausomai nuo bangos dažnio. Taigi, turėdami šviesos spektrą galime ištirti jo savybes, priklausomai nuo to, kokios spalvų kombinacijos jame matomos.

Analogiškai tiriami ir signalai: išskyrę signalo komponentes, pagal jų dažnį galime daug suprasti apie signalo kilmę ir savybes. Pavyzdžiui, triukšmams dažnai būdingi kitokie dažniai nei prasmingoms signalo komponentėms – atsiranda galimybė eliminuoti (nufiltruoti) nepageidaujamo dažnio komponentes ir vėl iš naujo “surinkti” signalą, apie tai kalbėsime 7.3 skyrelyje.

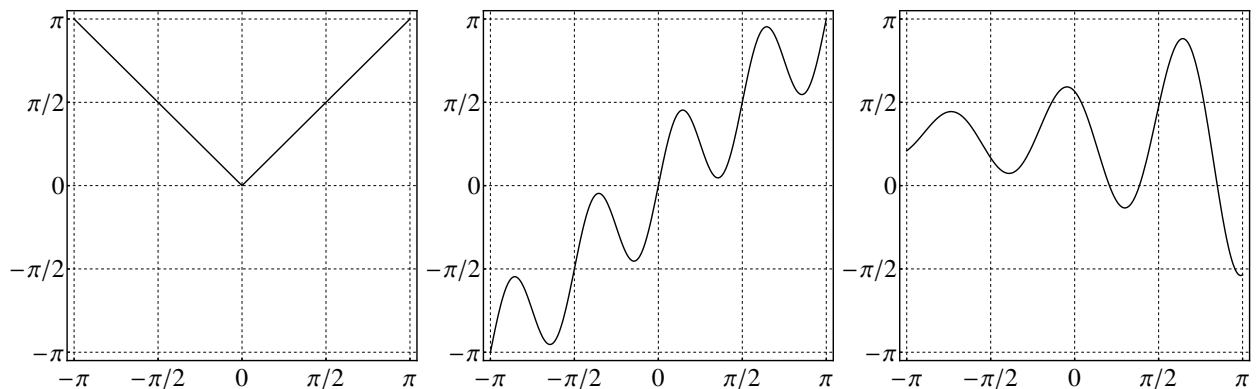
6.1.2 Lyginės ir nelyginės funkcijos

Trigonometriniame skleidinyje (36) elementariųjų signalo komponentių vaidmenį atlieka funkcijų seka

$$\sin t, \sin 2t, \dots, \sin Mt.$$

Siekdami kuo geriau atkartoti originalaus signalo formą, didiname jų skaičių M . Kai $M \rightarrow \infty$, gauname begalinę sinusų seką. Žinoma, praktiškai visada apsiribojame baigtiniu, nors ir pakankamai dideliu M .

Ar tikrai bet kokios formos signalą $f(t)$ intervale $-\pi \leq t \leq \pi$ galime išreikšti sumuodami sinusoides? Iš karto krinta į akis tai, kad tiek $\sin t$, tiek $\sin 2t$ (ir taip toliau) yra *nelyginės* funkcijos.



57 pav.: Iliustracija kairėje: lyginės funkcijos pavyzdys. Iliustracija centre: nelyginės funkcijos pavyzdys. Iliustracija dešinėje: nei lyginės, nei nelyginės funkcijos pavyzdys.

Prisiminkime, kad *lygine* funkcija vadinama funkcija $f(t)$, pasižyminti savybe

$$f(t) = f(-t),$$

taigi, bet kurios lyginės funkcijos grafikas simetriškas ordinačių ašies $t = 0$ atžvilgiu (žr. 57 pav., grafikas kairėje).

Tuo tarpu *nelygine* funkcija laikoma funkcija $f(t)$, tenkinanti

$$-f(t) = f(-t),$$

vadinasi, visų nelyginių funkcijų grafikai simetriški koordinačių pradžios taško $(0, 0)$ atžvilgiu (žr. 57 pav., grafikas centre).

Tačiau daugybė funkcijų yra nei lyginės, nei nelyginės (žr. 57 pav., grafikas dešinėje).

Kadangi visos sinusoidės – nelyginės funkcijos, tai jas dauginami iš konstantų ir sumuodami, taip pat gausime nelyginę funkciją. Todėl skleidiniu (36), net ir su kaip norima dideliu M , galime bandyti išreikšti tik nelygines funkcijas. Kadangi mūsų nagrinėtas pjūklo formos signalas kaip tik ir buvo nelyginė funkcija (intervale $-\pi \leq t \leq \pi$ apibrėžta kaip $f(t) = t/2$, o toliau periodiškai pratęsta), ją pavyko išskaidyti vien tik į sinusinių bangų komponentes.

Vietoje sinusų imdami kosinusus

$$1 = \cos 0t, \cos t, \cos 2t, \dots,$$

dauginami iš konstantų ir sumuodami, taip pat apibrėžtume tam tikrą trigonometrinį skleidinį, tačiau jis būtų tinkamas tik lyginio tipo signalams išskaidyti – visi kosinusai yra lyginės funkcijos.

Vadinasi, norėdami išreikšti trigonometrinį skleidinį bet kokios formos signalą (kuris gali būti tiek lyginė, tiek nelyginė, o dažniausiai nei lyginė, nei nelyginė funkcija), turime panaudoti ir sinusus, ir kosinusus.

6.1.3 Furje skleidinio apibrėžimas. Signalų spektras

Trigonometrine baze intervale $-\pi \leq t \leq \pi$ vadinama funkcijų seka (sistema)

$$1, \cos t, \sin t, \cos 2t, \sin 2t, \dots \quad (37)$$

Šias funkcijas dauginami iš konstantų a_k , b_k ir sumuodami, apibrėžiame signalo $f(t)$, kurio apibrėžimo sritis yra intervalas $-\pi \leq t \leq \pi$, *Furje skleidinį* (angliškai “Fourier series”):

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kt + \sum_{k=1}^{\infty} b_k \sin kt. \quad (38)$$

Skleidinio koeficientai a_0 , a_k , b_k vadinami *Furje koeficientais*. Netrukus (6.1.4 skyrelyje) pateiksime motyvą, kodėl išraiškoje (38) patogiau rašyti $a_0/2$ vietoje a_0 .

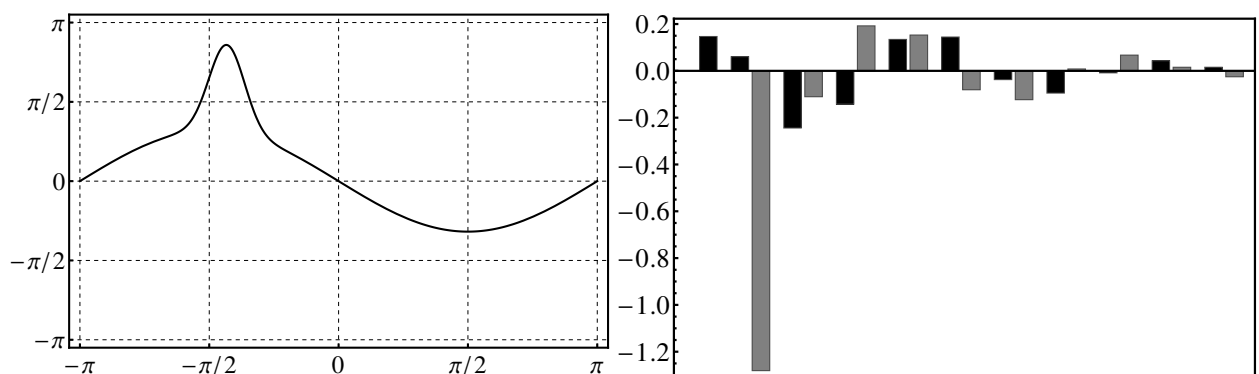
Pastebėkime, kad intervale $-\pi \leq t \leq \pi$ tiek kosinusų, tiek sinusų vidutinė reikšmė lygi nuliui. Taigi, laisvasis skleidinio narys $a_0/2$ nusako signalo $f(t)$ vidurkį (kai $-\pi \leq t \leq \pi$):

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) dt = \frac{a_0}{2}.$$

Signalų $f(t)$ komponentų dariniai, Furje skleidinyje turintys vienodą dažnį, vadinami *harmonikomis*, pavyzdžiui,

$$a_1 \cos t + b_1 \sin t - \text{pirmoji harmonika}, \quad a_2 \cos 2t + b_2 \sin 2t - \text{antroji harmonika, ir t. t.}$$

Signalų tyrimas, paremtas harmonikų tyrimu, vadinamas signalo *spektrine analize* (angliškai “spectral analysis”), taip pat *Furje analize* (angliškai “Fourier analysis”) arba tiesiog *harmonine analize* (angliškai “harmonic analysis”).



58 pav.: Iliustracija kairėje: signalo priklausomybė nuo laiko t (laiko srityje). Iliustracija dešinėje: to paties signalo priklausomybė dažnio srityje (signalų spektras), pateikti Furje koeficientai a_0, a_1, \dots, a_{10} ir b_1, b_2, \dots, b_{10} . Koeficientų a_k reikšmės reprezentuoja juodi stulpeliai, koeficientų b_k – pilki.

Iš esmės signalo komponentų išskyrimas Furje skleidiniu reiškia to paties signalo aprašymą *dažnio srityje*. Būtent, *laiko srityje*, kurioje kintamu dydžiu yra laikas t , signalas išreiškiamas funkcija $f(t)$. Tas pats signalas gali būti išreiškiamas Furje koeficientais (a_k, b_k) *dažnio srityje*, kurioje kintamu dydžiu yra dažnis k . Furje koeficientų priklausomybė nuo k (signalų priklausomybė nuo dažnio) vadinama *signalų spektru*.

Formules Furje koeficientų apskaičiavimui išvesime 6.1.4 skyrelyje.

Signalą galima vienareikšmiškai nusakyti vien tik jo spektru, t. y. žinant tik jo Furje koeficientus, per skleidinį (38) atstatyti signalo priklausomybę laiko srityje.

Pavyzdžiui, signalas, kurio išraiška laiko srityje yra $f(t) = \cos(t)$, dažnio srityje gali būti aprašytas taip: $a_1 = 1$, visi kiti a_k bei b_k lygūs nuliui.

Sudėtingesnio signalo priklausomybės laiko ir dažnio srityse pavaizduotos 58 pav. Kaip matyti iš spektro, jeigu Furje skleidinys (38) konverguoja, tai jo koeficientai $a_k \rightarrow 0$ ir $b_k \rightarrow 0$, kai $k \rightarrow \infty$. Taigi, praktiškai pakanka baigtinio Furje koeficientų skaičiaus apytiksliam signalo aprašymui.

Furje skleidinio egzistavimo sąlygos ir galimi apibendrinimai

Formaliai matematiškai žiūrint, ne visos funkcijos, apibrėžtos intervale $-\pi \leq t \leq \pi$ gali būti išreikštos Furje skleidiniu (38). Tačiau funkcija turi būti iš tiesų egzotinė, kad nebūtų įmanoma to padaryti! Įrodyta, kad pakanka, jog intervale $-\pi \leq t \leq \pi$ egzistuotų baigtinis funkcijos modulio integralas, funkcija turėtų baigtinį skaičių trūkių bei baigtinį kiekį lokaliųjų maksimumų ir minimumų – tuomet funkciją galima išskleisti Furje skleidiniu.

Taikymuose sutinkami signalai šiuos reikalavimus tenkina.

Gali kilti klausimas: ar signalus įmanoma išskaidyti tik į trigonometrinių bazinių funkcijų (37) sumą? Pasirodo, ne tik – galima naudoti ir kitokią signalo komponentų bazę, pavyzdžiui, išreikšti jį pjūklo pavidalo funkcijų suma. Vis dėlto, praktikoje dažniausiai naudojama kosinusų ir sinusų bazė, nes būtent šios funkcijos labiausiai atitinka daugelį realiame gyvenime stebimų virpesių (muzikos instrumentų stygų virpėjimas, įvairios švytuoklės, radio bangos ir t. t.).

6.1.4 Furje koeficientų apskaičiavimas

6.1.1 skyrelyje aproksimuodami pjūklo formos signalą Furje skleidiniu, “atspėjome” jo Furje koeficientus ir beliko įsitikinti, kad skleidinio grafikas iš tiesų panašus į originalaus signalo grafiką.

Dabar nurodysime būdą kaip suskaičiuoti spektrą (Furje koeficientus) bet kokiam signalui.

Norėdami apskaičiuoti Furje skleidinio (38) koeficientą a_n (fiksuotam $n \geq 1$), padauginame abi lygybės (38) puses iš $\cos nt$ ir suintegruojame gautą išraišką intervale $-\pi \leq t \leq \pi$. Gauname:

$$\begin{aligned} \int_{-\pi}^{\pi} f(t) \cos nt \, dt &= \frac{a_0}{2} \int_{-\pi}^{\pi} \cos nt \, dt + \sum_{k=1}^{\infty} a_k \int_{-\pi}^{\pi} \cos kt \cos nt \, dt \\ &+ \sum_{k=1}^{\infty} b_k \int_{-\pi}^{\pi} \sin kt \cos nt \, dt, \quad n = 1, 2, \dots \end{aligned} \quad (39)$$

Labai panašiai elgiamės norėdami apskaičiuoti koeficientą b_n . Vienintelis skirtumas – dauginame (38) iš $\sin nt$. Suintegravę turime:

$$\begin{aligned} \int_{-\pi}^{\pi} f(t) \sin nt \, dt &= \frac{a_0}{2} \int_{-\pi}^{\pi} \sin nt \, dt + \sum_{k=1}^{\infty} a_k \int_{-\pi}^{\pi} \cos kt \sin nt \, dt \\ &+ \sum_{k=1}^{\infty} b_k \int_{-\pi}^{\pi} \sin kt \sin nt \, dt, \quad n = 1, 2, \dots \end{aligned} \quad (40)$$

Naudosimės lygybėmis (galiojančiomis su sveikaisiais $k, n = 1, 2, \dots$):

$$\int_{-\pi}^{\pi} \cos kt \cos nt \, dt = \begin{cases} 0, & k \neq n, \\ \pi, & k = n, \end{cases} \quad \int_{-\pi}^{\pi} \sin kt \sin nt \, dt = \begin{cases} 0, & k \neq n, \\ \pi, & k = n, \end{cases} \quad (41)$$

$$\int_{-\pi}^{\pi} \sin kt \cos nt \, dt = 0, \quad \int_{-\pi}^{\pi} \cos nt \, dt = 0, \quad \int_{-\pi}^{\pi} \sin nt \, dt = 0. \quad (42)$$

Šios lygybės nesunkiai patikrinamos, integralus paverčiant elementariaisiais integralais, naudojant gerai žinomas trigonometrines formules

$$\begin{aligned}\cos \alpha \cos \beta &= \frac{1}{2} (\cos(\alpha - \beta) + \cos(\alpha + \beta)), \\ \sin \alpha \sin \beta &= \frac{1}{2} (\cos(\alpha - \beta) - \cos(\alpha + \beta)), \\ \sin \alpha \cos \beta &= \frac{1}{2} (\sin(\alpha - \beta) + \sin(\alpha + \beta)).\end{aligned}$$

Remiantis formulėmis (41) ir (42), dešinėje lygybės (39) pusėje lieka tik dydis $a_n\pi$, o dešinėje lygybės (40) pusėje – tik $b_n\pi$. Taigi, gauname išraiškas Furje skleidinio (38) koeficientų radimui (vietoje indekso n vėl rašome k):

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos kt \, dt, \quad b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin kt \, dt, \quad k = 1, 2, \dots \quad (43)$$

Koeficientą a_0 išreiškiame tiesiog suintegravę skleidinį (38) intervale $-\pi \leq t \leq \pi$ ir pasinaudoję lygybėmis (42):

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \, dt. \quad (44)$$

Kadangi kai $k = 0$, $\cos kt = 1$, tai a_0 skaičiavimo formulė (44) sutampa su formule (43), su $k = 0$. Šių dviejų formulių suderinamumas ir yra motyvas, kodėl Furje skleidinyje (38) laisvasis narys apibrėžtas kaip $a_0/2$, o ne a_0 .

Beje, kalbant matematinės analizės terminais, formulės (41) ir (42) reiškia, kad trigonometrinės bazės (37) funkcijos yra tarpusavyje *ortogonalios*, t. y. bet kurių dviejų bazei priklausančių skirtingų funkcijų *skaliarinė sandauga* (apibrėžta kaip šių funkcijų integralas intervale $-\pi \leq t \leq \pi$), yra lygus nuliui.

Jeigu $f(t)$ – lyginė funkcija, tai visi $b_k \equiv 0$, taigi lyginės funkcijos Furje skleidinyje dalyvauja tik kosinusai. Taip pat, jeigu $f(t)$ – nelyginė funkcija, tuomet visi $a_k \equiv 0$, vadinasi nelyginės funkcijos Furje skleidinį nusako vien tik sinusoidžių komponentės (ši faktą jau esame patikrinę 6.1.1 skyrelyje pjūklo pavidalo signalui – nelyginei funkcijai). Ši išvada įrodoma iš formulių (43), (44), naudojantis geometrine funkcijų simetrija ordinačių ašies $t = 0$ (lyginėms funkcijoms) arba koordinačių pradžios taško $(0, 0)$ (nelyginėms funkcijoms) atžvilgiu (žr. 6.1.2 skyrelį).

Ne visoms funkcijoms $f(t)$ apibrėžtiniai integralai išraiškose (43) ir (44) gali būti apskaičiuojami tiksliai (iš analitinių formulių). Tačiau bet kokiam taikymuose sutinkamam signalui šiuos integralus galima įvertinti apytiksliai (su kaip norima maža paklaida), naudojant *skaitinio integravimo* (angliškai “numerical integration”) algoritmus, pavyzdžiui, Simpsono kvadratinę formulę (angliškai “Simpson’s rule”).

6.1.5 Kai signalas apibrėžtas ne intervale $-\pi \leq t \leq \pi$

Atkreipkime dėmesį, kad Furje skleidinyje (38) pirmoji harmonika yra periodinė (su periodu 2π) funkcija, o visos kitos harmonikos turi mažesnius periodus, bet periodą 2π taip pat. Todėl ir šiuo skleidiniu išreikšta funkcija $f(t)$ automatiškai yra periodinė su periodu 2π . Dažniausiai signalas, kurio spektrą norėtume surasti, nėra periodinis (arba jo periodas nelygus 2π), be to, jo stebėjimo laike intervalas $0 \leq t \leq T$ nesutampa su iki šiol nagrinėtu intervalu $-\pi \leq t \leq \pi$.

