



Projekta Nr.3
**“Oriģinālu signālu apstrādes paņēmieni izveide un
izpēte
konkurētspējīgu IT tehnoloģiju radīšanai”**

INFORMATĪVĀ ATSKAITE

PAR PROJEKTA OTRĀ ETAPA REALIZĀCIJU
(2006. gada aprīlis - jūnijs)

Atskaite sagatavota: 30.06.2006

IZPILDĪTĀJA līguma uzskaites Nr. 1-23/36

PASŪTĪTĀJA līguma uzskaites Nr. 1-23/36

Programmas koordinators: Dr.habil.sc.comp. I. Biļinskis, Valsts aģentūra
„Elektronikas un datorzinātņu institūts”

Projekta vadītājs: Dr. sc. comp. M. Greitāns, Valsts aģentūra „Elektronikas un
datorzinātņu institūts”

Saturs

PROJEKTA OTRĀ ETAPA IZPILDES PAMATOJUMS UN KOPSAVILKUMS.....	4
IEVADS.....	9
1. LĪMEŅU-ŠĶĒRSOJUMA ANALOGS-CIPARU PĀRVEIDOTĀJU ĪPAŠĪBU ANALĪZE	10
1.1. SIGNĀLS-KVANTĒŠANAS TROKSNIS ATTIECĪBA.....	10
1.2. NOLAŠU BLĪVUMS.....	12
1.3. DISKRETIZĀCIJAS LAIKA INTERVĀLI	13
2. SIGNĀLA REKONSTRUKCIJA NO LĪMEŅU ŠĶĒRSOJUMA NOTIKUMA NOLASĒM	15
2.1. EKSISTĒJOŠO METOŽU IZPĒTE, TO TRŪKUMI UN PRIEKŠROCĪBAS.....	15
2.2. SINC FUNKCIJAS PARAMETRU ADAPTĀCIJA ATBILSTOŠI LOKĀLAM NOLAŠU BLĪVUMAM	17
2.3. METOŽU DARBĪBAS MODELĒŠANA UN SALĪDZINOŠĀ ANALĪZE	19
3. NESTACIONĀRU SIGNĀLU APSTRĀDES BEZTAKTĒTĀJA SISTĒMAS MODELĒŠANA	22
3.1. NESTACIONĀRU SIGNĀLU ANALOGS-CIPARU PĀRVEIDOŠANA IZMANTOJOT LĪMEŅU-ŠĶĒRSOJUMA PRINCIPU	22
3.2. LAIKA-FREKVENČU REPREZENTĀCIJU MODELĒŠANA	23
3.2.1. Īssintervāla Furjē transformācijas pieeja.....	23
3.2.2. Vilnīša analīzes pieeja	25
3.2.3. Vignera sadalījuma funkcija	26
4. DFT-VEIDA PĀRVEIDOJUMU IZSTRĀDE, KAS ATBILST UN ŅEM VĒRĀ AR LŠ-ACP IEGŪTO LAIKĀ NEVIENMĒRĪGO NOLAŠU SPECIFIKU.....	29
4.1. VISPĀRINĀTĀ DFT IZTEIKSME.....	29
4.2. DFT UZLABOJUMS AR SIGNĀLA NOLAŠU INTERPOLĀCIJAS PALĪDZĪBU.....	31
4.3. DFT UZLABOJUMS AR ATJAUNOŠANAS KĻŪDAS NOLAŠU INTERPOLĀCIJAS PALĪDZĪBU.....	33
4.4. UZLABOTO DFT-VEIDĪGO PĀRVEIDOJUMU MODELĒŠANAS REZULTĀTI.....	34
5. DASP VIRTUĀLO INSTRUMENTU ATTĪSTĪŠANA IZMANTOJOT MODIFICĒTOS ALGORITMUS.....	35
5.1. PLATJOSLĪGA UN PLAŠA DINAMISKĀ DIAPAZONA CIPAROTĀJA VADĪBA LABVIEW VIDĒ.....	35
5.2. JAUNRADĪTO SIGNĀLU APSTRĀDES ALGORITMU IMPLEMENTĀCIJA LABVIEW	36
5.3. GRAFISKO LIETOTĀJU INTERFEISU IZVEIDE	37
6. ASINHRONO MIKROPROCESORU PIELIETOJAMĪBA AR BEZTAKTĒTĀJA ANALOGS-CIPARU PĀRVEIDOTĀJU IEGŪTU DATU ENERGO-EKONOMISKAI APSTRĀDEI.....	38
6.1 ENERĢIJAS PATĒRIŅA MINIMIZĀCIJAS MODEĻI.....	38
6.2 ASINHRONĀ PROCESORA ENERĢIJAS TAUPĪBAS IESPĒJAS.....	39
6.3 ASINHRONO PROCESORU RAKSTUROJUMI UN LIETOJUMS.....	40
7. PĒTĪJUMI DATU APSTRĀDES SISTĒMU REALIZĀCIJAI BEZTAKTĒTĀJA ELEKTRONISKĀS IEKĀRTĀS.....	42
7.1 TRANSFORMĀCIJU AR TAISNSTŪRA FUNKCIJĀM PIELIETOJUMS.....	42
7.2 GSM SIGNĀLU APSTRĀDES BEZTAKTĒTĀJA SISTĒMAS MODELIS	45
IZMANTOTĀS LITERATŪRAS SARAKSTS.....	48

1. PIELIKUMS. PUBLIKĀCIJA ŽURNĀLĀ „ELECTRONICS AND ELECTRICAL ENGINEERING”, ISSN 1392-1215, NO. 4 (68), 2006, M. GREITANS, I. HOMJAKOVŠ, ENHANCED DIGITAL SIGNAL PROCESSING OF SIGNAL-DEPENDENTLY SAMPLED SIGNALS.

2. PIELIKUMS. PUBLIKĀCIJA ŽURNĀLĀ „ELECTRONICS AND ELECTRICAL ENGINEERING”, ISSN 1392-1215, NO. 4 (68), 2006, A. BAUMS, ENERGY CONSUMPTION OPTIMISATION IN HARD REAL-TIME SYSTEM CMOS PROCESSORS.

3. PIELIKUMS. PREZENTĀCIJA KONFERENCĒ „ELECTRONICS'2006, THE 10TH INTERNATIONAL CONFERENCE. KAUNAS UNIVERSITY OF TECHNOLOGY, MAY 24, 2006” M. GREITANS, M. KALBERGS, ABILITIES AND APPLICATION OF LEVEL-CROSSING SAMPLING IN DIGITAL SIGNAL PROCESSING.