Tokiu atveju, galime transformuoti visas skleidinio (38) harmonikas taip, kad vietoje periodo 2π (kaip iki šiol) jos turėtų periodą, lygų T . Tuomet visą signalo stebėjimo trukmę $0 \leq t \leq T$ nagrinėsime kaip vieną transformuoto skleidinio periodą.

Taigi, Furje skleidinys, kurio vienas periodas yra intervalas $0 \leq t \leq T$ atrodo taip:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(\omega_0 k t) + \sum_{k=1}^{\infty} b_k \sin(\omega_0 k t), \quad \omega_0 = \frac{2\pi}{T}. \quad (45)$$

Analogiškai kaip 6.1.4 skyrelyje, išvedame formules Furje skleidinio (45) koeficientų apskaičiavimui:

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(\omega_0 k t) dt, \quad k = 0, 1, \dots, \quad (46)$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(\omega_0 k t) dt, \quad k = 1, 2, \dots \quad (47)$$

Šias formules galima gauti ir kintamųjų pakeitimu integraluose (43).

6.2 Furje skleidinys kompleksinių funkcijų bazėje

Eksperimentuose registruojami signalai išreiškia tam tikrų fizikinių dydžių (pavyzdžiui, įtampos, temperatūros, garso slėgio ir t.t.) kitimą. Šie dydžiai, taigi ir signalo reikšmės, kaip taisyklė yra realieji skaičiai. Todėl, jeigu nagrinėdami signalų apdorojimo algoritmus, pradėsime kalbėti apie *kompleksinius skaičius*, iš pirmo žvilgsnio tai gali atrodyti keistokai.

Vis dėlto, netrukus pamatysime, kad Furje skleidinio kompleksinių funkcijų bazėje formulė ženkliai kompaktiškesnė, palyginus su iki šiol nagrinėta (38). Be to, pasitaiko ir signalų, įgyjančių kompleksines reikšmes – juos galėsime apdoroti betarpiškai, nekonvertuodami į realiąją ir menamąją dalis. Taip pat verta pastebėti, kad jeigu programavimo kalba palaiko kompleksinių skaičių aritmetiką, algoritmo realizacija supaprastėja, jei signalas skaidomas į kompleksines komponentes, o ne į realiąsias.

Taigi, šiame skyrelyje nagrinėsime Furje skleidinį kompleksinių funkcijų bazėje.

Trumpai prisiminsime kompleksinius skaičius ir pagrindines su jais susijusias sąvokas.

6.2.1 Kompleksinių skaičių aritmetika

Kompleksinis skaičius z išreiškiamas forma

$$z = \alpha + i\beta, \quad \text{čia } i = \sqrt{-1} \text{ vadinamas menamuoju vienetu,}$$

o realieji skaičiai α ir β vadinami atitinkamai *realiąja* ir *menamąja* z dalimis ir žymima $\alpha = \operatorname{Re} z$, $\beta = \operatorname{Im} z$.

Dažnai naudojami dydžiai: kompleksinio skaičiaus z *jungtinis* \bar{z} ir *modulis* $|z|$:

$$\bar{z} = \alpha - i\beta, \quad |z| = \sqrt{\alpha^2 + \beta^2}.$$

Dviejų kompleksinių skaičių $z_1 = \alpha_1 + i\beta_1$ ir $z_2 = \alpha_2 + i\beta_2$ sudėtis, atimtis, daugyba ir dalyba skaičiuojamos pagal formules:

$$z_1 \pm z_2 = \alpha_1 \pm \alpha_2 + i(\beta_1 \pm \beta_2), \quad z_1 z_2 = \alpha_1 \alpha_2 - \beta_1 \beta_2 + i(\alpha_1 \beta_2 + \alpha_2 \beta_1),$$

$$\frac{z_1}{z_2} = \frac{\alpha_1\alpha_2 + \beta_1\beta_2 + i(\alpha_2\beta_1 - \alpha_1\beta_2)}{\alpha^2 + \beta^2}.$$

Kompleksinį skaičių z galima interpretuoti kaip tašką dvimatėje Dekarto koordinatinių sistemoje su koordinatėmis (α, β) . Tuomet, remiantis Pitagoro teorema, $|z|$ yra šio taško atstumas iki koordinatinių pradžių taško $(0, 0)$. Taigi, tą patį tašką z galima išreikšti *polinėmis* koordinatėmis $(|z|, \varphi)$, kur φ žymi kampą tarp atkarpos, jungiančios tašką z ir koordinatinių pradžių, ir Dekarto koordinatinių sistemos teigiamosios abscisų ašies dalies:

$$\alpha = |z| \cos \varphi, \quad \beta = |z| \sin \varphi, \quad -\pi < \varphi \leq \pi.$$

Kampas φ vadinamas kompleksinio skaičiaus z *argumentu*, kuris taip pat žymimas $\arg z$. Jeigu $\alpha > 0$, tuomet $\varphi = \arctan(\beta/\alpha)$, kitais atvejais argumento skaičiavimui yra šiek tiek sudėtingesnės formulės. Kompleksinio skaičiaus argumento skaičiavimui daugelis programavimo kalbų turi atan2 funkciją: $\varphi = \text{atan2}(\beta, \alpha)$, kuri duoda korektišką rezultatą bet kokiems α ir β .

Taigi kompleksinį skaičių z galima užrašyti ir taip:

$$z = |z| (\cos \varphi + i \sin \varphi), \quad -\pi < \varphi \leq \pi.$$

Nagrinėjant funkcijų $\cos \varphi$, $\sin \varphi$ ir $e^{i\varphi}$ Teiloro eilutes, išvedama vadinamoji Oilerio formulė (angliškai “Euler’s formula”)

$$e^{i\varphi} = \cos \varphi + i \sin \varphi. \quad (48)$$

Vadinasi, kompleksinis skaičius z polinėse koordinatėse įgyja kompaktišką formą:

$$z = |z| e^{i\varphi}, \quad |e^{i\varphi}| = 1, \quad -\pi < \varphi \leq \pi.$$

Ši forma patogi dauginant ir dalinant kompleksinius skaičius $z_1 = |z_1| e^{i\varphi_1}$ ir $z_2 = |z_2| e^{i\varphi_2}$:

$$z_1 z_2 = |z_1| |z_2| e^{i(\varphi_1 + \varphi_2)}, \quad \frac{z_1}{z_2} = \frac{|z_1|}{|z_2|} e^{i(\varphi_1 - \varphi_2)}.$$

6.2.2 Furje skleidinio kompleksinių funkcijų bazėje apibrėžimas

Dėl paprastumo pradžioje vėl tarkime, kad tiriamas signalas $f(t)$ apibrėžtas laiko intervale $-\pi \leq t \leq \pi$.

Nagrinėsime kompleksinių funkcijų bazę

$$\{\dots, e^{-i2t}, e^{-it}, 1, e^{it}, e^{i2t}, \dots\} = \{e^{ikt}, k = 0, \pm 1, \pm 2, \dots\}. \quad (49)$$

Remiantis Oilerio formule (48), šios funkcijos susijusios su trigonometrinės bazės (37) funkcijomis. Kadangi $|e^{ikt}| = 1$, tai k -toji kompleksinės bazės funkcija nusako taško judėjimą vienetinio spindulio apskritimu (kurio centras koordinatinių pradžių), su kampiniu greičiu k radianai per sekundę. Jeigu $k > 0$, judėjimas vyksta prieš laikrodžio rodyklę, jeigu $k < 0$ – pagal laikrodžio rodyklę. Be to, visos šios funkcijos yra periodinės su periodu 2π .

Analogiškai kaip darėme 6.1.3 skyrelyje, apibrėžiame signalo $f(t)$ *Furje skleidinį kompleksinių funkcijų bazėje*:

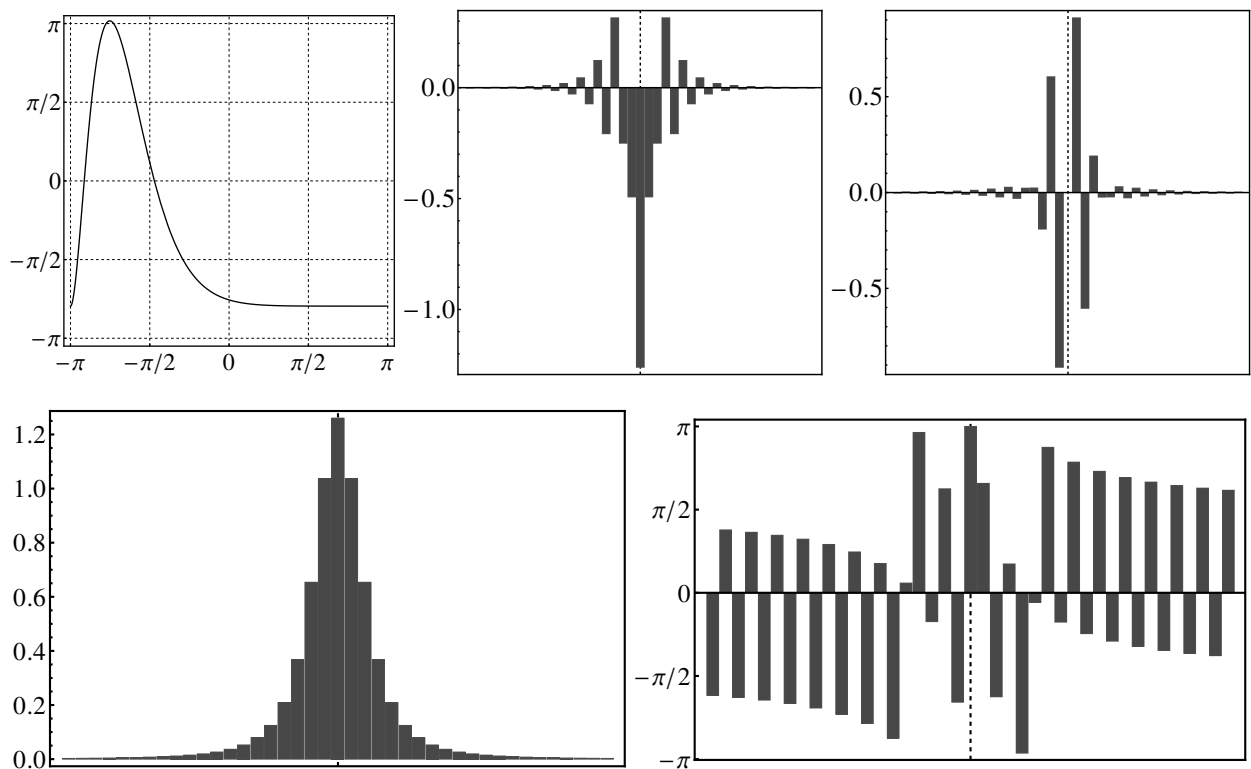
$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{ikt}. \quad (50)$$

Taip pat kaip ir anksčiau (realiuoju atveju), kompleksinio Furje skleidinio nariai, turintys vienodą dažnį, vadinami *harmonikomis*. Konstanta c_0 yra pastovioji skleidinio komponentė,

$c_1 e^{it} + c_{-1} e^{-it}$ – pirmoji harmonika, $c_2 e^{i2t} + c_{-2} e^{-i2t}$ – antroji harmonika, ir t. t.

Skleidinio (50) koeficientai c_k vadinami *kompleksiniais Furje koeficientais* arba tiesiog Furje koeficientais, jeigu iš konteksto aišku kokioje bazėje apibrėžtas skleidinys. Kaip ir realiuoju atveju, c_k priklausomybė nuo dažnio k vadinama *signalo kompleksiniu spektru*, o jeigu nekyla dviprasmiškumo, tai tiesiog signalo spektru.

Bendru atveju koeficientai c_k yra kompleksiniai skaičiai, tačiau, kaip parodysime toliau – tirdami skleidinio savybes (žr. 6.3.1 skyrelį), pats skleidinys (50) yra realioji funkcija (susumavus visus dydžius, menamoji dalis išnyksta).



59 pav.: Signalų priklausomybė nuo laiko t ir jo kompleksiniai Furje koeficientai (kompleksinis spektras) c_k , $k = -20, -19, \dots, 0, \dots, 20$. Spektrų iliustracijose punktyrine linija pažymėta Furje koeficientų simetrijos ašis $k = 0$. Iliustracija viršuje kairėje: signalo priklausomybė nuo t . Iliustracija viršuje centre: $\operatorname{Re} c_k$ priklausomybė nuo dažnio k . Iliustracija viršuje dešinėje: $\operatorname{Im} c_k$ priklausomybė nuo k . Iliustracija apačioje kairėje: *amplitudžių spektras* – $|c_k|$ priklausomybė nuo k . Iliustracija apačioje dešinėje: *fazių spektras* – $\arg c_k$ priklausomybė nuo k . Analizuojant signalus, dažniausiai skaičiuojamas tik amplitudžių spektras ir kartais fazių spektras.

Kompleksinių Furje koeficientų modulių seka (priklausomybė nuo dažnio k)

$$|c_k|, \quad k = 0, \pm 1, \pm 2, \dots,$$

vadinama *amplitudžių spektru*, o koeficientų argumentų seka

$$\arg c_k, \quad k = 0, \pm 1, \pm 2, \dots,$$

vadinama *fazių spektru*.

Kartais dar nagrinėjama koeficientų modulių kvadratų $|c_k|^2$ priklausomybė nuo dažnio k , kuri vadinama *galios spektru*.

Amplitudžių spektras parodo, kokia yra kiekvienos iš signalo komponentės (harmonikos) įtaka bendroje jų sumoje (50). Dažniausiai analizuojant signalo priklausomybę dažnių srityje (jo spektrą) tiriamas būtent amplitudžių spektras, o fazių spektrui paprastai skiriamas mažesnis dėmesys. Realiųjų ir menamųjų Furje koeficientų dalių $\operatorname{Re} c_k$ ir $\operatorname{Im} c_k$ spektrai ne tokie informatyvūs ir, kaip taisyklė, neskaičiuojami.

Kaip ir realiųjų atveju, signalą galima vienareikšmiškai nusakyti vien tik jo spektru, t. y. žinant tik jo Furje koeficientus c_k , per skleidinį (50) atstatyti signalo priklausomybę laiko srityje. Tačiau vien tik amplitudžių spektras arba vien tik fazių spektras signalo vienareikšmiškai neaprašo – reikia žinoti ir amplitudžių, ir fazių spektrą.

Kadangi kaip matysime toliau (žr. 6.3.1 skyrelį apie Furje skleidinio savybes), jeigu signalas $f(t)$ įgyja tik realiąsias reikšmes, tai koeficientai c_k ir c_{-k} yra tarpusavyje kompleksiskai jungtiniai, vadinasi, amplitudžių spektras simetriškas atžvilgiu $k = 0$, tai yra, $|c_k| = |c_{-k}|$. Tuo tarpu fazių spektras pasižymi antisimetrija: $\arg c_k = -\arg c_{-k}$.

Konkreto signalo įvairios spektro dalys pavaizduotos 59 pav.

Praktiškai apsiribojama signalo išskaidymu į baigtinį komponentių (harmonikų) skaičių. Pavyzdžiui, 59 pav. parodytas 20 harmonikų spektras. Matyti, kad didinant dažnį k , Furje koeficientai c_k artėja prie nulio, taigi visi signalo dažniai praktiškai yra baigtiniame intervale. Jeigu signalas turėtų begalinio arba labai didelio pločio spektrą (begalinį arba labai platų intervalą, kuriame stebimos ženkliai didesnės už nulį amplitudžių spektro reikšmės), tai reikštų, kad laiko t srityje šis signalas būtų itin osciliuojantis.

Beje, iš kompleksinio skaičiaus argumento (fazės) apibrėžimo (žr. 6.2.1 skyrelį) išplaukia, kad jeigu $z \rightarrow 0$, nebūtinai $\arg z \rightarrow 0$ (kad ir ant labai mažo spindulio apskritimo, taškas vis tiek gali būti bet kokiame kampas tarp 0 ir 2π radianų su absčių ašimi). Taigi fazių spektrui dažniausiai negalioja $\arg c_k \rightarrow 0$, kai $k \rightarrow \infty$, kaip ir matyti 59 pav.

Ryšys tarp realiųjų Furje koeficientų a_k , b_k ir kompleksinių Furje koeficientų c_k

Pasinaudojus Oilerio formule (48), įmanoma išreikšti skleidinį (50) per anksčiau nagrinėtas trigonometrinės bazės funkcijas:

$$f(t) = c_0 + \sum_{k=1}^{\infty} (c_k + c_{-k}) \cos kt + \sum_{k=1}^{\infty} i (c_k - c_{-k}) \sin kt.$$

Sulyginę šią išraišką su (38), gauname sąryšius tarp realiųjų Furje koeficientų a_k , b_k ir kompleksinių Furje koeficientų c_k :

$$c_0 = \frac{1}{2} a_0, \quad c_k = \frac{1}{2} (a_k - i b_k), \quad c_{-k} = \frac{1}{2} (a_k + i b_k), \quad k = 1, 2, \dots \quad (51)$$

Kompleksinių Furje koeficientų apskaičiavimas

Į sąryšius (51) įsistatę realiųjų Furje koeficientų išraiškas (43) ir (44) ir vėl pasinaudoję Oilerio formule (48), gauname formules kompleksiniams Furje koeficientams skaičiuoti:

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-ikt} dt, \quad k = 0, \pm 1, \pm 2, \dots \quad (52)$$

Priminsime, kad ši formulė galioja kai signalas, kurio skleidinio ieškome, apibrėžtas laiko intervale $-\pi \leq t \leq \pi$.

Kai signalas apibrėžtas ne intervale $-\pi \leq t \leq \pi$

Bendru atveju, jeigu signalo $f(t)$ apibrėžimo sritis yra intervalas $0 \leq t \leq T$, samprotaujant analogiškai kaip 6.1.5 skyrelyje, jo Furje skleidinys kompleksinių funkcijų bazėje yra

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{i\omega_0 k t}, \quad \omega_0 = \frac{2\pi}{T}, \quad (53)$$

nusakantis periodinę funkciją su periodu T . Signalo stebėjimo trukmė $0 \leq t \leq T$ nagrinėjama kaip vienas skleidinio (53) periodas.

Tokiu atveju signalo kompleksiniai Furje koeficientai skaičiuojami pagal formules (kurios išvedamos iš sąryšių (51) įrašius formules (46) ir (47)):

$$c_k = \frac{1}{T} \int_0^T f(t) e^{-i\omega_0 k t} dt, \quad k = 0, \pm 1, \pm 2, \dots \quad (54)$$

Kaip ir realiųjų atveju (žr. pastabą 6.1.4 skyrelio pabaigoje), programuojant signalo kompleksinio spektro radimo algoritmą, integralai (52) arba (54) kaip taisyklė skaičiuojami ne analitiškai, bet taikant skaitinio integravimo metodus.

6.3 Kompleksinio spektro savybės

6.3.1 Paprasčiausios spektro savybės

Tarkime, kad signalas $f(t)$ įgyja tik realiąsias reikšmes.

Išvardinsime keletą paprasčiausių signalo $f(t)$ kompleksinio Furje skleidinio (50) arba (53) koeficientų c_k (signalų spektro) savybių. Jos išplaukia iš sąryšių (51).

1. c_0 visada yra realusis skaičius, o kompleksiniai koeficientai c_k ir c_{-k} – tarpusavyje kompleksiskai jungtiniai: $c_{-k} = \bar{c}_k$, $k = 1, 2, \dots$
2. Iš 1. savybės bei sąryšių (51) seka, kad

$$|c_k| = |c_{-k}| = \frac{\sqrt{a_k^2 + b_k^2}}{2}, \quad \arg c_k = -\arg c_{-k}, \quad k = 1, 2, \dots$$

Taigi, amplitudžių spektras simetriškas atžvilgiu $k = 0$, o fazių spektras pasižymi antisimetrija: $\arg c_{-k} = -\arg c_k$. Šių savybių grafinį patvirtinimą galima pamatyti 59 pav.

3. Remiantis 1. savybe, Furje skleidinių (50) arba (53) menamoji dalis lygi nuliui. Vadinasi, nors atskiri skleidinio dėmenys yra kompleksiniai dydžiai, tačiau visi kartu sumoje jie apibrėžia realiąją funkciją $f(t)$ (kaip ir reikėjo tikėtis). Pavyzdžiui skleidinio (50) pirmoji harmonika gali būti užrašyta taip:

$$c_1 e^{it} + c_{-1} e^{-it} = 2 \operatorname{Re} c_1 \cos t - 2 \operatorname{Im} c_1 \sin t.$$

Šios savybės negalioja, jeigu tarp signalo $f(t)$ reikšmių yra kompleksinių skaičių su nelygia nuliui menamąja dalimi. Tokiu atveju net ir realiųjų trigonometrinių funkcijų bazėje (37) apibrėžto skleidinio (38) koeficientai a_k , b_k bus kompleksiniai skaičiai. Tačiau net ir kompleksiniams signalams galima skaičiuoti ir analizuoti spektrą pagal šiame skyriuje išdėstytus algoritmus.

6.3.2 Signalų, padauginto iš konstantos, ir dviejų signalų sumos spektras

Tarkime, q yra fiksuota realioji konstanta, o $f(t)$ signalas, kurio spektras apibrėžtas kompleksiniais Furje koeficientais c_k .

Kaip keičiasi spektras, padauginus signalą iš konstantos, tai yra, ką galima pasakyti apie signalo $q f(t)$ spektrą?

Pažymėkime signalo $q f(t)$ Furje koeficientus c'_k . Jeigu Furje koeficientai apibrėžti formulėmis (52), tai

$$c'_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} q f(t) e^{-ikt} dt = q \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-ikt} dt = q c_k, \quad k = 0, \pm 1, \pm 2, \dots$$

Tą patį dėsnį galime

$$c'_k = q c_k, \quad k = 0, \pm 1, \pm 2, \dots, \quad (55)$$

gautume ir atveju, kai Furje koeficientai apibrėžti (54) lygybėmis.

Išvada. Signalą padauginus iš konstantos, jo amplitudžių spektrą reikia padauginti iš tos pačios konstantos, o fazių spektras nesikeičia.

Taip pat ištirsime, ką galima pasakyti apie dviejų signalų sumos spektrą, jeigu žinomi kiekvieno iš signalų Furje koeficientai.

Signalų $f_1(t)$ kompleksinius Furje koeficientus pažymėkime c_{1k} , o signalo $f_2(t)$ – pažymėkime c_{2k} . Signalų sumos $f_1(t) + f_2(t)$ Furje koeficientus pažymėkime c'_k . Tuomet, jeigu Furje koeficientai apibrėžti formulėmis (52), turime

$$\begin{aligned} c'_k &= \frac{1}{2\pi} \int_{-\pi}^{\pi} (f_1(t) + f_2(t)) e^{-ikt} dt = \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f_1(t) e^{-ikt} dt + \frac{1}{2\pi} \int_{-\pi}^{\pi} f_2(t) e^{-ikt} dt = c_{1k} + c_{2k}, \quad k = 0, \pm 1, \pm 2, \dots \end{aligned}$$

Taigi (tą patį gautume ir nagrinėdami Furje koeficientus (54)),

$$c'_k = c_{1k} + c_{2k}, \quad k = 0, \pm 1, \pm 2, \dots \quad (56)$$

Išvada. Susumavus signalus, jų Furje koeficientai taip pat susisumuoja.

Abi išvadas galima apibendrinti taip: signalo Furje koeficientų radimas yra *tiesinis operatorius*

$$q_1 f_1(t) + q_2 f_2(t) \xrightarrow{\text{Furje koeficientų radimas}} q_1 c_{1k} + q_2 c_{2k}.$$

6.3.3 Signalų energijos išsilaikymas spektre (Parsevalio teorema)

Nors vadinama teorema, iš tiesų tai svarbi signalo spektro savybė, rodanti, kad *suminė signalo energija per visą stebėjimo laiką t lygi suminei signalo spektro energijai per visus dažnius k* .

Ši savybė ir vadinama *Parsevalio teorema* (angliškai “Parseval’s theorem”). Matematiškai tai išreiškiama taip:

$$\sum_{k=-\infty}^{\infty} |c_k|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(t)|^2 dt. \quad (57)$$

signalui, apibrėžtam intervale $-\pi \leq t \leq \pi$ Furje skleidiniu (50), arba

$$\sum_{k=-\infty}^{\infty} |c_k|^2 = \frac{1}{T} \int_0^T |f(t)|^2 dt. \quad (58)$$

signalui, apibrėžtam intervale $0 \leq t \leq T$ Furje skleidiniu (53).

Tarkime, kad $f(t)$ yra elektros srovės, tekančios 1Ω (Omo) varžą turinčiu laidininku, stiprumo signalas. Tada, paėmus $T = 1$ s (sekundė), dydis

$$\frac{1}{T} \int_0^T |f(t)|^2 dt$$

yra galia, reikalinga srovei palaikyti vienos sekundės laiko tarpą.

Taigi, Parsevalio teorema (58) teigia, kad tą pačią galią galima apskaičiuoti susumavus visas signalo galios spektro komponentes. Kitaip sakant, išskaidžius signalą $f(t)$ į dažnines komponentes, signalo laikinės priklausomybės suminė energija niekur neprapuola – tiesiog pasiskirsto per harmonikų galią apibrėžiančius koeficientus.

Parsevalio teoremos įrodymas nesudėtingas – tereikia Furje skleidinį padauginti iš sau jungtinio ir apskaičiuoti sandaugos integralą. Pavyzdžiui, skleidinio (50) atveju turime (remiantis kompleksinių skaičių savybe $z\bar{z} = |z|^2$)

$$|f(t)|^2 = f(t)\overline{f(t)} = \sum_{k=-\infty}^{\infty} c_k e^{ikt} \sum_{m=-\infty}^{\infty} \bar{c}_m e^{-imt} = \sum_{k,m=-\infty}^{\infty} c_k \bar{c}_m e^{i(k-m)t}$$

Kadangi elementarus integralas

$$\int_{-\pi}^{\pi} e^{i(k-m)t} dt = \begin{cases} 0, & k \neq m, \\ 2\pi, & k = m, \end{cases}$$

suintegravus iš dvigubos sumos lieka tik vienguba:

$$\int_{-\pi}^{\pi} |f(t)|^2 dt = 2\pi \sum_{k=-\infty}^{\infty} |c_k|^2.$$

Taigi, gavome lygybę (57).

6.3.4 Baigtinio Furje skleidinio paklaida. Gibbs'o efektas

Programuodami signalo spektro radimo algoritmą, negalime (o ir nėra prasmės) skaičiuoti be galo daug Furje koeficientų. Taigi, praktiškai išskaidome signalą $f(t)$ į baigtinį kiekį M harmonikų (panašiai, kaip darėme šio skyriaus pradžioje, formulėje (36)):

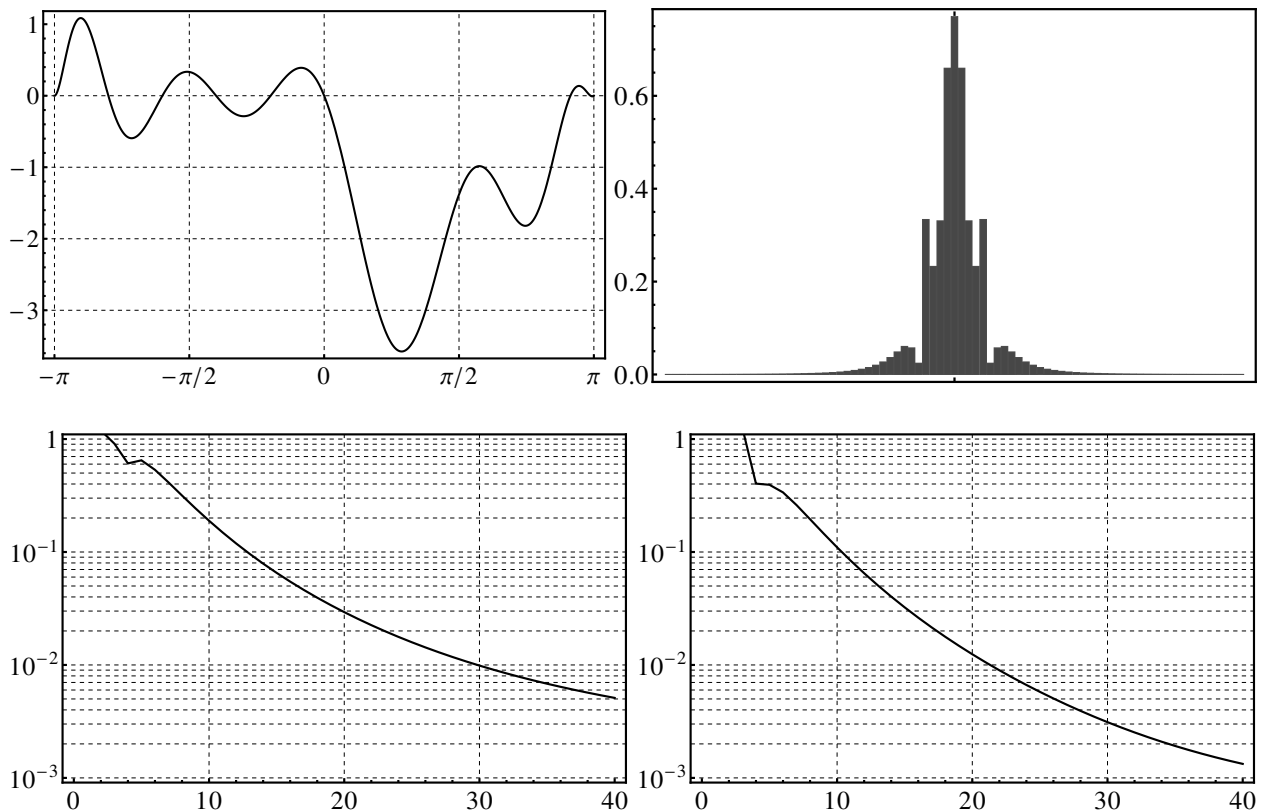
$$f(t) \approx \sum_{k=-M}^M c_k e^{ikt}.$$

Panagrinėkime kaip elgiasi Furje skleidinio paklaida, priklausomai nuo M . Natūralu tikėtis, kad didinant M (įskaitant vis daugiau harmonikų į signalą aproksimuojantį skleidinį), paklaida turėtų artėti (konverguoti) į nulį.

Dėl paprastumo vėl apsiribokime atveju kai $-\pi \leq t \leq \pi$, t. y. Furje skleidinio (50) koeficientai apibrėžti lygybėmis (52). Furje skleidinio (53) atvejis su koeficientais (54) gaunamas tiesiškai transformuojant intervalą $-\pi \leq t \leq \pi$ į $0 \leq t \leq T$, taigi nieko naujo nepastebėtume.

Apibrėšime dvi paklaidas (tiksliai matematiškai sakant, paklaidą dviejose skirtingose normose). Pirmoji paklaida $\|\varepsilon(M)\|_C$ (vadinama paklaida C normoje) natūrali – įvertina maksimalų skirtumą (moduliu) tarp dviejų lyginamų dydžių (signalų ir jo baigtinio skleidinio):

$$\|\varepsilon(M)\|_C = \max_{-\pi \leq t \leq \pi} \left| f(t) - \sum_{k=-M}^M c_k e^{ikt} \right|. \quad (59)$$

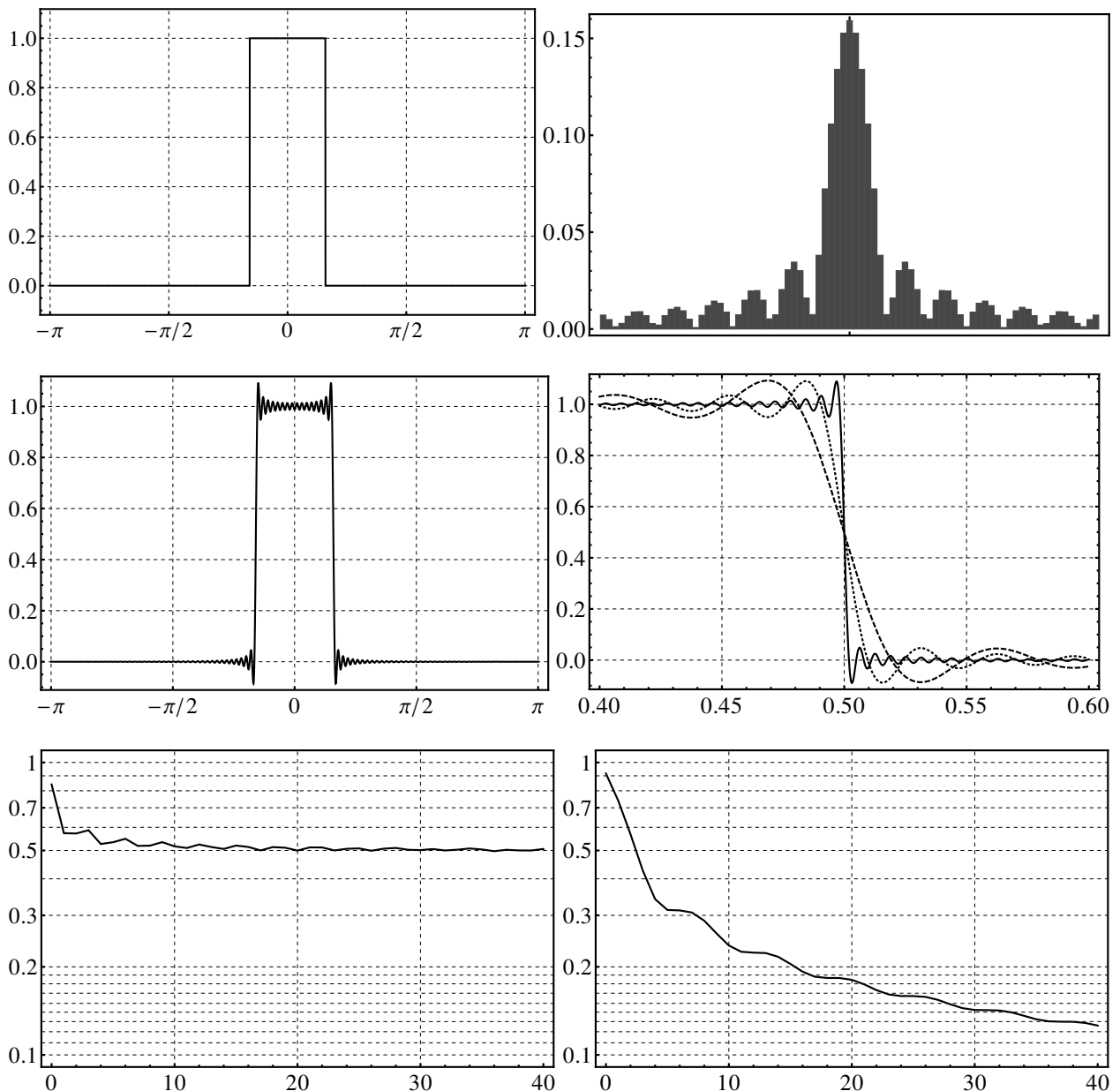


60 pav.: Iliustracija viršuje kairėje: *glodaus* signalo $f(t)$ priklausomybė nuo laiko t . Iliustracija viršuje dešinėje: signalo *amplitudžių spektras* – $|c_k|$ priklausomybė nuo k . Iliustracija apačioje kairėje: paklaidos $\|\varepsilon(M)\|_C$ priklausomybė nuo įskaitomų į Furje skleidinį harmonikų skaičiaus M (ordinačių ašyje – logaritminis mastelis). Iliustracija apačioje dešinėje: paklaidos $\|\varepsilon(M)\|_{L_2}$ priklausomybė nuo M (ordinačių ašyje – logaritminis mastelis). Paklaidos rodo, kad didinant M , Furje skleidinio grafikas vis labiau ir labiau sutampa su signalo grafiku.