4. PIELIKUMS. PUBLIKĀCIJA KONFERENCES RAKSTOS, INTERNATIONAL REMOTE ENGINEERING VIRTUAL INSTRUMENT SYMPOSIUM, „REV2006”, 29-30 JUNE 2006, MARIBOR, SLOVENIA. M. GREITANS, VIRTUAL INSTRUMENTATION IN AN EXTENDED FREQUENCY RANGE USING DIGITAL ALIAS-FREE SIGNAL PROCESSING.

5. PIELIKUMS. PUBLIKĀCIJA KONFERENCES RAKSTOS, INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND MULTIMEDIA APPLICATIONS, „SIGMAP2006” M. GREITANS. PROCESSING OF NON-STATIONARY SIGNAL USING LEVEL-CROSSING SAMPLING

6. PIELIKUMS. MAĢISTRA DARBA ANOTĀCIJA. R. ŠĀVELIS „SIGNĀLA LAIKFORMAS ATJAUNOŠANA NO LĪMEŅU ŠĶĒRSOJUMA NOTIKUMU NOLASĒM”.

7. PIELIKUMS. IZSTRĀDĀTĀS DATORPROGRAMAS.

- A LĪMEŅU-ŠĶĒRSOJUMA ANALOGS-CIPARU PĀRVEIDOTĀJU ĪPAŠĪBU ANALĪZEI..
- B SIGNĀLA REKONSTRUKCIJAI NO LĪMEŅU ŠĶĒRSOJUMA NOTIKUMA NOLASĒM.
- C NESTACIONĀRU SIGNĀLU APSTRĀDES BEZ TAKTĒTĀJA SISTĒMAS MODELĒŠANAI..
- D IZSTRĀDĀTO DFT – VEIDĪGO PĀRVEIDOJUMU MODELĒŠANAI.
- E PĒTĪJUMIEM DATU APSTRĀDES SISTĒMU REALIZĀCIJAI BEZTAKTĒTĀJA ELEKTRONISKĀS IEKĀRTĀS.

Projekta otrā etapa izpildes pamatojums un kopsavilkums

Valsts pētījumu programmas mērķis ir veidot Latvijas informātikas industrijas zinātnisko bāzi, dot būtisku ieguldījumu (pētījumus un izstrādes) jaunāko informācijas tehnoloģiju radīšanā, attīstībā un piemērošanā Latvijas vajadzībām, sagatavot augstākās kvalifikācijas speciālistus un iepludināt tos Latvijas IT darba tirgū, kāpināt Latvijas IT industrijas konkurētspēju pasaules tirgū, izstrādāt valsts nozīmes informatizācijas projektu zinātnisko pamatu būtisku tautsaimniecības problēmu risināšanai, palielināt Latvijas zinātnieku konkurētspēju ES projektu izcīnīšanā IT jomā.

Projekta mērķis ir oriģinālu signālu diskrētās apstrādes paņēmienu izveide un to darbības izpēte, kas integrācijā ar modernām elektroniskām tehnoloģijām dod iespējas rast aktuālu IT problēmu risinājumus un ļauj veidot uz zinātņu ietilpības rēķina konkurētspējīgas elektroniskās iekārtas.

Projekta 2. etapa „Darba uzdevumā” definētie uzdevumi:

1. Līmeņu-šķērsojumu analogs-ciparu pārveidotāju (LŠ-ACP) īpašību analīze, un šādā veidā pārveidotu signālu rekonstrukcijas paņēmienu izstrāde.
2. Nestacionāru signālu apstrādes beztaktētāja sistēmas modeļa izveide, un iegūto ciparu signālu laika-frekvenču raksturošanas datorprogrammu izveide.
3. DASP teorijas un uz tās bāzes veidotu virtuālo instrumentu attīstīšana, radot DFT-veidīgus pārveidojumus, kas atbilstu un ņemtu vērā neregulāri izvietotas nolases (tajā skaitā iegūtas ar līmeņu-šķērsošanas diskretizāciju).
4. Jaunradīto datu apstrādes paņēmienu realizācijas asinhronās elektronikas sistēmās un/vai mikroprocesoros iespēju izpēte.
5. Izpildes laikā iegūtos rezultātus prezentēt starptautiskās zinātniskās konferencēs un publicēt LZP sarakstā iekļautajos zinātniskajos žurnālos (izdevumos).

Projekta 2. etapa definēto uzdevumu izpildes rezultāti:

1. Paveiktos darbus un 1.uzdevuma izpildes rezultātus nosacīti var iedalīt divās grupās:
1) līmeņu-šķērsojuma analogs-ciparu pārveidotāja īpašības; 2) rekonstrukcija no signāla nolasēm, kas iegūtas ar šāda veida ACP. Attiecībā uz pirmo daļu ir veikta līmeņu-šķērsojuma ACP īpašību padziļināta analīze. Parādīts, kādā veidā šāda tipa pārveidojuma signāls-(kvantēšanas)troksnis attiecība ir atkarīga no ieejas signāla statistiskajiem parametriem un no līmeņu-šķērsojuma laika momentu fiksēšanas precizitātes, kuru uzlabojot iespējams attiecīgi palielināt sasniedzamo SkTA. Iegūtas izteiksmes, kas saista līmeņu-šķērsojuma ACP veidoto nolašu blīvumu ar signāla spektrālajām īpašībām divu veida signālu gadījumos: sinusoidāls signāls, frekvenču joslā ierobežots Gausa troksnis ar konstantu jaudas spektrālo blīvumu. Parādītas iespējas novērtēt laika intervālus starp signāla nolasēm sinusoidālam signālam un laika intervālu robežas vispārīga signāla gadījumā. Iegūtie rezultāti prezentēti konferencē “Electronics'2006” (Kauņa, Lietuva), izmantoti publikācijai, kas akceptēta prezentācijai konferencē „Signal processing and multimedia applications SIGMAP'06” (Setubala, Portugāle). To zinātniskā un tautsaimnieciskā nozīmība raksturojas ar faktu, ka iepazīstot un apzinot LŠ-ACP īpašības, tiek palielināta to praktisko pielietojumu iespējamība, kas nozīmē būtisku soli uz beztaktētāja elektronisko risinājumu veidošanu.

Attiecībā uz 1. uzdevuma otro daļu ir izpētītas signāla rekonstrukcijas iespējas no tā līmeņu šķērsojuma notikuma nolasēm. Aplūkota un modelēta (izstrādāta datorprogrammu kopa) literatūrā atrodamo (sinc, splainu un dažāda paveida iteratīvo (Voronova, lineārā, Marvasti, adaptīvo svaru)) metožu, kas radītas apzināti neregulāri diskretizētu signālu atjaunošanai, darbība. Piedāvātas oriģinālas rekonstrukcijas metodes, kas signāla atjaunošanu veic ar uzlabotu sinc-funkciju palīdzību, kuru parametri (amplitūda, nobīde un mērogošana) tiek adaptēti lokāliem signāla līmeņu šķērsojuma notikumu laika momentiem. Apskatīti divi gadījumi – atjaunojošās funkcijas tiek veidotas kā simetriskas un kā asimetriskas. Jaunradītās metodes ir pielāgotas specifiskajam ar LŠ-ACP neregulāri iegūto nolašu apstrādes gadījumam un ļauj uzlabot signālu rekonstrukcijas kvalitāti. Iegūtie rezultāti izmantoti R. Šaļeļa maģistra

darbā, kas RTU ETF tika aizstāvēts ar vērtējumu “teicami”. Izpildītā uzdevuma zinātniskā un tautsaimnieciskā nozīmība pamatojas ar to, ka signāla rekonstrukcija ir viens no pašiem fundamentālākiem uzdevumiem signālu apstrādē, jo pēc korektas signālu rekonstrukcijas ir iespējams analīzei pielietot praktiski visus citus zināmos paņēmienu. Piemēram, konkrētā gadījumā ir iespējams LŠ-ACP nevienmērīgi izvietotās nolases transformēt uz klasisko ACP nolašu kopu, un tad veikt apstrādi ar zināmām, vienmērīgai diskretizācijai pielietojamām metodēm.

2. Izmantojot 1. etapā uzsāktās iestrādes nevienmērīgi diskretizētu nestacionāru signālu apstrādē, ir izveidots beztaktētāja sistēmas modelis nestacionāru signālu apstrādei, kas balstās uz līmeņu-šķērsojuma analogs-ciparu pārveidojuma principu datu ieguvei. Radīta datorprogrammu kopa, kas modelē trīs populārāko nestacionāru signālu analīzes pieeju (īssintervāla Furjē transformācijas, vilnīšanalīzes un Vignera sadalījuma) darbību, un veikti pētījumi to pielietojamībai līmeņu-šķērsojuma nolašu apstrādei. Iegūti rezultāti, kas parāda katra paņēmienu priekšrocības un trūkumus, tādejādi norādot virzienus turpmākiem pētījumiem. Uzsāktas iestrādes oriģinālu paņēmienu izveidei, kas ļautu Vignera sadalījuma pieeju izmantot LŠ-ACP datu laika-frekvenču reprezentāciju iegūšanai, jo tiešā veidā šī pieeja ir izmantojama tikai vienmērīgi diskretizētiem signāliem, un literatūrā līdz šim nav atrodamas tās paplašināšanas veidi arī nevienmērīgas diskretizācijas gadījumam. Iegūtie rezultāti izmantoti publikācijas sagatavošanai, kura akceptēta prezentācijai konferencē „Signal processing and multimedia applications SIGMAP'06” (Setubal, Portugāle). To zinātnisko un praktisko nozīmību nosaka fakts, ka reālas dabas, piemēram, multimediju signāli pēc savas būtības ir nestacionāri. No tautsaimniecības viedokļa nozīmība ir potenciālā multimediju iekārtu izstrādē, tajā skaitā SoC (*System on chip*) tehnoloģijā, kuras spētu izmantot asinhrono elektronisko sistēmu priekšrocības.

3. Paveiktos darbus un iegūtos rezultātus 3. uzdevuma izpildē nosacīti var iedalīt divās grupās. Pirmā attiecas uz DASP virtuālo instrumentu attīstīšanu, bet otrā uz uzlabotu DFT-veidīgu algoritmu izstrādi, kurus perspektīvā var izmantot minētajos instrumentos. Attiecībā uz DASP teorijas balstītu virtuālo instrumentu attīstīšanu ir uzsāktas iestrādes jaunas paaudzes DASP virtuālo instrumentu radīšanai, kas orientētos uz National Instruments programmēšanas rīka LabVIEW izmantošanu. Tā ir grafiskas programmēšanas vide, kurā veidotas instrumentu datorprogrammas gala lietotājs var viegli adaptēt un papildināt konkrētu uzdevumu vajadzībām bez īpašām programmēšanas zināšanām. Virtuālo instrumentu izveides sekmīgumu nosaka trīs aspekti: 1) iespējas sadarbībai ar aparatūras portiem, 2) efektīva apstrādes algoritmu implementācija un 3) ērta vide lietotāja grafisko interfeisu radīšanai. Ir izpētītas un praktiski apgūtas iespējas LabVIEW sadarbībai ar personālā datora aparatūras portiem un interfeisiem, izveidota programmatūra DASP ciparotāja vadībai un kontrolei caur EPP interfeisu. Signālu apstrādes implementēšanai ir izveidoti paņēmienu no ciparotāja nolasīto un apstrādei paredzēto datu noformēšanai masīvu un matricu struktūrās, kā arī iegūta praktiskā darba pieredze matricu un/vai masīvu algebrā formalizētu algoritmu realizācijai LabVIEW bloku diagrammās. Izmantojot LabVIEW piedāvātās iespējas lietotāju grafisko interfeisu veidošanai, ir radīti instrumentu paneļu prototipi. Šīs uzdevuma daļas izpildes laikā iegūtie rezultāti ir prezentēti konferencē „Remote engineering and virtual instrumentation REV'06” (Maribova, Slovēnija) un publicēti konferences rakstos. To nozīmība varbūt ir mazāk zinātniska, bet ļoti vērtīga no praktiskā pielietojuma viedokļa, jo parāda principiālās iespējas un veic iestrādes jaunas paaudzes DASP virtuālo instrumentu veidošanai, kuru ražošana var būt interesanta Latvijas IT industrijai.