Antroji paklaida $\|\varepsilon(M)\|_{L_2}$ (vadinama paklaida L_2 normoje) šiek tiek “silpnesnė” ir panaši į lygybės (57) dešinėje esantį integralą:

$$\|\varepsilon(M)\|_{L_2} = \sqrt{\int_{-\pi}^{\pi} \left| f(t) - \sum_{k=-M}^M c_k e^{ikt} \right|^2 dt}. \quad (60)$$

Taigi, prisiminę 6.3.3 skyrelį, galime pasakyti, kad tai iš esmės yra suminė paklaidos energija (per visą laiko intervalą). Geometriškai paklaida $\|\varepsilon(M)\|_{L_2}$ reiškia plotą tarp paklaidos kvadrato kreivės ir abscisių ašies. Vadinasi, jos mažumas reiškia, kad signalas ir jo Furje skleidinys artimi *beveik visur* intervale $-\pi \leq t \leq \pi$, tačiau keletui t reikšmių gali ženkiai skirtis (įsivaizduokite paklaidos kreivę, kuri lygi nuliui beveik visur, bet turi keletą labai plonų, bet gana aukštų “smailių” – plotas po jos grafiku bus artimas nuliui).



61 pav.: *Gibbs'o efekto demonstracija*. Iliustracijos viršuje: *turinčio trūkius* signalo $f(t)$ priklausomybė nuo laiko t (kairėje); signalo *amplitudžių spektras* – $|c_k|$ priklausomybė nuo k (dešinėje). Iliustracijos viduryje: Furje skleidinio su $M = 100$ priklausomybė nuo t (kairėje); Furje skleidinių su $M = 100$ (brūkšninė kreivė), $M = 200$ (taškinė kreivė) ir $M = 1000$ (išstinė kreivė) priklausomybės nuo t fragmentas (dešinėje). Iliustracijos apačioje: paklaidos $\|\varepsilon(M)\|_C$ priklausomybė nuo įskaitomų į Furje skleidinį harmonikų skaičiaus M , ordinačių ašyje – logaritminis mastelis (kairėje); paklaidos $\|\varepsilon(M)\|_{L_2}$ priklausomybė nuo M , ordinačių ašyje – logaritminis mastelis (dešinėje). Gibbs'o efektas rodo, kad didinant harmonikų skaičių M , Furje skleidinys *signalų trūkio taškų aplinkose neartėja* (nekonverguoja) į tiksliai signalo reikšmes. Todėl paklaida $\|\varepsilon(M)\|_C \not\rightarrow 0$, kai $M \rightarrow \infty$. Tačiau visur kitur (kur signalas glodus), skleidinys puikiai aproksimuoja signalą, taigi paklaida $\|\varepsilon(M)\|_{L_2} \rightarrow 0$, kai $M \rightarrow \infty$.

Paklaidų tyrimo kompiuterinio eksperimento būdu rezultatai pateikti 60 pav. ir 61 pav..

Iliustracijoje 60 pav. ištyrėme kaip tiksliai Furje skleidinys aproksimuoja *glodų* (be trūkių) signalą. Kaip ir buvo galima tikėtis, matyti, kad abi paklaidos $\|\varepsilon(M)\|_C$ ir $\|\varepsilon(M)\|_{L_2}$ artėja į nulį, kai M didėja. Pavyzdžiui, kai $M = 30$, baigtinis Furje skleidinys sudarytas iš 30 harmonikų (komponenčių) visuose

intervalo $-\pi \leq t \leq \pi$ taškuose yra absoliutiniu tikslumu nutolęs nuo signalo ne daugiau kaip per $\|\varepsilon(30)\|_C \approx 0.01$. Padidinus M , tikslumas dar padidėja. Tai reiškia, kad glodaus signalo atveju Furje skleidinio algoritmas dirba be priekaištų.

Jeigu signalas nėra glodus (tolydus) – turi trūkių, Furje skleidinys taip pat iš esmės atkartoja signalo formą, tačiau *signalų trūkio taškų aplinkose* paklaida *neartėja prie nulio*, žr. 61 pav. Ši Furje skleidinio savybė vadinama *Gibbs'o efektu* arba *Gibbs'o reiškiniu* (angliškai “Gibbs phenomenon”). Matyti (61 pav., iliustracija viduryje dešinėje), kad arti trūkio taško net $M = 1000$ harmonikų turinčio skleidinio maksimali svyravimo amplitudė yra to paties dydžio kaip ir atvejais $M = 100$, $M = 200$. Taigi, kai $M \rightarrow \infty$, paklaida artėja prie nulio visiems t , išskyrus signalo trūkio taškų aplinkas. Tai reiškia, kad paklaida $\|\varepsilon(M)\|_C$ – maksimalus (per visą laiko t intervalą) skirtumas tarp signalo ir jo skleidinio, mažėja tik iki tam tikros ribos. Toliau didinant M , paklaidos $\|\varepsilon(M)\|_C$ reikšmės nebemažėja. Tačiau visur kitur (kur signalas trūkių neturi), jokių problemų nėra, paklaida didėjant M gali tapti kaip norima maža. Todėl integralinė paklaida $\|\varepsilon(M)\|_{L_2} \rightarrow 0$, kai $M \rightarrow \infty$.

Gibbs'o efektas taip pat matomas ir šio skyriaus pradžioje nagrinėtame pavyzdyje, žr. 56 pav.

7 Diskrečioji Furje transformacija. Furje filtras

7.1 Skaitmeninio signalo spektras. Diskrečioji Furje transformacija

Furje skleidiniu, kurį nagrinėjome 6 skyriuje, signalą išskaidę į dažnines komponentes (harmonikas), galime pastebėti signalo savybes, kurios nematomos analizuojant jo priklausomybę laiko skalėje, taip pat jį apdoroti.

Pavyzdžiui, siekiant iš signalo pašalinti triukšmus (tarkime, įvairius cypimus ir traškesius iš garso įrašo), dažnai apdorojamas ne pats signalas, bet jo spektras (dažninių komponentių visuma). Triukšmams kaip taisyklė būdingi kitokie dažniai nei prasmingai signalo informacijai (pavyzdžiui, žmogaus balsui, muzikos instrumentų skambėjimui). Kadangi signalą galima vienareikšmiškai atstatyti iš jo spektro, prieš atstatydami signalą pašaliname iš spektro harmonikas, kurių dažnis būdingas triukšmams. Tokiu būdu gauname filtruotą signalą. Žinoma, gali būti filtruojami ne tik garso įrašai.

7.1.1 Diskrečiosios Furje transformacijos (ir jai atvirkštinės) apibrėžimai

6 skyriuje Furje skleidinį apibrėžėme tolydaus laiko signalams $f(t)$. Šiame skyriuje kalbėsime apie skaitmeninius signalus, nusakytus jų reikšmėmis

$$f_0, f_1, \dots, f_{N-1},$$

ir tokių signalų spektro skaičiavimo algoritmus.

Praktiškai visi eksperimentinės kilmės duomenys aprašomi skaitmeniniais (diskretaus laiko), o ne analoginiais (tolydaus laiko) signalais.

Laikysime, kad $t = 0$ yra signalo stebėjimo pradžia, o eksperimentiniai duomenys surinkti naudojant pastovų laiko diskretizacijos žingsnį Δt tarp dviejų iš eilės einančių signalo reikšmių matavimo momentų:

$$f_0 = f(0), f_1 = f(\Delta t), f_2 = f(2\Delta t), \dots, f_{N-1} = f((N-1)\Delta t).$$

Apie skaitmeninių signalų diskretizacijos žingsnio pasirinkimą (kaip dažnai eksperimente matuoti mus dominančio proceso reikšmes) kalbėjome 1.2.4 skyrelyje.

Prisiminkime kaip 6 skyriuje apibrėžėme tolydaus laiko signalo $f(t)$ Furje skleidinį harmonikomis, turinčiomis periodą 2π :

$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{ikt}, \quad c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-ikt} dt = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-ikt} dt,$$

žr. formules (50) ir (52). Integravimas nuo $-\pi$ iki π gali būti pakeistas integravimu nuo 0 iki 2π dėl integruojamų funkcijų $f(t)$ ir e^{-ikt} periodiškumo.

Panašiai kaip 6 skyriuje, laikysime, kad skaitmeninio signalo reikšmių f_j registracijos laikai $t_j = j\Delta t$ išsidėstę intervale $t_j \in [t_0, t_N] = [0, 2\pi]$, taigi

$$t_j = j\Delta t = j \frac{2\pi}{N}, \quad f_j = f(t_j), \quad j = 0, 1, \dots, N-1, \quad \Delta t = \frac{2\pi}{N}.$$

Tai yra, signalo tyrimo trukmė atitinka vieną harmonikų periodą, lygų 2π . Vadinasi, jeigu prijungtume signalo reikšmę f_N , ji dėl periodiškumo būtų lygi pirmajai registruoto signalo reikšmei f_0 : $f_N = f_0$.

Šiaip jau signalas neturi būti periodinis (o praktiškai ir nebūna). Tiesiog laikome, kad visi duomenų surinkimo laikai yra viename harmonikų periode $[0, 2\pi]$. Jeigu reali signalo stebėjimo trukmė

$t_j \in [0, T]$ nesutampa su intervalu $[0, 2\pi]$, transformuojame signalo reikšmių registravimo laikus t_j , padaugindami juos iš konstantos $2\pi/T$. Faktiškai tokia transformacija reiškia tik dažnių, kuriuos atitiks k -tasis Furje koeficientas c_k , pasislinkimą. Detaliau šį klausimą aptarsime 7.1.3 skyrelyje.

Nagrinėdami skaitmeninius signalus, turime baigtinį kiekį signalo reikšmių, lygų N . Įstatę į Furje skleidinio išraišką diskretaus laiko reikšmes t_0, t_1, \dots, t_{N-1} (prie kurių žinome signalo vertes), gausime N lygčių, skirtų nežinomųjų – Furje koeficientų c_k radimui. Vadinasi, norint vienareikšmiškai išspręsti gautą lygčių sistemą, nežinomųjų skaičius taip pat turi būti lygus N . Taigi, diskretaus signalo atveju, apibrėžiame jo Furje skleidinį kaip N komponentų sumą su koeficientais c_0, c_1, \dots, c_{N-1} :

$$f_j = \sum_{k=0}^{N-1} c_k e^{ikt_j}, \quad j = 0, 1, \dots, N-1. \quad (61)$$

Išraiškos (61) ir yra N lygčių sistema, nežinomųjų c_0, c_1, \dots, c_{N-1} (diskretaus signalo spektro) radimui.

Kadangi $t_j = j2\pi/N$, šios lygčių sistemos sprendinį įmanoma užrašyti kompaktiškais formulėmis. Išvesime jas.

Norėdami apskaičiuoti Furje koeficientą c_n (fiksotam $n = 0, 1, \dots, N-1$), padauginame abi lygybės (61) puses iš e^{-int_j} ir susumuokime pagal j nuo 0 iki $N-1$. Turėsime

$$\sum_{j=0}^{N-1} f_j e^{-int_j} = \sum_{k=0}^{N-1} c_k \sum_{j=0}^{N-1} e^{i(k-n)t_j}. \quad (62)$$

Irašę $t_j = j2\pi/N$, apskaičiuojame sumą

$$\sum_{j=0}^{N-1} e^{i(k-n)t_j} = \sum_{j=0}^{N-1} e^{i(k-n)j\frac{2\pi}{N}} = \begin{cases} 0, & k \neq n, \\ N, & k = n. \end{cases} \quad (63)$$

Pastaroji lygybė triviali kai $k = n$ (sumuojame vieneta nuo 0 iki $N-1$, taigi N kartų). Jeigu $m = k - n \neq 0$, griežtai matematiškai įrodoma, kad sudėję visus narius gausime nulį, nagrinėjant geometrinės progresijos a_j sumos formulę:

$$a_0 = 1, \quad a_j = a_0 r^j, \quad r = e^{i\frac{2\pi m}{N}}, \quad \sum_{j=0}^{N-1} a_j = \frac{a_0(1-r^N)}{1-r} = 0.$$

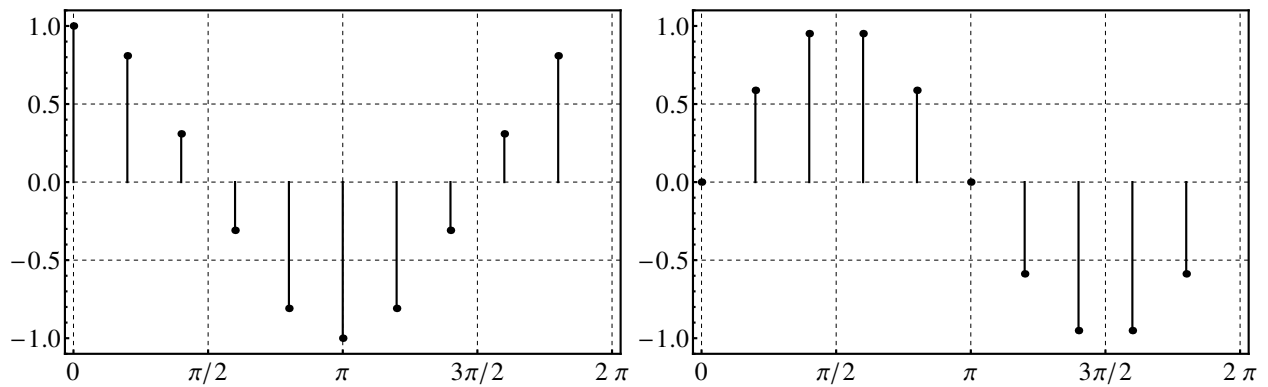
Geometriškai įsitikinti lygybės (64) teisingumu galima remiantis Oilerio formule (48)

$$\sum_{j=0}^{N-1} e^{ij\frac{2\pi m}{N}} = \sum_{j=0}^{N-1} \cos\left(j\frac{2\pi m}{N}\right) + i \sum_{j=0}^{N-1} \sin\left(j\frac{2\pi m}{N}\right) = 0. \quad (64)$$

Ši suma lygi nuliui, nes, pavyzdžiui, kosinusų sumoje yra tiek teigiamų, tiek neigiamų reikšmių – susumavus jos viena kitą naikina. Analogiškai ir sinusų sumoje. Grafinė to demonstracija pateikta 62 pav. Jeigu N nelyginis, tarp teigiamų ir neigiamų kosinusų reikšmių nėra tiesioginės simetrijos (kaip 62 pav.), tačiau net ir tuo atveju galima pastebėti, kad visų jų suma lygi nuliui ir lygybė (64) galioja. Taip pat tai teisinga su bet koku sveiku $m \neq 0$, nes $2\pi m$ visada yra tiek kosinuso, tiek sinuso periodas.

Grįžkime prie lygčių (62). Dėl savybės (63), gauname

$$\sum_{j=0}^{N-1} f_j e^{-i\frac{2\pi}{N}jn} = Nc_n,$$



62 pav.: Demonstracija, kad sumoje (64) kosinusai ir sinusai įgyja tiek teigiamas, tiek neigiamas reikšmes, kurios susumavus naikina viena kitą. Atvejis $m = 1$, $N = 10$. Iliustracija kairėje: reikšmės $\cos\left(j \frac{2\pi m}{N}\right)$, kai $j = 0, 1, \dots, N - 1$. Iliustracija dešinėje: reikšmės $\sin\left(j \frac{2\pi m}{N}\right)$, kai $j = 0, 1, \dots, N - 1$.

arba (vietoje indekso n vėl rašome k),

$$c_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i \frac{2\pi}{N} j k}, \quad k = 0, 1, \dots, N-1. \quad (65)$$

Formulė (65), iš kurios apskaičiuojame diskretaus (skaitmeninio) signalo $f = (f_0, f_1, \dots, f_{N-1})$ spektrą – Furje koeficientus c_k , vadinama *diskrečiąja Furje transformacija* (angliškai “discrete Fourier transform”).

Išraiškos, leidžiančios iš turimo spektro c_k atstatyti skaitmeninį signalą $f = (f_0, f_1, \dots, f_{N-1})$, gaunamos į lygybes (61) įrašius $t_j = j2\pi/N$:

$$f_j = \sum_{k=0}^{N-1} c_k e^{i \frac{2\pi}{N} j k}, \quad j = 0, 1, \dots, N-1, \quad (66)$$

ir vadinamos *atvirkštine diskrečiąja Furje transformacija* (angliškai “inverse discrete Fourier transform”).

Alternatyvūs diskrečiosios Furje transformacijos (ir jai atvirkštinės) apibrėžimai

Literatūroje kartais diskrečioji Furje transformacija apibrėžiama formule

$$c_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} f_j e^{-i \frac{2\pi}{N} j k}, \quad k = 0, 1, \dots, N-1.$$

Tuomet atvirkštinė diskrečioji Furje transformacija yra

$$f_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} c_k e^{i \frac{2\pi}{N} j k}, \quad j = 0, 1, \dots, N-1.$$

Nuo mūsų apibrėžtų formulių (65) ir (66) jos skiriasi tik daugikliu $1/\sqrt{N}$, bet turi simetrinį pavidalą.

Dar sutinkamas variantas: išraiška (65) be daugiklio $1/N$, o formulė (66) su daugikliu $1/N$.

Diskrečioji Furje transformacija (ir jai atvirkštinė), nenaudojant kompleksinių skaičių

Jeigu programavimo kalba nepalaiko kompleksinių skaičių aritmetikos, realizuojant diskrečiosios Furje transformacijos (65) ir jai atvirkštinės (66) algoritmus, teks abi formules išreikšti realiaisiais skaičiais.

Užrašysime šias išraiškas. Pažymėję realiąją Furje koeficiento c_k dalį A_k , o menamąją B_k , t. y.

$$c_k = A_k + i B_k,$$

įstatę į lygybę (65) Oilerio formulę (48), gauname diskrečiosios Furje transformacijos formules

$$A_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \cos\left(\frac{2\pi}{N} jk\right),$$

$$B_k = -\frac{1}{N} \sum_{j=0}^{N-1} f_j \sin\left(\frac{2\pi}{N} jk\right), \quad k = 0, 1, \dots, N-1.$$

Atvirkštinės diskrečiosios Furje transformacijos formules išreiškiame iš lygybių (66):

$$f_j = \sum_{k=0}^{N-1} (A_k + i B_k) \left(\cos\left(\frac{2\pi}{N} jk\right) + i \sin\left(\frac{2\pi}{N} jk\right) \right), \quad j = 0, 1, \dots, N-1.$$

Jeigu signalo reikšmės f_j yra realieji skaičiai, tai pastarosios išraiškos menamoji dalis bus lygi nuliui. Vadinasi, atvirkštinė diskrečioji Furje transformacija turi pavidalą

$$f_j = \sum_{k=0}^{N-1} \left(A_k \cos\left(\frac{2\pi}{N} jk\right) - B_k \sin\left(\frac{2\pi}{N} jk\right) \right), \quad j = 0, 1, \dots, N-1.$$

Panašumas tarp diskrečiosios Furje transformacijos ir jai atvirkštinės

Šiame skyrelyje pateiktos formulės suteikia galimybes kompiuteriu vykdyti skaitmeninių signalų Furje analizę (spekro tyrimus) ir apdorojimą.