Attiecībā uz 3. uzdevuma otro daļu ir izstrādāta virkne DFT-veidīgu pārveidojumu, kas padziļināti ņem vērā signāla nevienmērīgi izvietotu nolašu laika momentus. Pārveidojumu rezultāti pilnībā atbilst DFT būtībai, kas nozīmē to, ka tiem ir pielietojams inversais DFT pārveidojums signāla atjaunošanai laika prezentācijā. Piedāvāto modifikāciju pamatideja ir minimizēt signāla atjaunošanas kļūdu ņemot vērā ne tikai tās vērtības signāla nolašu laika

momentos, bet arī starp tiem ar tik pat lielu svarīgumu. Lai to paveikti ir izvēlēti divi risinājuma ceļi: 1) signāla nolašu interpolācija, 2) atjaunošanas kļūdas nolašu interpolācija. Iegūtas izteiksmes gadījumiem, kad signāls tiek interpolēts ar nulltās kārtas un pirmās kārtas polinomiem, kā arī, ja pielietota frekvenču joslā ierobežota interpolācija ar sinc funkcijām. Kļūdas interpolācija aprakstīta gadījumiem, kad to veic ar nulltās un pirmās kārtas polinomiem. Piedāvātie algoritmi simulēti, rezultāti apkopoti un salīdzināti, ja piedāvātie uzlabojumi ir pielietoti ar LŠ-ACP iegūtiem datiem. Iegūtie rezultāti prezentēti konferencē *Electronics'2006*” (Kauņa, Lietuva) un publicēti žurnālā *„Electronics and Electrical Engineering”*. To zinātniskā nozīmība saistīta ar faktu, ka uzlabotās DFT-veidīga pārveidojuma izteiksmes padziļināti ņem vērā nolašu nevienmērīgumu un tādā veidā kompensē neortogonalitāšu radītos artifaktus. Potenciālā tautsaimnieciskā nozīmība ir saistīta ar pielietojumu apzināti un/vai neapzināti nevienmērīgi diskretizētu datu apstrādes sistēmās.

4. Turpinot jau pirmajā projekta etapā iesākto darbību, kas vērsta uz jaunradīto signālu apstrādes pieeju virzību realizācijai praktiskās elektroniskās iekārtās, šajā projekta etapā ir izdalījušies divi galvenie novirzieni: 1) implementācija ar universālu mikroprocesoru palīdzību un 2) implementācija ar specializēto mikroshēmu palīdzību. Pirmajā novirzienā ir veikti darbi saistībā ar izpēti potenciāliem ieguvumiem, ko var dot asinhrono mikroprocesoru izmantošana, īpašu vērtību pievēršot energo ekonomiskumam. Ir izstrādāti enerģijas patēriņa minimizācijas modeļi un iegūti rezultāti, kas ļauj novērtēt, ka izmantojot asinhronu ACP un asinhronu datu apmaiņu ar mikroprocesoru ekonomija var būt aptuveni 5 reizes. Ir apkopoti jau pieejamo asinhrono mikroprocesoru raksturojumi un lietojuma īpatnības, kā arī apzinātas to enerģijas patēriņa taupīšanas iespējas. Iegūtie rezultāti daļēji izmantoti publikācijai žurnālā *„Electronics and Electrical Engineering”* un prezentācijai konferencē *„Electronics'2006”* (Kauņa, Lietuva). Paveiktais ir vēl tikai pašā sākuma fāzē un tā zinātnisko un tautsaimniecisko nozīmību vairāk rādīs darbu turpinājums nākamajos etapos. Tomēr ir jāatzīmē, ka tās potenciāls ir ārkārtīgi liels, jo var dod risinājumus kādā veidā iespējams radīt īpaši maz tērējošas iekārtas, kas ir svarīgi, gan mobilo iekārtu darbības paildzināšanai starp uzlādēm, gan kosmosa, gan militāros pielietojumos.

Otrā novirzienā, kas saistīts ar specializēto mikroshēmu izveidi, ir turpināti pirmā etapā iesāktie pētījumi un čipu projekti, kas saistīti ar taisnstūra funkciju pielietošanu elektroniskā aparatūrā reālā-laika apstrādes pārveidojumos. Ir apskatīts efekts, ko dod taisnstūra funkciju samēra veidošana pēc amplitūdā vienmērīga sadalījuma principa. Kā praktiska pielietojuma piemērs ir izveidots GSM signālu apstrādes beztaktētāja sistēmas modelis. Izveidotas datorprogrammas, kas simulē datu bitu pārraidi. Šobrīd ir iegūti rezultāti, kas parāda sistēmas darbību viena datu kanāla raidīšanas gadījumā bez un ar aditīva trokšņa klātbūtni. Rezultātu nozīmība slēpjas faktā, ka šobrīdējie pētījumi ļauj izteikt pieņēmumu, ka pielietojot piedāvāto sistēmas modeli būtu iespējams datu pārraidi veikt ar līdz pat vairāku simtu reižu lielāku caurlaidību, nekā to nosaka esošais GSM standarts. Tomēr, lai objektīvi novērtētu risinājumu praktisko pielietojamību ir jāveic papildus pētījumi un modelēšana, kā nākamo etapu minot vairāku kanālu vienlaicīgas darbības režīmu.

5. Projekta 2. etapa darba uzdevumu izpildē iegūtie rezultāti ir izmantoti sekojošu publikāciju, maģistra darba un konferenču prezentāciju veidošanā:

1. M. Greitans, I. Homjakovs, *Enhanced Digital Signal Processing of Signal-Dependently Sampled Signals*. „ELECTRONICS AND ELECTRICAL ENGINEERING”, ISSN 1392-1215, No. 4 (68), 2006 lpp. 9-14.
2. A. Baums, *Energy consumption Optimisation In Hard Real-Time System CMOS Processors*. „ELECTRONICS AND ELECTRICAL ENGINEERING”, ISSN 1392-1215, No. 4 (68), 2006, lpp. 19-22.
3. M. Greitans, *Virtual instrumentation in an extended frequency range using Digital Alias-free Signal Processing*. Proc. of International Remote Engineering Virtual Instrument Symposium, „REV2006”, Maribor, Slovenia, 29-30. Jūnijs 2006.g. (published on CD).

4. M. Greitans. *Processing of Non-Stationary Signal Using Level-Crossing Sampling*. International Conference on Signal Processing and Multimedia Applications, „SIGMAP2006” Setubal, Portugāle, 7-10. Augusts 2006.g. (akceptēts konferencei).
5. R. Šāvelis, *Signāla laikformas atjaunošana no līmeņu šķērsojuma notikumu nolasēm*. Maģistra darbs, RTU Elektronikas un telekomunikāciju fakultāte, Radioelektronikas institūts, Rīga, 2006.g., 70.lpp.
6. M. Greitans, M. Kalbergs, *Abilities and application of level-crossing sampling in digital signal processing*. Prezentācija konferencē „ELECTRONICS'2006, the 10th international conference”, Kaunas University of Technology, 24. Maijs, 2006.g.