Atkreipkime dėmesį, kiek panašios yra diskrečiosios Furje transformacijos (65) ir atvirkštinės diskrečiosios Furje transformacijos (66) išraiškos. Prieš formulę (65) stovi daugiklis $1/N$, o prieš (66) jo nėra, taip pat lygybėje (65) eksponentės rodiklis su minuso ženklu, o (66) – su plusu. Tačiau, išskyrus šiuos neesminius skirtumus, savo forma abi šios išraiškos absoliučiai vienodos.

Vadinasi, iš esmės tiek diskrečiąją Furje transformaciją, tiek atvirkštinę diskrečiąją Furje transformaciją galima skaičiuoti viena ir ta pačia kompiuterine paprograme (procedūra). Skaičiuojant diskrečiąją Furje transformaciją, įvedimo duomenimis bus diskrečios signalo reikšmės, o vykdant atvirkštinę diskrečiąją Furje transformaciją – Furje koeficientai. Taip pat, priklausomai nuo atliekamo veiksmo, keisis eksponentės rodiklio ženklas bei gaunamus rezultatus reikės padalinti iš duomenų skaičiaus N .

7.1.2 Diskrečiosios Furje transformacijos savybės

Diskrečioji Furje transformacija pasižymi panašiomis savybėmis, kaip tolydaus laiko signalo spektras. Šias savybes esame aptarę skyreliuose 6.3.1, 6.3.2 ir 6.3.3.

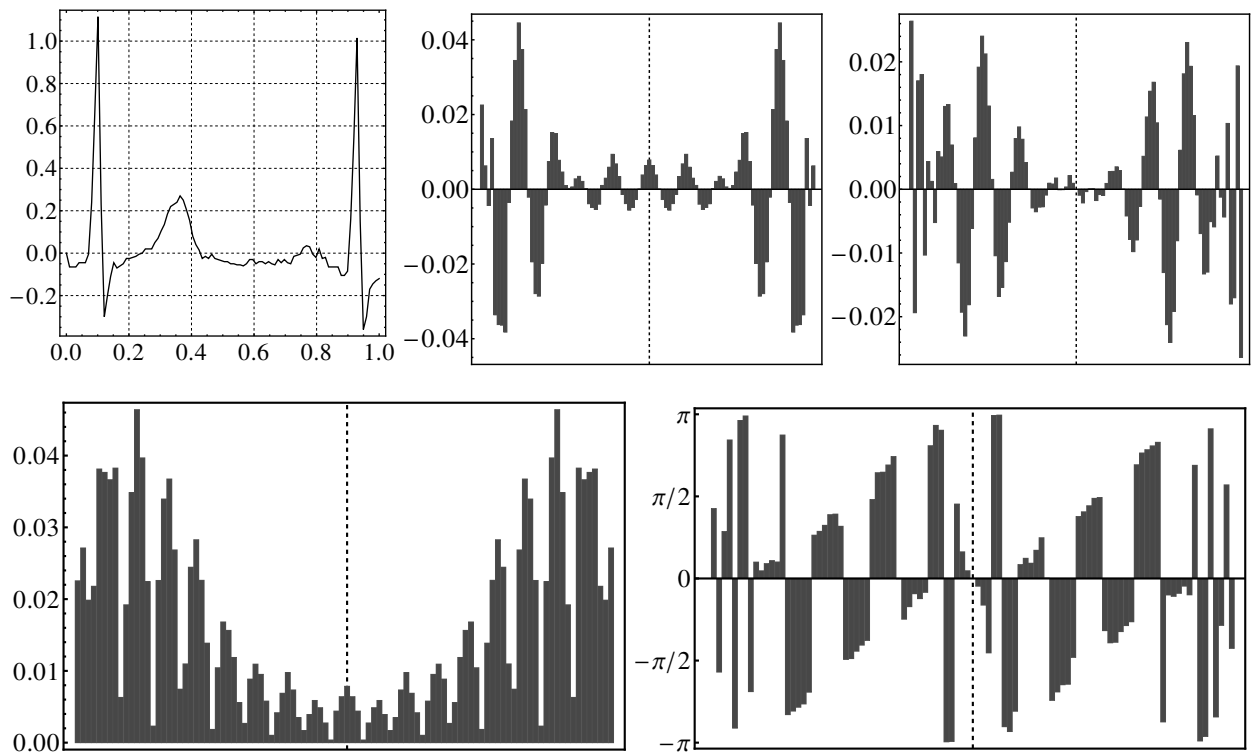
Išvardinsime diskrečiosios Furje transformacijos (65) savybes.

Spektro simetrija

Tarkime, kad visos skaitmeninio signalo $f = (f_0, f_1, \dots, f_{N-1})$ reikšmės yra realieji skaičiai. Tuomet c_0 yra realusis skaičius, lygus skaitmeninio signalo reikšmių aritmetiniam vidurkiui:

$$c_0 = \frac{1}{N} \sum_{j=0}^{N-1} f_j.$$

Tuo įsitikiname formulėje (65) parinkę $k = 0$.



63 pav.: Skaitmeninio signalo (elektrokardiogramos V4 kanalo fragmento) priklausomybė nuo diskretaus laiko t_j ir jo diskrečioji Furje transformacija c_k , $k = 0, 1, \dots, N-1$, $N = 100$. Spektro iliustracijose punktyrine linija pažymėta Furje koeficientų simetrijos ašis $k = N/2$. Iliustracija viršuje kairėje: signalo priklausomybė nuo t . Iliustracija viršuje centre: $\operatorname{Re} c_k$ priklausomybė nuo dažnio k . Iliustracija viršuje dešinėje: $\operatorname{Im} c_k$ priklausomybė nuo k . Iliustracija apačioje kairėje: *amplitudžių spektras* – $|c_k|$ priklausomybė nuo k . Iliustracija apačioje dešinėje: *fazių spektras* – $\arg c_k$ priklausomybė nuo k . Analizuojant signalus, dažniausiai skaičiuojamas tik amplitudžių spektras ir kartais fazių spektras.

Visi kiti Furje koeficientai bendru atveju gali būti kompleksiniai skaičiai, tačiau galioja *simetrijos* savybė:

$$c_{N-k} = \bar{c}_k, \quad k = 1, 2, \dots, N-1. \quad (67)$$

Priminsime, kad \bar{c}_k žymi skaičiaus c_k kompleksiskai jungtinį.

Simetrijos savybė (67) reiškia, kad *amplitudžių spektras* $|c_k|$ simetriškas atžvilgiu $k = N/2$, o *fazių spektras* pasižymi antisimetrija:

$$|c_{N-k}| = |c_k|, \quad \arg c_{N-k} = -\arg c_k, \quad k = 1, 2, \dots, N-1.$$

Tai yra, $|c_{N-1}| = |c_1|$, $|c_{N-2}| = |c_2|$ ir taip toliau.

Taip pat simetrija atžvilgiu $k = N/2$ pasižymi realiosios c_k dalys $\operatorname{Re} c_k$, tuo tarpu menamosios dalys $\operatorname{Im} c_k$ – antisimetriškos.

Šių savybių grafinį patvirtinimą galima pamatyti 63 pav.

Spektro simetrijos savybė (67) įrodoma tiesiogiai iš koeficientų c_k apibrėžimo (65), bei remiantis tuo, kad $e^{-i2\pi j} = 1$:

$$c_{N-k} = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i\frac{2\pi}{N}(N-k)j} = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i2\pi j} e^{i\frac{2\pi}{N}kj} = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{i\frac{2\pi}{N}kj} = \bar{c}_k.$$

Spektro simetrija (67) gali negalioti, jeigu tarp signalo reikšmių f_j yra kompleksinių skaičių su nelygia nuliui menamąja dalimi.

Diskrečioji Furje transformacija – tiesinis operatorius

Diskrečioji Furje transformacija pasižymi *tiesiškumo* savybe, analogiškai kaip ir tolydaus laiko signalų atveju, žr. 6.3.2 skyrelį.

Nagrinėkime du skaitmeninius signalus $f = (f_0, f_1, \dots, f_{N-1})$ ir $h = (h_0, h_1, \dots, h_{N-1})$. Signalų f Furje koeficientus pažymėkime c_{fk} , o signalo h Furje koeficientus pažymėkime c_{hk} .

Taip pat tarkime, kad q_1 ir q_2 yra fiksuotos konstantos.

Signalų tiesinės kombinacijos $q_1 f + q_2 h$ Furje koeficientus pažymėkime c'_k . Tuomet galioja

$$c'_k = q_1 c_{fk} + q_2 c_{hk}, \quad k = 0, 1, \dots, N-1. \quad (68)$$

Lygybė (68) teisinga nes, remiantis diskrečiosios Furje transformacijos formule (65), gauname

$$\begin{aligned} c'_k &= \frac{1}{N} \sum_{j=0}^{N-1} (q_1 f_j + q_2 h_j) e^{-i\frac{2\pi}{N}jk} = q_1 \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i\frac{2\pi}{N}jk} + q_2 \frac{1}{N} \sum_{j=0}^{N-1} h_j e^{-i\frac{2\pi}{N}jk} = \\ &= q_1 c_{fk} + q_2 c_{hk}. \end{aligned}$$

Vadinasi, diskrečioji Furje transformacija yra *tiesinis operatorius*:

$$q_1 f + q_2 h \xrightarrow{\text{diskrečioji Furje transformacija}} q_1 c_{fk} + q_2 c_{hk}.$$

Atskiru atveju, parinę $q_2 = 0$, gauname tokią išvadą: skaitmeninio signalo reikšmės padauginus iš realiosios konstantos, jo amplitudžių spektrą reikia padauginti iš tos pačios konstantos, o fazių spektras nesikeičia.

Signalų energijos išsilaikymas spektre

Galioja analogiška 6.3.3 skyrelyje nagrinėtai (tolydaus laiko signalams) savybė, teigianti, kad skaitmeninį signalą $f = (f_0, f_1, \dots, f_{N-1})$ diskrečiosios Furje transformacijos būdu išskaidžius į spektrą, signalo reikšmių suminė energija (galia) niekur neprapuola – tiesiog pasiskirsto per galios spektro koeficientus $|c_k|^2$.

Matematiškai ši savybė vadinama Parsevalio teorema diskrečiajai Furje transformacijai (65) ir užrašoma taip:

$$\sum_{k=0}^{N-1} |c_k|^2 = \frac{1}{N} \sum_{j=0}^{N-1} |f_j|^2. \quad (69)$$

Lygybė (69) įrodoma padauginus f_j apibrėžiančias išraiškas (66) iš joms kompleksiskai jungtinių (kadangi $|f_j|^2 = f_j \bar{f}_j$), susumavus gautus reiškinius pagal j nuo 0 iki $N-1$ ir pasinaudojus savybe (63).

7.1.3 Spektro ir realių dažnių ryšys

Gali kilti klausimas: kokį realų dažnį (kiek hercų) atitinka kiekvienas Furje koeficientas c_k ?

Pirmiausiai nustatykime kiek harmonikų (skirtingo dažnio komponentų) yra (66) skleidinyje. Kadangi, kaip jau esame nustatę tirdami spektro simetriją,

$$e^{i\frac{2\pi}{N}(N-k)j} = e^{i2\pi j} e^{-i\frac{2\pi}{N}kj} = e^{-i\frac{2\pi}{N}kj},$$

vadinasi, prie koeficientų c_k ir c_{N-k} stovi vienodą dažnį turinčios funkcijos. Taigi, konstanta c_0 yra pastovioji skleidinio komponentė,

$$\begin{aligned} c_1 e^{i\frac{2\pi}{N}j} + c_{N-1} e^{i\frac{2\pi}{N}(N-1)j} & \text{– pirmoji harmonika,} \\ c_2 e^{i\frac{2\pi}{N}2j} + c_{N-2} e^{i\frac{2\pi}{N}(N-2)j} & \text{– antroji harmonika, ir t. t.} \end{aligned}$$

Todėl, skirtingus dažnius atitinka tik Furje koeficientai c_0 (pastovioji komponentė, atitinka nulinį dažnį), pora (c_1, c_{N-1}) , pora (c_2, c_{N-2}) , ..., pora $(c_{N/2-1}, c_{N/2+1})$ ir vienas koeficientas $c_{N/2}$ (tarkime, kad N – lyginis skaičius, kitu atveju irgi turėsime panašiai).

Kadangi skleidinio (66) harmonikos e^{ikt_j} ($t_j = j2\pi/N$) yra 2π periodinės funkcijos, vadinasi abu koeficientai c_k ir c_{N-k} atitinka $k/(2\pi)$ Hz (hercų – virpesių per sekundę) dažnį signale, čia $k = 0, 1, \dots, N/2$.

Tačiau, nagrinėdami diskrečiąją Furje transformaciją, esame padarę prielaidą, kad signalo stebėjimo trukmė yra 2π ilgio intervalas: $0 \leq t \leq 2\pi$.

Jeigu tikroji signalo stebėjimo trukmė yra intervalas $0 \leq t \leq T$, vadinasi iš tiesų signalas svyruoja $T/(2\pi)$ kartų platesnėje laiko skalėje nei 2π ilgio intervalas.

Tai reiškia, kad tikrieji dažniai, kuriuos atitinka c_k , yra $T/(2\pi)$ kartų mažesni.

Išvada. Tarkime, turime realius duomenis, registruotus su diskretizacijos žingsniu Δt (kiek sekundžių skiria du iš eilės einančių signalo reikšmių registravimo momentus), taigi, visi jų registracijos laikai patenka į intervalą $[0, T]$, $T = N\Delta t$. Tuomet, pagal diskrečiąją Furje transformaciją (65) apskaičiuoti k -tasis ir $N-k$ -tasis Furje koeficientai c_k , c_{N-k} atitinka k/T Hz dažnio galingumą signale, $k = 0, 1, \dots, N/2$. Jų modulių kvadratų suma $|c_k|^2 + |c_{N-k}|^2$ rodo k/T Hz dažnio galingumą.

Pastebėkime, kad didžiausias dažnis, kurį dar gali atspindėti diskretizuoto signalo spektras gaunamas su $k = N/2$ ir lygus

$$\frac{N/2}{T} = \frac{N/2}{N\Delta t} = \frac{1}{2\Delta t},$$

taigi pusei signalo diskretizacijos dažnio. Jeigu diskretizuojant signalą diskretizacijos dažnis buvo parinktas pagal Naikvisto–Šenono teoremą (žr. 1.2.4 skyrelį), tai, kaip matome, garantuoja, kad spektre bus informacija apie visus signale esančius dažnius, įskaitant ir patį aukščiausią.

7.1.4 Konvoliucijos (sąsūkų) ir koreliacijos ryšys su Furje transformacija

Nagrinėkime diskrečias sekas (realiųjų skaičių sekas) x_j ir y_j :

$$x_0, x_1, \dots, x_{N-1}, \quad y_0, y_1, \dots, y_{N-1}.$$

Apibrėžimas. Dviejų diskrečių sekų x_j ir y_j *sąsūka* arba *konvoliucija* (angliškai “convolution”) vadinama diskreti seka P_j , apibrėžta suma

$$P_j = \sum_{m=0}^{N-1} x_m y_{j-m}, \quad j = 0, 1, \dots, N-1. \quad (70)$$

Taip pat nagrinėsime panašią diskrečių seką (ją dabar vadinsime koreliacija)

$$Q_j = \sum_{m=0}^{N-1} x_m y_{j+m}, \quad j = 0, 1, \dots, N-1, \quad (71)$$

kuri primena tarpusavio koreliaciją, nagrinėtą 5 skyriuje.

Formulėse (70) ir (71) laikome, kad sekos y_j reikšmės periodiškai pratęsimos:

$$y_{-N} = y_0 = y_N, \quad y_{-N+1} = y_1 = y_{N+1}, \quad y_{-N+2} = y_2 = y_{N+2}, \quad \dots,$$

tai yra, $y_{j \pm N} = y_j$, $j = 0, 1, \dots, N-1$.

Dydžius, apibrėžtus panašiai kaip suma (71), sutikome nagrinėdami signalo glodinimą *slenkančio vidurkio algoritmu* (žr. 2.1.1 skyrelį), signalų *tarpusavio koreliaciją* (žr. 5.2 skyrelį) ir *autokoreliaciją* (žr. 5.3 skyrelį). Tam, kad šie mūsų anksčiau nagrinėti dydžiai būtų apibrėžti identišškai kaip (71), reikėtų periodiškai pratęsti analizuojamų signalų reikšmes – analogiškai kaip pratęsėme y_j reikšmes.

Parodysime, kad dydžiai P_j ir Q_j , apibrėžti lygybėmis (70) ir (71), gali būti skaičiuojami ne tiesiogiai, bet taikant Furje transformacijos ir atvirkštinės Furje transformacijos algoritmus.

7.2 skyrelyje išnagrinėsime itin našų (kompiuterio procesoriaus laiko požiūriu) greitosios Furje transformacijos algoritmą, kuris tinka skaičiuoti diskrečiąją Furje transformaciją (65) ir jai atvirkštinę (66). Taigi, šis algoritmas gali būti naudojamas kaip tarpinis metodas ir kituose signalų analizės ir apdorojimo algoritmuose: slenkančio vidurkio glodinyje, dviejų signalų tarpusavio koreliacijos funkcijos reikšmių skaičiavime, vieno signalo autokoreliacijos funkcijos reikšmių radime (šiek tiek pakoregavus šiuos metodus apibrėžiančias formules). Apdorojant labai daug reikšmių turinčius signalus (jeigu N labai didelis), tai gali itin ženkliai sumažinti kompiuterinių skaičiavimų trukmę.

Pažymėkime $c_k(x)$ sekos x_j Furje koeficientus, o $c_k(y)$ – sekos y_j Furje koeficientus. Taip pat pažymime $c_k(P)$ ir $c_k(Q)$ – sekų P_j ir Q_j Furje koeficientus.

Tuomet galioja lygybės:

$$c_k(P) = N c_k(x) c_k(y), \quad c_k(Q) = N \overline{c_k(x)} c_k(y), \quad k = 0, 1, \dots, N-1, \quad (72)$$

čia $\overline{c_k(x)}$ yra kompleksiskai jungtiniai dydžiai Furje koeficientams $c_k(x)$.

Taigi, apskaičiavę $c_k(x)$ ir $c_k(y)$ (pavyzdžiui, greitosios Furje transformacijos algoritmu), pritaikome atvirkštinę Furje transformaciją (vėl, pavyzdžiui, greitosios Furje transformacijos metodu) pagal formulę (72) gautiems dydžiams $c_k(P)$ arba $c_k(Q)$ ir surandame seką P_j arba Q_j .

Irodysime lygybes (72).

Pirmiausiai gauname pagalbinę tapatybę, įvedus naują sumavimo indeksą $l = j \pm m$ ir pasinaudojus tuo, kad (dėl periodinio pratęsimo) bet kuri skaičių y_l bei funkcijos $e^{-i2\pi lk/N}$ iš eilės einančių N reikšmių seka (pagal indeksą l , pavyzdžiui, kai $l = -m, -m+1, \dots, N-1-m$ arba $l = 0, 1, \dots, N-1$ arba $l = m, m+1, \dots, N-1+m$) turi tuos pačius elementus (tik išdėstytus kita tvarka)

$$\begin{aligned} \sum_{j=0}^{N-1} y_{j \pm m} e^{-i \frac{2\pi}{N} j k} &= \sum_{l=-m}^{N-1 \pm m} y_l e^{-i \frac{2\pi}{N} (l \mp m) k} = e^{\pm i \frac{2\pi}{N} m k} \sum_{l=0}^{N-1} y_l e^{-i \frac{2\pi}{N} l k} = \\ &= N e^{\pm i \frac{2\pi}{N} m k} c_k(y). \end{aligned}$$

Tada, remdamiesi (65), ir pastarąją pagalbinę lygybę, išvedame

$$\begin{aligned} c_k(P) &= \frac{1}{N} \sum_{j=0}^{N-1} \left(\sum_{m=0}^{N-1} x_m y_{j-m} \right) e^{-i \frac{2\pi}{N} jk} = \frac{1}{N} \sum_{m=0}^{N-1} x_m \left(\sum_{j=0}^{N-1} y_{j-m} e^{-i \frac{2\pi}{N} jk} \right) = \\ &= N c_k(y) \frac{1}{N} \sum_{m=0}^{N-1} x_m e^{-i \frac{2\pi}{N} mk} = N c_k(y) c_k(x). \end{aligned}$$

Analogiškai gauname

$$c_k(Q) = N c_k(y) \frac{1}{N} \sum_{m=0}^{N-1} x_m e^{i \frac{2\pi}{N} mk} = N c_k(y) \overline{c_k(x)}.$$

7.2 Greitoji Furje transformacija

Diskrečioji Furje transformacija (65) – tai algoritmas, kurį būtina žinoti, norint kompiuterio pagalba analizuoti skaitmeninių signalų spektrus. Nežinodami šio algoritmo teorinės prasmės ir savybių, o tik akiai skaičiuodami, nesuvoksime ir skaičiavimo rezultatuose – signalų spektruose esančios informacijos. Pati formulė (65) atrodo nesudėtingai tiek teoretiko, tiek programuotojo požiūriu. Tačiau tarp signalų analizės ir apdorojimo specialistų praktiškai nesutiksime tokių, kurie skaičiuotų spektrus diskrečiosios Furje transformacijos metodu (nepaisant to, kad jį tikrai lengva suprogramuoti kompiuteriu). To priežastis – šis algoritmas labai neefektyvus.

Jeigu reiktų apdoroti nedaug reikšmių ($N < 1000$) turintį skaitmeninį signalą, skaičiavimai ilgai neužtruktų. Bet vos tik duomenų kiekis pasiektų dešimtis tūkstančių, tektų apsišarvuoti kantrybe. Jeigu $N = 10^6$ (milijonu reikšmių diskretizuotas signalas), spektro reikšmių lauktime apie savaitę. Tačiau dirbant su realiais signalais dažnai būtina apdoroti labai didelius duomenų srautus.

1965 metais JAV mokslininkai James W. Cooley ir John W. Tukey paskelbė straipsnį, kuriame aprašė labai efektyvų algoritmą, realizuojantį diskrečiąją Furje transformaciją. Šis algoritmas dabar vadinamas *greitąja Furje transformacija* (angliškai “fast Fourier transform” arba tiesiog “FFT”). Tiesa, vėliau paaiškėjo, kad dar apie 1805 m. labai panašias idėjas savo skaičiavimuose naudojo žinomas vokiečių matematikas Johann Carl Friedrich Gauss, taigi 1965 m. šis algoritmas buvo iš naujo atrastas ir išpopuliarintas.

Greitosios Furje transformacijos metodas (kurį nagrinėsime šiame skyrelyje) ir įvairios jo atmainos aprašytos daugybėje knygų (žr., pvz., [1, 2, 3, 13]). Taip pat šis algoritmas optimizuotas įvairių architektūrų kompiuterių procesoriams, sukurtos efektyvios paprogramių bibliotekos, tiek atvirojo kodo (pvz., [14, 15]), tiek komercinės (pvz., [16, 17]).

Greitoji Furje transformacija sėkmingai išnaudoja išraiškoje (65) dalyvaujančių harmonikų periodiškumo savybes, taip išvengiant nereikalingų skaičiavimų, kurie būtų atliekami diskrečiojoje Furje transformacijoje. Tikslioje aritmetikoje abu algoritmai duoda identiškus rezultatus, tačiau realiuose skaičiavimuose (kuriuose neišvengiamos apvalinimo paklaidos), greitoji Furje transformacija netgi tiksliau suranda signalo spektrą, palyginus su diskrečiąja Furje transformacija. Didesnio tikslumo priežastis paprasta: našesnis algoritmas tiesiog vykdo mažiau aritmetinių skaičiavimų, todėl ir apvalinimo paklaidos pasireiškia mažiau kartų.

7.2.1 Greitosios ir diskrečiosios Furje transformacijų palyginimas

Egzistuoja daug greitosios Furje transformacijos atmainų, tačiau mes nagrinėsime tik vieną – vadinamąjį Cooley–Tukey algoritmą, kuris dažniausiai naudojamas ir yra paprasčiausias. Šis algorit-

mas turi esminę prielaidą: apdorojamo skaitmeninio signalo reikšmių skaičius N turi būti lygus tam tikram dvejetainio laipsniui:

$$N = 2^n, \quad \text{kur } n > 0 \text{ yra sveikasis skaičius.} \quad (73)$$

Pavyzdžiui, galimos reikšmės $N = 2^7 = 128$, $N = 2^{10} = 1024$, $N = 2^{20} = 1048576$, žinoma ir visi kiti dvejetainio laipsniai. Tai, kad algoritmas gali veikti tik su prielaida (73), nėra jo esminis trūkumas – paprastai eksperimente registruojant signalo reikšmes nesunku nustatyti norimą jų kiekį.

Reikėtų pasakyti, kad yra sukurti greitosios Furje transformacijos algoritmai, nereikalaujantys sąlygos (73) bei pasižymintys panašiu veikimo greičiu, tačiau ir ženkliai sudėtingesni. Egzistuoja taipogi žymiai sudėtingesni metodai, veikiantys irgi su prielaida (73), bet keletą kartų greičiau nei Cooley–Tukey algoritmas. Vis dėlto, visos žinomos greitosios Furje transformacijos atmainos turi tą patį *algoritminį sudėtingumą* $O(N \log_2 N)$, t. y. reikalingų atlikti aritmetinių operacijų skaičius proporcingas $N \log_2 N$, skiriasi tik proporcingumo konstantos.

Nagrinėdami diskrečiosios Furje transformacijos formulę

$$c_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i \frac{2\pi}{N} jk}, \quad k = 0, 1, \dots, N-1.$$

matome, kad visų koeficientų c_k suradimui reikia apskaičiuoti N^2 kompleksinių sandaugų (su sąlyga, kad skaičius $e^{-i2\pi jk/N}$ turime paskaičiuavę iš anksto ir saugome masyvuose) bei $N(N-1)$ kompleksinių sudėčių. Taigi, diskrečiosios Furje transformacijos algoritmo sudėtingumas $O(N^2)$.

Kaip matysime toliau (žr. 7.2.3 skyrelį), tuos pačius koeficientus c_k randant Cooley–Tukey greitosios Furje transformacijos algoritmu, reikia skaičiuoti apytiksliai

$$\frac{N}{2} \log_2 N$$

kompleksinių sandaugų (apytiksliai, nes kai kuriais atvejais vienas dauginamųjų būna lygus 1 arba -1 , taigi iš principo dar šiek tiek ši skaičių galima sumažinti). Taip pat reikia skaičiuoti $N \log_2 N$ kompleksinių sudėčių, tačiau sudėties veiksmas kompiuteryje vykdomas keletą kartų greičiau nei daugybos.

Taigi, greitoji Furje transformacija kur kas efektyvesnė skaičiavimo laiko požiūriu, nei diskrečioji. Skirtumas ypač esminis dideliems N , žr. 64 pav. Pavyzdžiui, kai $N = 1024$, greitojoje Furje transformacijoje bus skaičiuojama apie 200 kartų mažiau sandaugų, tiek pat kartų sutrumpės viso algoritmo vykdymo laikas.

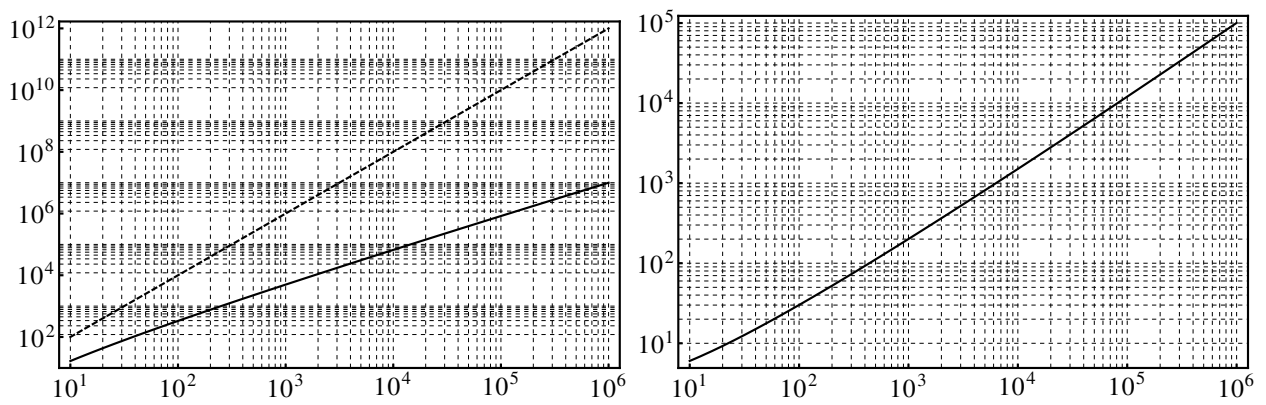
Dar labiau įtikinantis pavyzdys toks: tarkime, reikia rasti signalo, turinčio $N = 2^{25} = 33554432$ reikšmių spektrą. Greitoji Furje transformacija skaičiuojant konkrečiu kompiuteriu užtrunka, sakykime, apie 5 sekundes. Pavyzdžiui, galite pabandyti Mathematica skaičiavimų terpėje įvesti komandą:

```
Timing[ Fourier[ RandomReal[ 1, 2^25 ] ]; ]
```

Identiškų rezultatų skaičiuodami tuo pačiu kompiuteriu diskrečiosios Furje transformacijos algoritmu lauktumėte daugiau nei 5 mėnesius!

Toliau detalai nagrinėsime Cooley–Tukey greitosios Furje transformacijos metodą, kuris tinka ne tik spektro c_k radimui, bet ir atvirkštinės Furje transformacijos realizacijai (žr. pastabą 7.1.1 skyrelio pabaigoje).

Beje, verta paminėti, kad greitajai Furje transformacijai galioja tos pačios savybės (žr. 7.1.2) skyrelį), kaip ir diskrečiąjai Furje transformacijai, kadangi abiem algoritmais gaunami tie patys signalo spektro koeficientai c_k .



64 pav.: Kompleksinių sandaugų skaičiaus, reikalingo diskrečiosios Furje transformacijos (DFT) ir Cooley–Tukey greitosios Furje transformacijos (FFT) algoritmams, palyginimas. Iliustracija kairėje: DFT (punktyrinė kreivė) ir FFT (ištininė kreivė) kompleksinių sandaugų skaičiaus priklausomybė nuo apdorojamo skaitmeninio signalo reikšmių skaičiaus N . Iliustracija dešinėje: kiek kartų greitesnis FFT algoritmas, palyginus su DFT metodu (santykis tarp punktyrinės kreivės ir ištininės kreivės grafikų iliustracijoje kairėje), priklausomai nuo N . Abu grafikai pateikti log-log mastelyje.

7.2.2 Cooley–Tukey greitosios Furje transformacijos algoritmas

Išankstinis duomenų paruošimas

Prieš pradedant apdoroti signalo reikšmes Cooley–Tukey greitosios Furje transformacijos algoritmu, būtina patikrinti ar signalo reikšmių kiekis N tenkina sąlygą (73).

Jeigu ši sąlyga netenkinama ir mes negalime kontroliuoti signalo reikšmių skaičiaus, įtakodami duomenų registravimo procesą, belieka dvi išeitys. Galima atmesti dalį turimų duomenų, taip, kad likęs reikšmių kiekis būtų lygus artimiausiam dvejetainio laipsniui (pavyzdžiui, jeigu $N = 1000$, atmetame 488 reikšmes ir apdorojame likusius $512 = 2^9$ skaičius. Tačiau duomenų kiekio sumažinimas taip pat sumažins gaunamo spektro dažninę skiriamąją gebą (rezoliuciją), taigi ši išeitis nėra patartina. Geresnis sprendimas – papildyti turimą duomenų seką nuliais, pridedant juos prie apdorojamų reikšmių sekos pabaigos. Taigi, turėdami $N = 1000$ reikšmių, jų sekos pabaigoje prirašome 24 nulius ir vykdome greitąją Furje transformaciją su $N = 1024 = 2^{10}$ ilgio skaitmeniniu signalu.

Cooley–Tukey algoritmas. Pagrindinių formulų išvedimas

Dėl trumpumo formulėje (65) atsisakysime daugiklio $1/N$, tai yra, išvesime algoritmą dydžių $C_k = Nc_k$ skaičiavimui:

$$C_k = Nc_k = \sum_{j=0}^{N-1} f_j e^{-i \frac{2\pi}{N} jk}, \quad k = 0, 1, \dots, N-1. \quad (74)$$

Pagrindinė Cooley–Tukey algoritmo idėja – suskaidyti sumą (74) į dvi sumas. Pirmojoje sumoje dalyvaus f_j reikšmės su lyginiais indeksais:

$$f_0, f_2, f_4, \dots, f_{N-2},$$

antrojoje – su nelyginiais indeksais:

$$f_1, f_3, f_5, \dots, f_{N-1}.$$

Kadangi, remiantis prielaida (73), $N = 2^n$, kiekvienoje iš šių sumų bus po 2^{n-1} narių. Vadinasi, pradinę sumą (74) su $N = 2^n$, suvedame į dvi tokio paties tipo sumas su $N = 2^{n-1}$. Toliau rekursiškai kartojame tą patį procesą.

Tačiau, kad algoritmas būtų optimizuotas, reikia pasinaudoti dydžių $e^{-i2\pi jk/N}$ savybėmis.

Detaliai išrašykime gaunamas išraiškas. Suskaidę sumą pagal lyginius ir nelyginius f_j indeksus, gauname:

$$C_k = \sum_{j=0}^{N/2-1} f_{2j} e^{-i\frac{2\pi}{N}(2j)k} + \sum_{j=0}^{N/2-1} f_{2j+1} e^{-i\frac{2\pi}{N}(2j+1)k}.$$

Iš antrosios sumos galime iškelti nepriklausantį nuo sumavimo indekso j daugiklį $e^{-i2\pi k/N}$:

$$C_k = \sum_{j=0}^{N/2-1} f_{2j} e^{-i\frac{2\pi}{N/2}jk} + e^{-i\frac{2\pi}{N}k} \sum_{j=0}^{N/2-1} f_{2j+1} e^{-i\frac{2\pi}{N/2}jk}.$$

Pažymėkime kompleksinį skaičių $W_N = e^{-i2\pi/N}$. Tuomet

$$C_k = \sum_{j=0}^{N/2-1} f_{2j} W_{N/2}^{jk} + W_N^k \sum_{j=0}^{N/2-1} f_{2j+1} W_{N/2}^{jk}, \quad k = 0, 1, \dots, N-1. \quad (75)$$

Pasirodo, formulė (75) pasižymi puikia savybe: pagal ją pakanka apskaičiuoti koeficientus $C_0, C_1, \dots, C_{N/2-1}$, tada likusioji pusė koeficientų $C_{N/2}, C_{N/2+1}, \dots, C_{N-1}$ gaunami labai pigiai (aritmetinių operacijų skaičiaus požiūriu).