Kopsavilkums: Projekts ir vērst uz jaunu signālu apstrādes paņēmienu radīšanu, kas kopā ar modernajām elektroniskām tehnoloģijām piedāvātu risinājumus uz zinātņu ietilpības rēķina konkurētspējīgu elektronisko iekārtu ražošanai. Projekta pirmajā etapā tika uzsākti pētījumi un veiktas iestrādes, kuru pamatā ir ideja līdzšinējās zināšanas un pieredzi DASP (*Digital Alias-free Signal Processing*) signālu apstrādes jomā saistīt ar beztaktētāja elektronisko iekārtu projektēšanas principiem. Tika apzinātas asinhronu elektronisko sistēmu priekšrocības – pazemināts enerģijas patēriņš, samazināta siltuma izdalīšana, automātiska adaptācija pie fizikālo apstākļu maiņas, neliels elektromagnētiskais izstarojums u.c. Taču, lai signālu apstrādes sistēmās varētu gūt labumu no šīm priekšrocībām, ir nepieciešams analogs-ciparu pārveidojumu vadīt nevis kā klasiskajā gadījumā ar taksimpulsiem, bet gan ar pašu signālu. Tas noved pie laikā specifiski nevienmērīgi izvietotām nolasēm un prasa radīt atbilstošas apstrādes metodes. Pirmajā etapā tika aplūkoti un modelēti veidi kā beztaktētāja režīmā iespējams veikt analogs-ciparu pārveidojumus, kas otrajā etapā ir turpināts padziļināti izpētīt līmeņa-šķērsojuma diskretizācijas īpašības. Pirmajā etapā tika apzinātas iespējas iegūto signālu apstrādei pielietot esošās nevienmērīgi diskretizētu signālu apstrādei radītās metodes, kas otrajā etapā ir turpināts izpētīt rezultātus attiecībā uz signālu rekonstrukciju. Ir uzsāktas iestrādes jaunu uzlaboti paņēmienu radīšanai, kas izmantojot apgūtās LŠ-ACP īpatnības spētu uzlabot rekonstrukcijas kvalitāti. Izstrādes šajā virzienā ir plānots turpināt arī nākamajos etapos. Iepriekšējā etapā notika darbs pie nevienmērīgi diskretizētu nestacionāru signālu analīzes, kas šajā etapā ir turpināts pievērsties gadījumam, ja ciparu signāla nolases tiek iegūtas kā līmeņu šķērsojuma notikumi. Šajā jomā pētījumu turpinājums būtu saistāms ar Vignera sadalījuma pieejas pielietojamību nevienmērīgi diskretizētam signālam.

Samērā liels akcents projekta ietvaros ir uz jaunradāmo oriģinālo ciparu signālu apstrādes paņēmienu saistību ar to realizāciju praktiskos risinājumos. Pirmajā etapā tika uzsāktas iestrādes saistībā ar taisnstūra funkciju izmantošanu un specializētu mikroshēmu projektēšanu. Šis novirziens ir turpināts otrajā etapā kā piemēru izvēloties GSM signālu pārraides simulēšanu. Papildus tam ir uzsākta darbošanās pie cita novirziena – vispārēja pielietojuma asinhronu mikroprocesoru izmantošanas. Apzināti pieejamie procesoru risinājumi un turpmākos etapos plānots tos asinhroni saistīt ar beztaktētāju ACP un šādā veidā veidot signālu apstrādes mezglus. Kā atsevišķs novirziens ir izceļams DASP virtuālo instrumentu attīstība, kurā šajā etapā ir apgūti trīs stūrakmeņi sekmīgai virtuālo instrumentu radīšanai LabVIEW vidē: sadarbība ar aparatūras interfeisiem, signālu apstrādes algoritmu realizācija masīvu un/vai matricu formulējumā, kā arī lietotāju grafisko interfeisu izveide. Nākamajos etapos uz šo iestrāžu pamata ir plānots implementēt jaunradītos algoritmus un turpināt darbu uz praktisku instrumentu radīšanu.

Konkrētais projekta otrā etapa izpildes mērķis bija pirmā etapā nosprausto ideju par asinhronu sistēmu priekšrocību un nevienmērīgi diskretizētu signālu apstrādes zināšanu sakopošanu jaunu konkurētspējīgu risinājumu izveidei virzīt uz praktiski pielietojamu metožu un algoritmu, kā arī implementācijas paņēmienu veidošanu.

Strādājot pie mērķa sasniegšanai nospraustajiem uzdevumiem ir iegūti sekojoši galvenie rezultāti:

- Iegūtas svarīgāko LŠ-ACP īpašību demonstrējošas izteiksmes,

- Izveidota datorprogrammu kopa nevienmērīgi diskretizētu signālu rekonstrukcijai ar pazīstamākām metodēm.
- Piedāvāti uzlabojumi rekonstrukcijai ar sinc-funkcijām LŠ-ACP nolašu gadījumam.
- Izveidota datorprogrammu kopa nevienmērīgi diskretizētu nestacionāru signālu laika-frekvenču reprezentāciju iegūšanai.
- Iegūtas uzlabotas izteiksmes DFT-veidīgu pārveidojumu veikšanai, kas padziļināti ņem vērā nolašu nevienmērīgumu.
- Apgūti paņēmienu un izveidoti prototipi virtuālo instrumentu izstrādei LabVIEW vidē.
- Izanalizētas iespējas un izveidoti modeļi enerģijas taupīgu datu apstrādes sistēmu izveidei uz vispārēja pielietojuma mikroprocesoru bāzes.
- Izveidots beztaktētāja sistēmas modelis telekomunikācijas signālu apstrādei ar potenciālu realizāciju specializētās mikroshēmās.
- Etapā iegūtie galvenie rezultāti ir pieejami sešās publikācijās.

Saistībā ar projekta etapā iegūtiem rezultātiem ir un sagatavotas četras publikācijas un ziņots divās starptautiskās konferencēs, kurās nolasīti četri referāti. Augusta mēnesī notiks vēl viena prezentācija konferencē, kas veltīta signālu apstrādei un multimediju pielietojumiem.