Išvesime lygybę, iš kurios galima apskaičiuoti koeficientą $C_{k+N/2}$, su sąlyga, kad jau turime C_k , $k = 0, 1, \dots, N/2 - 1$. Į išraišką (75) vietoje indekso k įrašykime $k + N/2$:

$$C_{k+N/2} = \sum_{j=0}^{N/2-1} f_{2j} W_{N/2}^{j(k+N/2)} + W_N^{k+N/2} \sum_{j=0}^{N/2-1} f_{2j+1} W_{N/2}^{j(k+N/2)}.$$

Kadangi, su bet koku sveikuoju j dydis $e^{-i2\pi j} = 1$, turime

$$W_{N/2}^{j(k+N/2)} = W_{N/2}^{jk} W_{N/2}^{jN/2} = W_{N/2}^{jk} e^{-i\frac{2\pi}{N/2}j(N/2)} = W_{N/2}^{jk} e^{-i2\pi j} = W_{N/2}^{jk}.$$

Taip pat, remiantis lygybe $e^{-i\pi} = -1$, gauname

$$W_N^{k+N/2} = W_N^k W_N^{N/2} = W_N^k e^{-i\frac{2\pi}{N} \frac{N}{2}} = W_N^k e^{-i\pi} = -W_N^k.$$

Vadinasi,

$$C_{k+N/2} = \sum_{j=0}^{N/2-1} f_{2j} W_{N/2}^{jk} - W_N^k \sum_{j=0}^{N/2-1} f_{2j+1} W_{N/2}^{jk}, \quad k = 0, 1, \dots, \frac{N}{2} - 1. \quad (76)$$

Kaip matome, gautoje formulėje (76) sumų skaičiuoti iš naujo nebereikia – jos identiškos kaip išraiškoje (75), taigi jau suskaičiuotos.

Cooley–Tukey algoritmas. Rekursija

Pažymėkime

$$A_k = \sum_{j=0}^{N/2-1} f_{2j} W_{N/2}^{jk}, \quad B_k = \sum_{j=0}^{N/2-1} f_{2j+1} W_{N/2}^{jk}, \quad k = 0, 1, \dots, \frac{N}{2} - 1. \quad (77)$$

Taip pat prisiminkime, kad naudojamas pažymėjimas

$$W_N = e^{-i \frac{2\pi}{N}}. \quad (78)$$

Remdamiesi lygybėmis (75) ir (76), gauname koeficientų C_k skaičiavimo formules:

$$C_k = \begin{cases} A_k + W_N^k B_k, & k = 0, 1, \dots, \frac{N}{2} - 1, \\ A_{k-N/2} - W_N^{k-N/2} B_{k-N/2}, & k = \frac{N}{2}, \frac{N}{2} + 1, \dots, N-1. \end{cases} \quad (79)$$

Siekiant maksimalios optimizacijos, įvairiems N ir k konstantas W_N^k galima iš anksto apskaičiuoti ir laikyti masyvuose, nes jos nepriklauso nuo apdorojamo signalo reikšmių f_j .

Pastebėkime, kad dydžiai A_k ir B_k , apibrėžti lygybėmis (77), yra absoliučiai analogiškos prigimties kaip pagrindiniai mūsų ieškomieji Furje koeficientai C_k , kurie formulėje (74) panaudojus pažymėjimą (78), turi pavidalą

$$C_k = \sum_{j=0}^{N-1} f_j W_N^{jk}.$$

Iš tiesų, pažymėję $\tilde{N} = N/2$, $\tilde{f}_j = f_{2j}$, $j = 0, 1, \dots, \tilde{N}-1$, galime užrašyti

$$A_k = \sum_{j=0}^{\tilde{N}-1} \tilde{f}_j W_{\tilde{N}}^{jk}.$$

Matome, kad ši suma visiškai analogiška C_k apibrėžiančiai sumai, tik A_k atveju kitaip apibrėžime signalo reikšmės bei jų skaičius sumažėjo dvigubai. Tokią pačią analogiją galime parodyti ir koeficientų B_k atveju.

Išvada. Pradinį uždavinį (C_k apskaičiavimą) su parametru $N = 2^n$ suvedėme į du tokio paties tipo uždavinius (A_k ir B_k apskaičiavimą) su dvigubai mažesniu parametru $N = 2^{n-1}$ (kuris taip pat yra dvejetainio laipsnis). Formulės (77) ir (79) nusako vieną Cooley–Tukey algoritmo žingsnį.

Akivaizdu, kad net jeigu dydžius A_k ir B_k skaičiuotume diskrečiosios Furje transformacijos metodu, jau gautume dvigubą algoritmo pagreitėjimą, nes $(N/2)^2 + (N/2)^2 = N^2/2 < N^2$.

Tačiau galime tiek A_k , tiek B_k sumoms vėl pakartoti Cooley–Tukey algoritmo žingsnį. Gausime 4 sumas, kiekvieną iš 2^{n-2} dėmenų. Tęsdami toliau, galiausiai išskaidysime pradinę C_k sumą į N „sumų“, kurių kiekvienoje tebus po vieną dėmenį, taigi teliks sudėti N skaičių.

Taigi, Cooley–Tukey greitosios Furje transformacijos algoritmas yra *rekursinis* metodas.

7.2.3 Cooley–Tukey algoritmo optimizavimas

Siekiant dar labiau optimizuoti Cooley–Tukey greitosios Furje transformacijos algoritmą, programuojant galima atsisakyti rekursijos, o duomenis (signalų reikšmės f_j) iš karto išdėstyti tokia tvarka, kokia jie būtų sumuojami paskutiniame (giliausiam) rekursijos lygyje.

Taip pat verta iš anksto suskaičiuoti ir laikyti masyvuose nepriklausančias nuo apdorojamo signalo reikšmių f_j kompleksines konstantas W_m^k , tik tokiems sveikiesiems m ir k , kuriems šių konstantų iš tiesų prireiks skaičiuojant.

Pasiaiškinsime kokia tvarka Cooley–Tukey algoritme apdorojamos signalo reikšmės f_j bei kurie koeficientai W_m^k reikalingi.

Algoritmo veikimo principą galima perprasti nagrinėjant nedaug reikšmių turinčių duomenų atvejį. Tarkime, $N = 8 = 2^3$. Tuomet duomenys (skaitmeninio signalo reikšmės) yra skaičiai

$$f = (f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7).$$

Mūsų tikslas – apskaičiuoti dydžius

$$C_k = f_0 + f_1 W_8^k + f_2 W_8^{2k} + f_3 W_8^{3k} + f_4 W_8^{4k} + f_5 W_8^{5k} + f_6 W_8^{6k} + f_7 W_8^{7k},$$

$$k = 0, 1, 2, 3, 4, 5, 6, 7.$$

Pirmajame Cooley–Tukey algoritmo rekursijos lygyje, remiantis formulėmis (77) ir (79), kiekvienam fiksuotam $k = 0, 1, 2, 3$ gausime dvi sumas A_k ir B_k iš keturių dėmenų:

$$C_k = A_k + W_8^k B_k, \quad C_{k+4} = A_k - W_8^k B_k, \quad k = 0, 1, 2, 3,$$

$$A_k = \left[f_0 + f_2 W_4^k + f_4 W_4^{2k} + f_6 W_4^{3k} \right], \quad B_k = \left[f_1 + f_3 W_4^k + f_5 W_4^{2k} + f_7 W_4^{3k} \right],$$

arba

$$C_k = \left[f_0 + f_2 W_4^k + f_4 W_4^{2k} + f_6 W_4^{3k} \right] + W_8^k \left[f_1 + f_3 W_4^k + f_5 W_4^{2k} + f_7 W_4^{3k} \right].$$

Antrajame rekursijos lygyje kiekvienam fiksuotam $k = 0, 1$ turėsime jau keturias sumas AA_k , AB_k , BA_k , BB_k , kiekvieną iš dviejų dėmenų:

$$A_k = AA_k + W_4^k AB_k, \quad A_{k+2} = AA_k - W_4^k AB_k,$$

$$B_k = BA_k + W_4^k BB_k, \quad B_{k+2} = BA_k - W_4^k BB_k, \quad k = 0, 1,$$

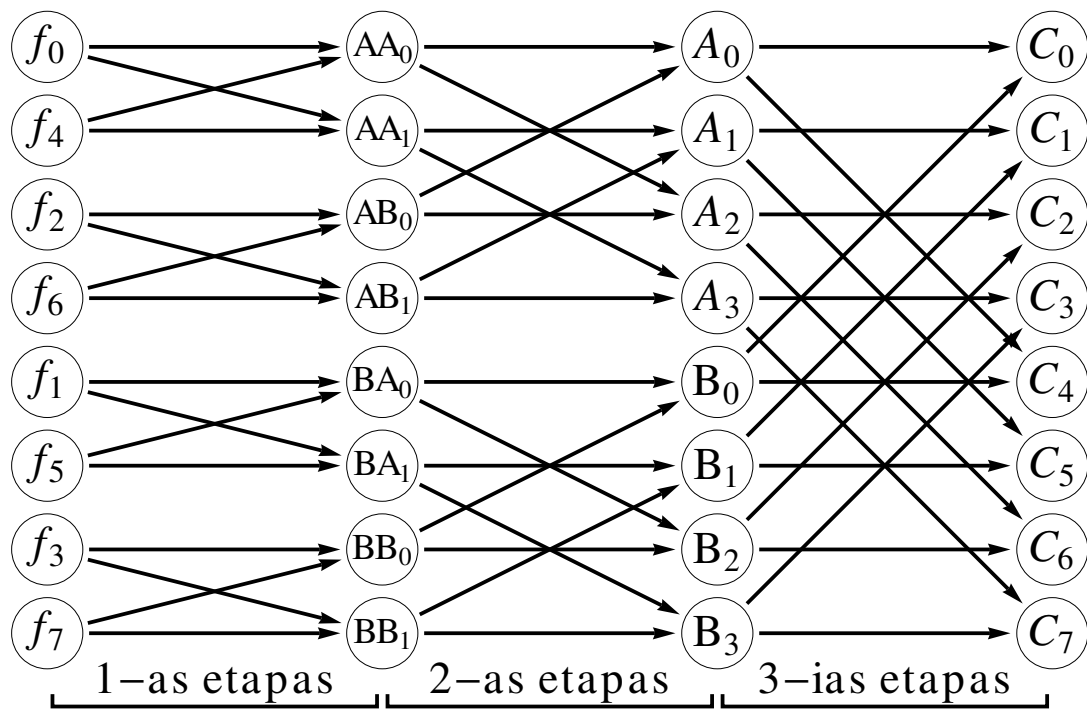
$$AA_k = \left(f_0 + f_4 W_2^k \right), \quad AB_k = \left(f_2 + f_6 W_2^k \right),$$

$$BA_k = \left(f_1 + f_5 W_2^k \right), \quad BB_k = \left(f_3 + f_7 W_2^k \right),$$

arba

$$C_k = \left[\left(f_0 + f_4 W_2^k \right) + W_4^k \left(f_2 + f_6 W_2^k \right) \right] + W_8^k \left[\left(f_1 + f_5 W_2^k \right) + W_4^k \left(f_3 + f_7 W_2^k \right) \right].$$

Trečiajame rekursijos lygyje formaliai būtų apibrėžtos aštuonios “sumos”, kiekviena sudaryta iš vienintelio dėmens – signalo reikšmės. Kokia tvarka išsidėsčiusios šios reikšmės ir iš kokių koeficientų juos reikės padauginti sumuojant, matyti jau iš antrojo rekursijos lygio formulių. Todėl iš karto galime pereiti prie 1-ojo algoritmo etapo, kuriame prasideda realūs skaičiavimai.



65 pav.: Cooley–Tukey greitosios Furje transformacijos algoritme vykdomų aritmetinių operacijų schema 8 taškų ($N = 8$) atveju. Rodyklės rodo kurie dydžiai iš praeito algoritmo etapo dalyvauja skaičiuojant sekančio etapo dydžius.

1-as etapas. Naudodami signalo reikšmes f_j ir koeficientus W_2^k , $k = 0, 1$, apskaičiuojame tarpinius dydžius AA_k, AB_k, BA_k, BB_k , $k = 0, 1$. Skaičiavimo schema pateikta 65 pav. Šioje schemoje rodyklės rodo kurie dydžiai dalyvauja atliekant aritmetines operacijas. Pavyzdžiui,

$$AA_0 = f_0 + f_4 W_2^0,$$

taigi iš f_0 ir iš f_4 eina po rodyklę į AA_0 . Rodyklė taip pat reiškia vieną aritmetinę operaciją: f_4 dauginamas iš konstantos W_2^0 ; rezultatas sudedamas su f_0 .

Beje, 1-ajame etape prireikia tik vienos konstantos $W_2^0 = 1$ (kadangi $W_2^1 = -W_2^0 = -1$). Kadangi visi daugikliai lygūs vienetui, šiame etape sandaugų skaičiavimo galima išvengti.

Indeksai j tradicine tvarka	Dvejetainiai indekso j bitai	Indekso j bitai atvirkščiai	Indeksai j atvirkščių bitų tvarka
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

1 lentelė: Cooley–Tukey algoritmo 8 taškų ($N = 8$) atveju optimalaus duomenų f_j išdėstymo tvarka pagal indeksus j su atvirkščiai užrašytais dvejetainiais bitais.

Taip pat, matome kokia tvarka reikėtų išdėstyti duomenis, kad jie būtų greičiausiai pasiekiami:

$$f_0, f_4, f_2, f_6, f_1, f_5, f_3, f_7.$$

Žiūrint į šią išdėstymo tvarką, sunkoka suvokti kaip ją būtų galima apibendrinti kai $N = 16$, $N = 32$ ir taip toliau. Viskas paaiškėja duomenų indeksus išreiškus ne dešimtainėje, bet dvejetainėje skaičiavimo sistemoje, žr. 1 lentelę. Tereikia reikšmių f_j indeksų dvejetainius bitus užrašyti atvirkštine (veidrodine) tvarka.

2-as etapas. Dabar jau naudojame praeitame – 1-ajame etape apskaičiuotus dydžius $AA_k, AB_k, BA_k, BB_k, k = 0, 1$ ir suskaičiuojame šio etapo tarpinius dydžius $A_k, B_k, k = 0, 1, 2, 3$. Skaičiavimo schema matoma 65 pav.

2-ajame etape reikalingos dvi konstantos $W_4^k, k = 0, 1$.

3-ias etapas. Turėdami praeitame (2-ajame) etape apskaičiuotus dydžius $A_k, B_k, k = 0, 1, 2, 3$, randame koeficientus $C_k, k = 0, 1, 2, 3, 4, 5, 6, 7$. Skaičiavimo schema pateikta 65 pav.

3-iajame etape naudojame konstantas $W_8^k, k = 0, 1, 2, 3$.

3-iasis etapas yra paskutinis, kadangi turėjome apdoroti $N = 2^3$ ilgio duomenų seką.

Visose algoritmo sandaugose vienas iš dauginamųjų yra kompleksinė konstanta W_m^k . Naudojamos tik konstantos (šiems indeksams m ir k):

$$W_m^k = e^{-i \frac{2\pi}{m} k} = \cos \frac{2\pi k}{m} - i \sin \frac{2\pi k}{m}, \quad m = 2, 4, 8, 16, \dots, N, \quad k = 0, 1, \dots, \frac{m}{2} - 1.$$

Beje, kai $k = 0, W_m^0 = 1$.

Cooley–Tukey algoritmo sudėtingumas

Įvertinkime kiek aritmetinių operacijų iš viso daroma Cooley–Tukey algoritme. Atimtį ir sudėtį laikysime ekvivalenčiomis kompiuterio procesoriaus laiko požiūriu operacijomis, o sandaugas suskaičiuosime atskirai.

Kaip matome iš nagrinėto pavyzdžio, reikalingi $n = \log_2 N$ etapai. Kiekviename etape skaičiuojama N sudėčių ir $N/2$ sandaugų. Sandaugų yra dvigubai mažiau nei sudėčių, nes, remiantis formulėmis (79), kiekviename etape antroji pusė dydžių skaičiuojama analogiškai kaip pirmoji pusė, tik pakeitus sudėtį į atimtį. Pavyzdžiui, mūsų nagrinėtame pavyzdyje su $N = 8$, 2-ajame etape skaičiavome dydžius A_k pagal formules

$$A_k = AA_k + W_4^k AB_k, \quad A_{k+2} = AA_k - W_4^k AB_k, \quad k = 0, 1,$$

taigi skaičiuojant A_{k+2} nebereikia skaičiuoti sandaugos $W_4^k AB_k$ (ji jau suskaičiuota, randant A_k).

Vadinasi, iš viso reikia atlikti $(N/2) \log_2 N$ kompleksinių sandaugų ir $N \log_2 N$ kompleksinių sudėčių. Sandaugų kiekį galima dar šiek tiek sumažinti, pasinaudojus tuo, kad $W_m^0 = 1$ bei kitais specialiais konstantų W_m^k atvejais. Pavyzdžiui, kaip matėme 1-ajame algoritmo etape galima iš viso išvengti dauginimo veiksmo.

Taigi, Cooley–Tukey greitosios Furje transformacijos algoritmo sudėtingumas yra $O(N \log_2 N)$.

7.3 Dažinių komponentų filtravimas Furje filtru

Vieni pagrindinių greitosios Furje transformacijos algoritmo, išnagrinėto 7.2 skyrelyje, taikymų praktikoje – triukšmo paveikto signalo filtravimas arba signalo tyrėją dominančių dažnių išryškinimas

(apie kitus taikymus trumpai kalbėjome 7.1.4 skyrelyje, o sąrašą būtų galima pratęsti, nes greitoji Furje transformacija naudojama daugelyje sričių).

2 skyriuje kalbėjome apie signalo glodinimo (tuo pačiu ir triukšmų sumažinimo) algoritmus slenkančio vidurkio metodu. Šie algoritmai veikia lokaliu principu: paimamos greta esančios signalo reikšmės ir suvidurkinamos, tikintis, kad triukšminės dedamosios eliminuos viena kitą. Tai gali duoti neblogų rezultatų, bet tik jeigu triukšmas sąlyginai nedidelės amplitudės (nedominuoja signalo reikšmių, visiškai jų neužgožia), be to, triukšmo reikšmės nepriklauso viena nuo kitos ir turi nulinį vidurkį (triukšminė dedamoji su vienoda tikimybe gali įgyti tiek teigiamas, tiek neigiamas vertes).

Šiame skyrelyje aptarsime *Furje filtro* veikimo principą ir pateiksime pavyzdžius.

Furje filtras apdoroja ne tiesiogiai pačias signalo reikšmes, tačiau signalo spektrą. Šis metodas gali pašalinti iš signalo nepageidaujamo dažnio komponentes (harmonikas) ir palikti tik tas harmonikas, kurios algoritmo vartotojo požiūriu yra informatyvios. Pašalinus tam tikrus dažnius (arba visą dažnių intervalą) iš signalo, visos signalo reikšmės paveikiamos globaliai, nepriklausomai nuo jų eiliškumo signale ar kaimynystės su kitomis reikšmėmis.