Apkopojot visu iepriekš rakstīto var apgalvot, ka projekta otrajam etapam nospraustie uzdevumi ir pilnībā izpildīti. Paveiktais darbs ir uzskatāms no vienas puses kā sekmīgs turpinājums iepriekšējā etapā uzsāktām iestrādēm, bet no otras puses kā daudzsološas iestrādes turpmākiem pētījumiem un izstrādēm. Galvenie uzdevumiem nākamajā etapā vajadzētu būt:

- Uzlabotu signāla rekonstrukcijas paņēmienu izstrāde, kas ņem vērā analogs-ciparu pārveidojuma īpatnības.
- Nestacionāru signālu apstrādes metožu attīstīšana nevienmērīgi diskretizētiem signāliem, uzmanību pievēršot reālu multimediju signālu analīzei.
- Iespēju izpēte signālapstrādes algoritmu implementācijai LabVIEW vidē un to realizācija.
- Asinhronu datu apstrādes sistēmu veidošana uz mikroprocesoru un specializēto mikroshēmu bāzes.

Papildus atskaitē minētajam ir jāatzīmē arī fakts, ka projekta izpildē ir iesaistījušies vairāki jaunie speciālisti. Trīs no tiem ir diplomētie inženieri, kas turpina mācības RTU elektronikas specialitātes maģistratūrā, viens šogad ar teicamu vērtējumu aizstāvēja maģistra darbu, kura tēma ir pilnībā saistīta ar šī valsts pētījuma projekta nostādņēm. Vēl viens telekomunikāciju maģistratūras students tika iesaistīts konkrētu uzdevumu risināšanā saistībā ar nestacionāriem signāliem. Tas atbilst Valsts pētījumu programmas mērķim par jaunu augstas kvalifikācijas speciālistu sagatavošanu. Pozitīvi vērtējams arī fakts, ka maģistru darba tēmas tiek saistītas ar aktuālu, mūsdienīgu problēmu izpēti.

levads.

Šī atskaite ir veltīta projekta otrajā etapā paveiktajiem darbiem, kas ir likumsakarīgs turpinājums pirmajā etapā izvirzītām idejām ar mērķi padarīt tās tuvākas praktiskiem pielietojumiem.

Galvenie darbu virzieni ir saistīti ar

- nolašu apstrādi, kuras iegūtas ar līmeņu šķērsojuma analogs-ciparu pārveidotāju, kā vienu no pamatlīdzekļiem datu iegūšanai priekš asinhronām sistēmām (1.nodaļa),
- signālu laikformas atjaunošanu, kā vienu no fundamentālākiem uzdevumiem ciparu signālu apstrādē (2. nodaļa),
- nestacionāru signālu apstrādi, jo reālie signāli pēc savas dabas tādi ir un arī signālvadīti pārveidojumi (piem., LŠ-ACP) labi adaptējas šim nosacījumam (3. nodaļa),
- DASP teorijas un paņēmienu attīstīšanu (4. nodaļa),
- iestrādēm kā jaunpiedāvātos signālu apstrādes paņēmienu var adaptēt praktiskiem pielietojumiem trīs apakšnovirzienos:
 - virtuālo instrumentu sistēmas (5. nodaļa),
 - iekārtu veidošana ar vispārēja pielietojuma mikroprocesoriem (6. nodaļa),
 - iekārtu veidošana radot specializētas mikroshēmas (7. nodaļa).

1. Līmeņu-šķērsojuma analogs-ciparu pārveidotāju īpašību analīze

Iepriekšējā projekta etapā tika izveidots līmeņu-šķērsojošā analogs-ciparu pārveidotāja modelis MATLAB vidē. Noskaidrotas līmeņu-šķērsojošā ACP priekšrocības un trūkumi salīdzinājumā ar klasisko ACP. Turpinot iesāktu darbu šai projekta etapā ir izpētītas līmeņu-šķērsojuma ACP galvenās īpašības: signāls-kvantēšanas troksnis attiecība un kvantēšanas kļūdas sadalījums, nolašu blīvums, laika intervāli starp divām nolāsēm.

1.1. Signāls-kvantēšanas troksnis attiecība

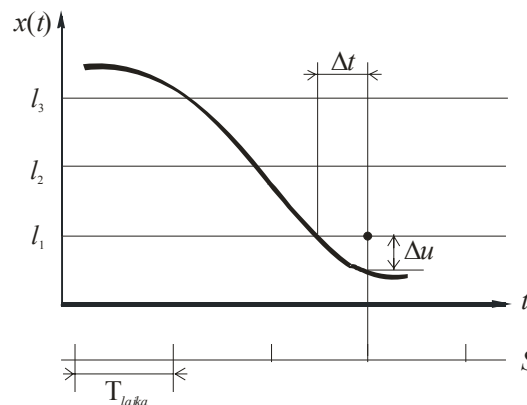
Klasiskajiem ACP tiek izmantota *sample-and-hold* princips, pēc kura seko kvantēšanas vai noapaļošanas process. Tā kā ir ierobežots bitu skaits, tad nav iespējams izslēgt noapaļošanas kļūdu, tāpēc klasiskajiem ACP ir ieviests jēdziens attiecība starp signāla un kvantēšanas trokšņa jaudām (Signal-to-(quantization)Noise Ratio, SNR). Klasiskiem ACP signāls-kvantēšanas troksnis attiecību var noteikt kā [13]

$$SNR_{dB} = 1,76 + 6,02 \cdot n, \quad (1.1)$$

kur n – ACP bitu skaits.

Kā var redzēt pēc formulas (1.1), tad klasiskajā ACP signāls-kvantēšanas troksnis attiecība ir atkarīgs tikai no ACP bitu skaita. Ja ir nepieciešams uzlabot signāls-kvantēšanas troksnis attiecību atliek tikai palielināt bitu skaitu.

Līmeņu-šķērsojuma analogs-ciparu pārveidotāja gadījumā nolases tiek noteiktas tai laika momentā, kad ieejas signāls šķērso noteiktu līmeni, tas ir, signāla amplitūda ir zināma precīzi. Ņemot vērā to faktu, ka nolases atrodas dažādos attālumos viena no otras, tad ir nepieciešams zināt nolases laika momentus. Praksē laika momentus mēra ar taimera palīdzību, kuram ir noteikta precizitāte T_{laika} (Skat. zīm. 1.1).



Zīm. 1.1. Līmeņu-šķērsojošā ACP darbības princips.

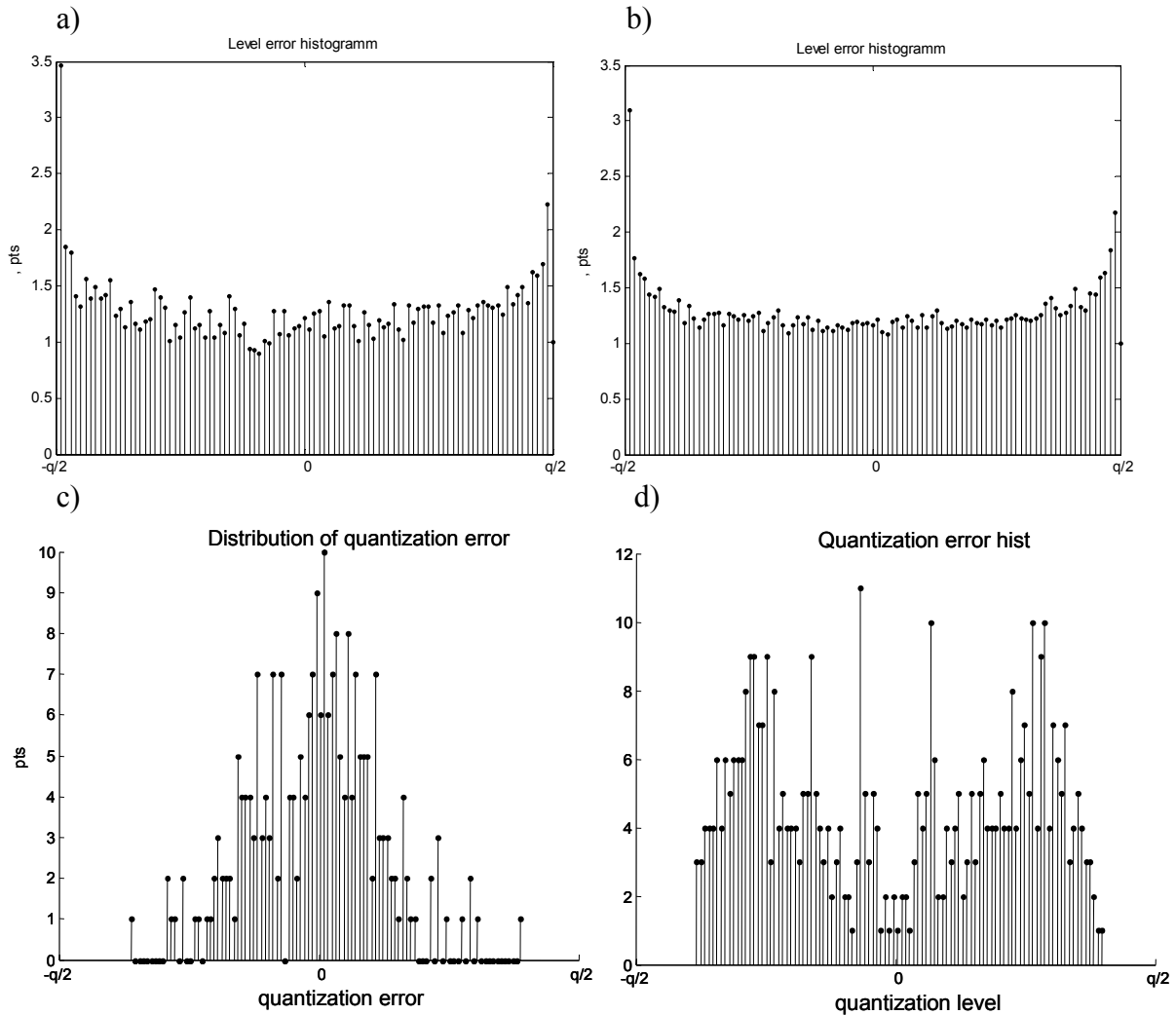
Kā var redzēt Zīm. 1.1. signāls krusto līmeni l_1 starp taimera laika momentiem, bet nolasei tiek piešķirts tuvākais laika moments. Šāds algoritms noved pie laika kļūdas Δt , kā arī signāla izmaiņas dēļ rodas amplitūdas nobīde Δu . Signāls-kvantēšanas troksnis attiecība līmeņu-šķērsojuma ACP gadījumā ir [11]

$$SNR_{dB} = 10 \log \left(\frac{3P_x}{P_{x'}} \right) + 20 \log \left(\frac{1}{T_{laika}} \right), \quad (1.2)$$

kur P_x – ieejas signāla jauda, $P_{x'}$ – ieejas signāla jaudas atvasinājums.

No formulas var redzēt, ka signāls-kvantēšanas troksnis nav atkarīgs no bitu skaita, bet gan no taimera precizitātes un ieejas signāla īpašībām. Lai uzlabotu signāls-kvantēšanas troksnis attiecību ir nepieciešams palielināt taimera precizitāti.

Klasiskā ACP gadījumā līmeņu kļūdas sadalījums ir konstants, tas ir, kļūdas varbūtība ir vienlīdz iespējama starp kvantēšanas līmeņiem (Skat. Zīm. 1.2 a un b).



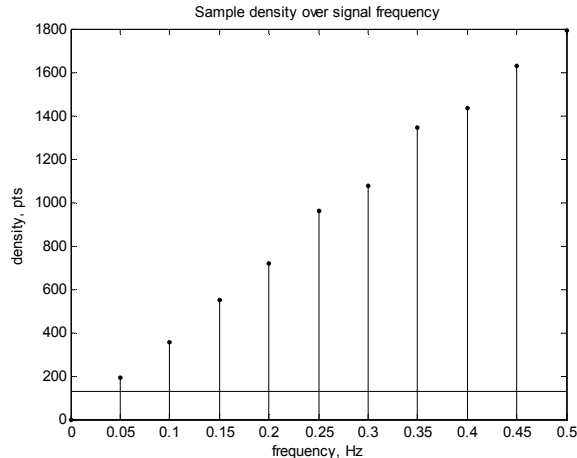
Zīm. 1.2. a) klasiskā ACP līmeņu kļūdas sadalījums troksnim. b) līmeņu kļūdas sadalījums sinusoidālam signālam. c) līmeņu kļūdas sadalījums trokšņveidīgam signālam, izmantojot līmeņu-šķērsojošo ACP. d) līmeņu kļūdas sadalījums sinusoidālam signālam, izmantojot līmeņu-šķērsojošo ACP.

Grafiki, kas attēloti zīmējumā 1.2 c un d ir iegūti ar līmeņu-šķērsojošo diskretizāciju. Kā var redzēt zīmējumā 1.2.c, ka pie 0 vērtības, kur ir vismazākā kļūdas varbūtības vērtības ir lielākas un tuvojoties pusei no attāluma starp līmeņiem kļūdas varbūtība samazinās. Sinusoidāla signāla gadījumā (1.2.d zīm.) kļūda sadalās savādāk. Pie 0 varbūtība ir mazāka un uz abām pusēm tā vispirms palielinās un tikai tad samazinās.