Taigi, Furje filtras gali eliminuoti ne tik triukšmus, bet ir išskirti tuos dažnius ar dažnių grupes (tokiu būdu išryškinant jų įtaką), kurios gali būti įdomios analizuojant signalą.

Skaitmeninio signalo filtravimo Furje metodu veikimo principas nesudėtingas: pirmiausiai naudodami diskrečiąją Furje transformaciją apskaičiuojame spektro koeficientus (Furje koeficientus) c_k . Žinoma, taupant skaičiavimų laiką, jų radimui verta taikyti greitosios Furje transformacijos algoritmą. Tuomet pakeičiame specifinius dažnius (kuriuos pasirenka vartotojas) arba ištisus dažnių intervalus atitinkančius spektrinius koeficientus c_k . Koeficiento c_k pakeitimas gali, pavyzdžiui, reikšti: c_k reikšmės užnulinimą; jo modulio reikšmės $|c_k|$ sumažinimą arba padidinimą iki nustatyto dydžio, paliekant nepakeistą fazę $\arg c_k$. Galiausiai, įvykdome atvirkštinę diskrečiąją Furje transformaciją (kaip taisyklė, vėl greitosios Furje transformacijos algoritmu). Gautų skaičių *realiosios dalys* (jeigu tarp apdorojamo pradinio signalo reikšmių nebuvo kompleksinių dydžių) ir bus filtruoto signalo reikšmės.

Svarbu prisiminti, kad reikšminiai yra tik Furje koeficientai c_k , kai $k = 0, 1, \dots, N/2$ (žr. 7.1.3 skyrelį), čia N – signalo reikšmių skaičius. Likusieji koeficientai (kai apdorojamas signalas neturi kompleksinių reikšmių) susiję su pirmaisiais simetrijos savybe (67):

$$c_{N-k} = \bar{c}_k, \quad k = 1, 2, \dots, N-1.$$

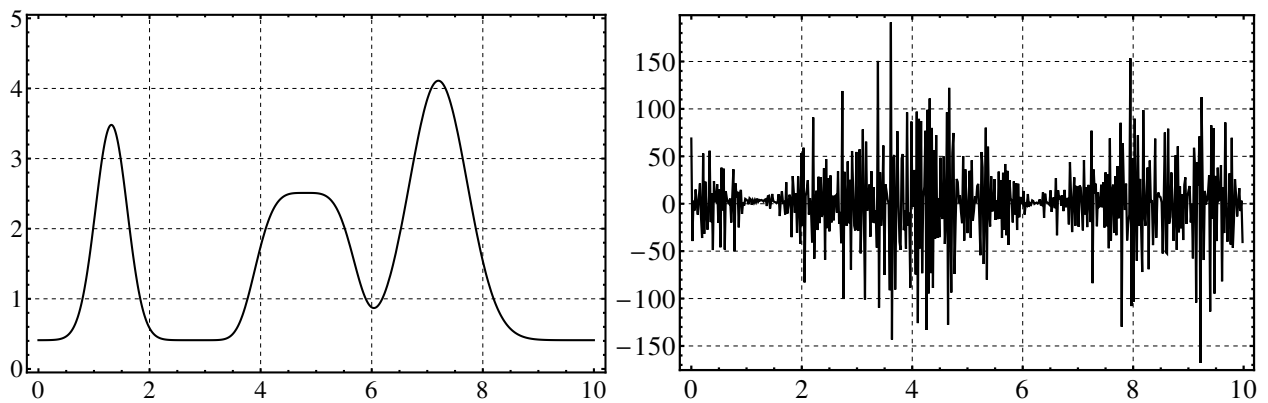
Taigi, dažninių komponentių filtravimo metu keičiant tam tikro koeficiento c_k reikšmę, būtina atitinkamai pakeisti ir c_{N-k} reikšmę, kad liktų galioti ši simetrijos savybė.

Furje filtro veikimo pavyzdys pateiktas 66 pav. ir 67 pav. Šiame pavyzdyje naudojamas *žemus dažnius praleidžiantis filtras* (angliškai “low-pass filter”), kuris veikia palikdamas signale tik žemą dažnį turinčias harmonikas. Tai yra, filtravimo metu spektre užnulinamos Furje koeficientų c_k , kurių indeksai viršija paliekamų harmonikų skaičių k_p , reikšmės:

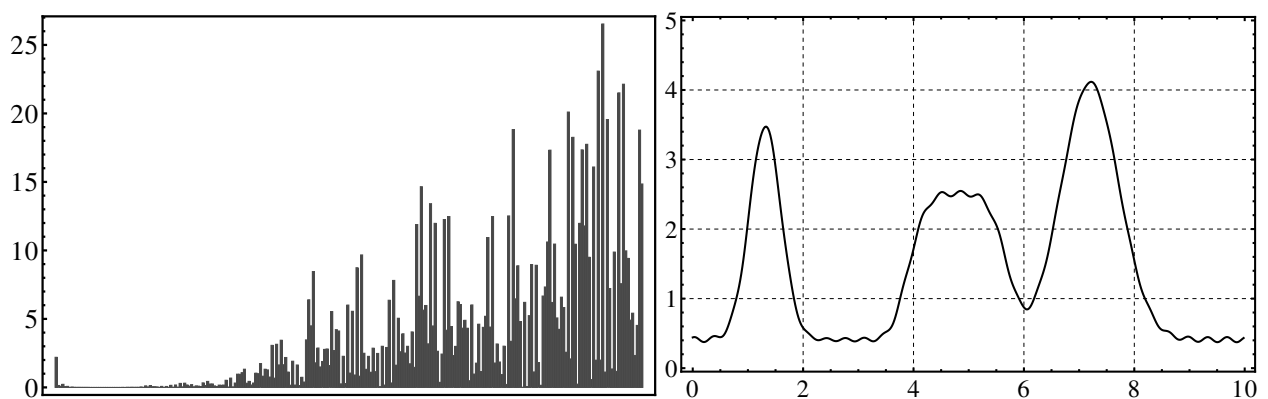
$$c_k = 0, \quad k = k_p + 1, k_p + 2, \dots, \frac{N}{2}, \dots, N - 1 - k_p, \quad (80)$$

o visi kiti koeficientai c_k paliekami nepakeisti.

Taikydamas tokį filtravimo algoritmą, vartotojas mėgina surasti optimalią parametro k_p vertę, kuri leistų ne tik pašalinti daugiausiai triukšmo, bet ir minimaliai iškreiptų signalą. Žinoma, taikant šį metodą, reikia žinoti kokie maždaug dažniai gali būti būdingi informatyviajai signalo daliai, o kokie triukšmams, taip pat, kaip maždaug turėtų atrodyti pats signalas. Kadangi, skaičiuojant gali tekti bandyti daug įvairių parametro k_p reikšmių, ypač svarbu taikyti maksimaliai greitą algoritmą Furje transformacijai ir jai atvirkštinei vykdyti.



66 pav.: Originalus signalas (ilustracija kairėje) ir paveiktas triukšmo signalas (ilustracija dešinėje). Matyti, kad triukšmas yra visiškai užgožęs signalo reikšmes – triukšmo amplitudė keliolika kartų viršija originalaus signalo amplitudę.



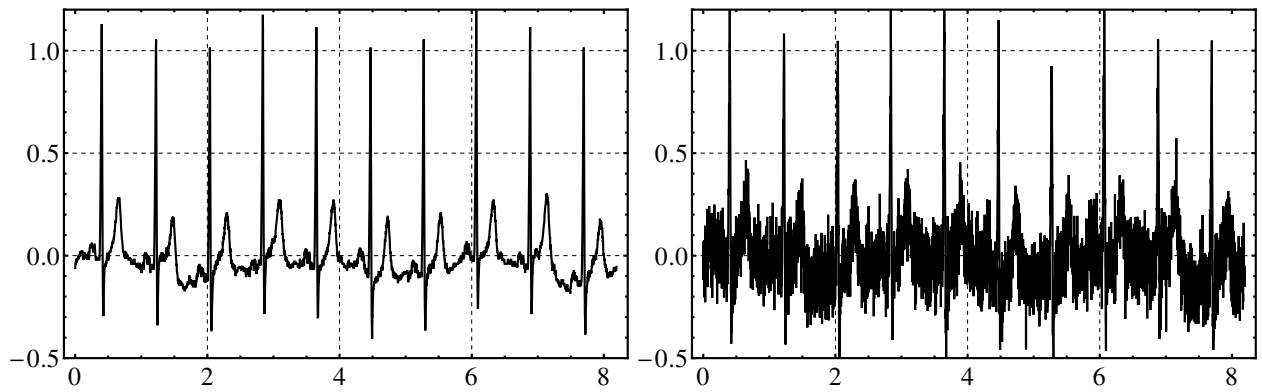
67 pav.: Ilustracija kairėje: apdorojamo užtriukšminto signalo (žr. 66 pav., ilustracija dešinėje) galios spektras: dydžiai $|c_k|^2$, $k = 0, 1, \dots, N/2$, $N = 512$. Iš spektro matosi, kad signale vyrauja aukštus dažnius turinčios komponentės (labai greitai osciliuojančios ir turinčios didelę amplitudę). Tačiau apatiniame kairiajame iliustracijos kampe taip pat matome ir tam tikrą žemų dažnių harmonikų buvimą. Todėl natūralu pabandyti palikti šias žemą dažnį turinčias komponentes, užnulinant visus spektro koeficientus c_k , kurių indeksas k didesnis už tam tikrą skaičių k_p (paliekamų harmonikų skaičių). Ilustracija dešinėje: Furje filtro (80) rezultatas – atstatytas signalas, gautas palikus pirmąsias $k_p = 30$ harmonikų.

Kaip matyti iš kito pavyzdžio, žr. 68 pav., 69 pav. ir 70 pav., kuriame taip pat naudojamas žemus dažnius praleidžiantis Furje filtras (80), nustačius per mažą filtravimo algoritmo parametro k_p reikšmę, pašalinamos ir esminės signalo dalys. Tačiau jeigu parenkame per didelį k_p , tai reiškia, kad po filtravimo išlieka ir dalis triukšmo.

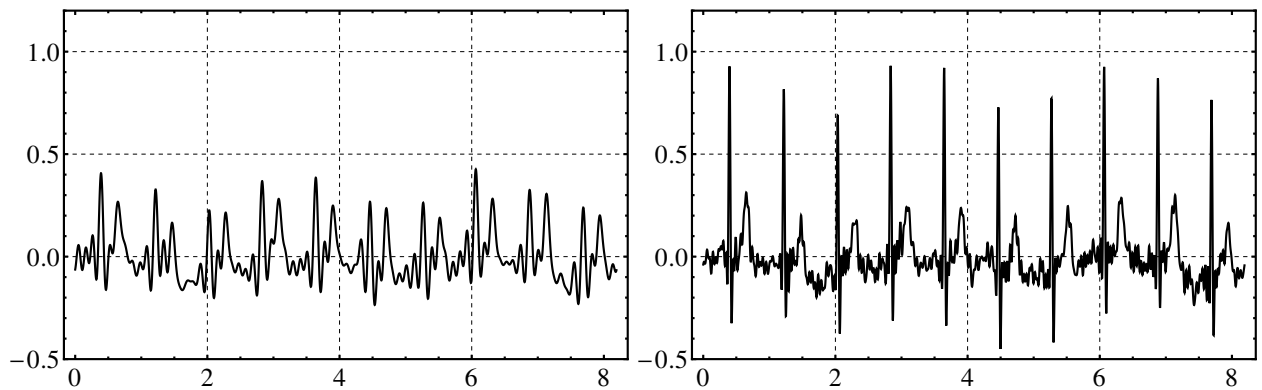
Prisiminkime, kad ryšį tarp k -tojo Furje koeficiento c_k ir konkretaus dažnio, išreikšto hercais, nustatėme 7.1.3 skyrelyje. Tai svarbu, nes dirbant su eksperimentiniais duomenimis, žinios apie informatyvius ir neinformatyvius dažnius kaip taisyklė išreikštos būtent hercais, o ne harmonikų skaičiumi. Tuomet galima taikyti 7.1.3 skyrelyje gautą formulę

$$c_k \text{ atitinka } \frac{k}{T} \text{ Hz} = \frac{k}{N\Delta t} \text{ Hz},$$

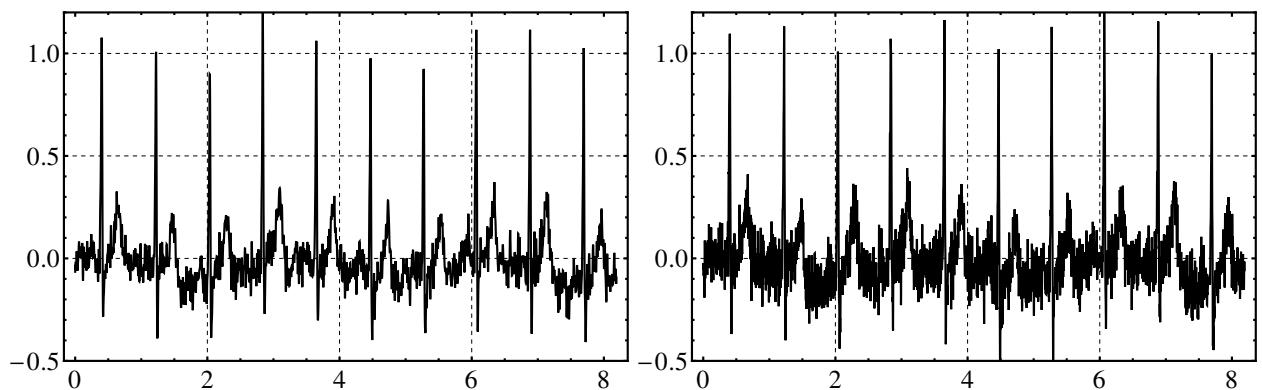
nusakančia kiek hercų atitinka konkretų indeksą k turintis spektro koeficientas c_k . Čia Δt yra signalo reikšmių diskretizacijos žingsnis pagal laiką t (išreikštas sekundėmis).



68 pav.: Elektrokardiogramos V4 kanalo fragmento (žr. 21 pav., grafiką kairėje) originalus signalas (ilustracija kairėje) ir paveiktas triukšmo (ilustracija dešinėje).



69 pav.: Paveikto triukšmo signalo, pavaizduoto 68 pav., iliustracijoje dešinėje, filtravimas Furje filtru (80). Iliustracija kairėje: filtruotas signalas, gautas palikus pirmąsias $k_p = 80$ harmonikų (pašalinti dažniai, viršijantys 9.8 Hz). Iliustracija dešinėje: filtruotas signalas, gautas palikus pirmąsias $k_p = 200$ harmonikų (pašalinti dažniai, viršijantys 24.4 Hz).



70 pav.: Paveikto triukšmo signalo, pavaizduoto 68 pav., iliustracijoje dešinėje, filtravimas Furje filtru (80). Iliustracija kairėje: filtruotas signalas, gautas palikus pirmąsias $k_p = 500$ harmonikų (pašalinti dažniai, viršijantys 61 Hz). Iliustracija dešinėje: filtruotas signalas, gautas palikus pirmąsias $k_p = 1000$ harmonikų (pašalinti dažniai, viršijantys 122 Hz).

Žemus dažnius praleidžiantis Furje filtras (80) praktikoje populiariausias, nes daugumai triukšmų būdingi aukšti dažniai.

Filtravimo taisyklė (80) apibrėžia taip vadinamą „aštrų“ filtravimą, kadangi spektro koeficientai

c_k iš karto užnulinami, vos tik indeksas k viršija kritinę reikšmę k_p . Kartais naudojamas *glodusis filtravimas*, kuriame indeksui k viršijus k_p , koeficientų c_k reikšmės ne iš karto užnulinamos, bet jų modulis $|c_k|$ mažinamas tolydžiai (pavyzdžiui, eksponentiškai), o fazė $\arg c_k$ nekeičiama.

Priklausomai nuo filtravimo tikslų, gali būti taikomos ir kitokios spektro koeficientų filtravimo taisyklės. Pavyzdžiui, *aukštus dažnius praleidžiantis filtras* (angliškai “high-pass filter”), tam tikram intervalui *priklausančius dažnius praleidžiantis filtras* (angliškai “band-pass filter”), arba tam tikram intervalui *priklausančius dažnius pašalinantis filtras* (angliškai “band-reject filter”).

Literatūra

- [1] Richard G. Lyons, Understanding Digital Signal Processing (3rd Edition), Prentice Hall, 2011.
- [2] Yukio Sato, Obrabotka signalov. Pervoje znakomstvo (rusų kalba, vertimas iš japonų kalbos), Dodeka, 1999.
- [3] Steven W. Smith, The Scientist & Engineer's Guide to Digital Signal Processing, California Technical Publishing, 1997, <http://www.dspguide.com/pdfbook.htm>
- [4] Rob J. Hyndman, Time Series Data Library, <https://pkg.yangzhuoranyang.com/tsdl>
- [5] Eastern Michigan University, Free Data Sources, <https://guides.emich.edu/data/free-data>
- [6] DataHub, Search for DataSets, <https://datahub.io/search>
- [7] Gareth Janacek, Practical Time Series Data for Analysis, <http://www.uea.ac.uk/~gj/book/data/datalist.html>
- [8] Wolfram Mathematica Documentation Center, Example Data, <http://reference.wolfram.com/mathematica/ref/format/EDF.html>
- [9] PhysioNet, The Research Resource for Complex Physiologic Signals, <https://physionet.org>
- [10] Sound Jay Sound Data Library, <https://soundjay.com>
- [11] Norbert Marwan, Recurrence Plots and Cross Recurrence Plots, <http://recurrence-plot.tk>
- [12] Norbert Marwan, M. Carmen Romano, Marco Thiel, Jürgen Kurths, Recurrence Plots for the Analysis of Complex Systems, Physics Reports, 438 (2007) 237–329. (Straipsnis PDF formatu yra autorių tinklalapyje [11])
- [13] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes: The Art of Scientific Computing (3rd Edition), Cambridge University Press, 2007.
- [14] FFTW C Subroutine Library, <http://fftw.org>
- [15] GNU Scientific Library (GSL), <http://www.gnu.org/software/gsl>
- [16] Intel® oneAPI Math Kernel Library (Intel® oneMKL), <https://software.intel.com/en-us/intel-mkl>
- [17] AMD Optimizing CPU Libraries (AOCL), <https://developer.amd.com/amd-aocl>