Dotā projekta otrajā etapā ir apskatītas tikai līmeņu-šķērsojošā ACP īpašības, tad turpmākais darbs ir paredzēts šo īpašību izmantošanai un detalizētākai analīzei.

1.2. Nolašu blīvums

Nolašu blīvums spēlē lielu lomu līmeņu-šķērsojošā ACP, jo tas ir atkarīgs no signāla īpašībām (Skat. zīm. 1.3). Klasiskajā ACP gadījumā nolašu skaits nemainās atkarībā no diskretizētā signāla frekvences, bet pielietojot līmeņa-šķērsojošo diskretizāciju nolašu skaits mainās atkarībā no signāla frekvences. Ja signāls mainās straujāk, tad nolases atrodas viena otrai tuvāk un otrādi, ja signāla frekvence ir zema, tad arī nolases atrodas viena no otras lielākā attālumā.



Zīm. 1.3. Nolašu blīvums atkarībā no signāla frekvences klasiskajam ACP un līmeņu - šķērsojošajam ACP.

Tā kā signāla forma ir saistīta ar signāla spektrālo saturu, tad izmantojot šo īpašību ir iespējams noteikt signāla momentāno maksimālo frekvenci.

Ja ieejā ir sinusoidāls signāls

$$x(t) = A \cdot \sin(2\pi f_0 t + \varphi), \quad (1.3)$$

kur A – signāla amplitūda, f_0 – signāla frekvence, t – laiks, φ – sākuma fāze. Nolašu blīvumu var noteikt pēc formulas

$$\sigma = 2R_{\Delta} f_0, \quad (1.4)$$

kur R_{Δ} - dažādo šķērsoto līmeņu skaits.

Kā var redzēt no formulas, tad nolašu blīvums harmoniskam signālam ir nemainīgs un ir atkarīgs tikai no signāla augstākās frekvences un šķērsoto līmeņu skaita.

Noteikt nolašu blīvumu platjoslas signālam nav tik vienkārši kā vienas harmoniskam signālam. Analītiskā ceļā ir izvestas formulas, kas nosaka nolašu blīvumu signālam ar Gausa sadalījumu, kura vidējā vērtība ir vienāda ar nulli un nemainīgu spektrālo blīvumu [12]

$$x(f) = \begin{cases} S & |f| \leq f_0 \\ 0 & \text{citur} \end{cases}. \quad (1.5)$$

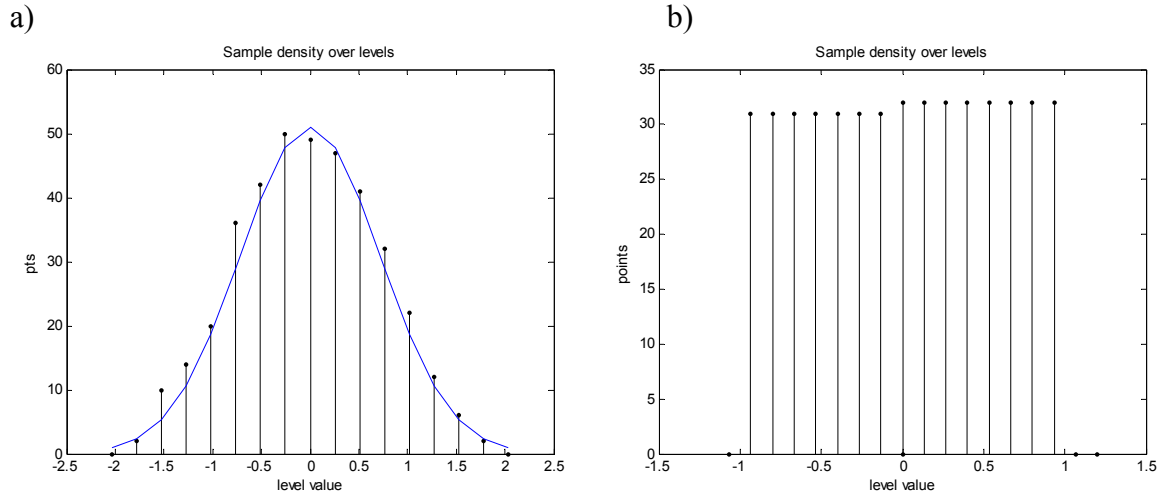
Paredzamais l_0 līmeņa šķērsojumu skaitu laika vienībā ir iespējams noteikt

$$E[\sigma_{l_0}] = \frac{2f_0}{\sqrt{3}} \exp\left(\frac{-l_0^2}{4Sf_0}\right). \quad (1.6)$$

Lai noteikti nolašu blīvumu, ir nepieciešams summēt visus kvantēšanas līmeņus l_k

$$E[\sigma] = \sum_{k=1}^{2^L-1} \sigma_{l_k}. \quad (1.7)$$

Turpmāk tiks apskatītas dažas līmeņu-šķērsojošā ACP īpašības. Kā vienu no pirmajām īpašībām var minēt nolašu sadalījumu pa līmeņiem (Zīm. 1.3).



Zīm. 1.4. Nolašu blīvuma sadalījums pa līmeņiem a) joslā ierobežotam troksnim, b) sinusoidālam signālam.

Zīmējumā 1.4.a grafikā ir attēlots nolašu blīvuma sadalījums pa līmeņiem joslā ierobežotam troksnim. Nolases sadalās pēc Gausa likuma, tas norāda uz to, ka signāls nav periodisks. Paskatoties uz zīmējumu 1.4.b, kurā ir attēlots sinusoidāls signāls, nolašu blīvuma sadalījums ir konstants, kas norāda, ka signāls šķērso visus līmeņus vienādu skaitu reizi. Tātad signāls ir periodisks.

1.3. Diskretizācijas laika intervāli

Kā vēl viens ļoti svarīgs process, kas raksturo diskretizācijas procesu ir laika intervāls starp divām blakus nolasēm

$$\Delta t_n = t_{n+1} - t_n, \quad (1.8)$$

kur Δt_n – laika intervāls starp divām nolasēm, t_{n+1} – nākošās nolasē laika moments, t_n – pašreizējās nolasē laika moments. Vidējais nolašu laika intervāls ir saistīts ar nolašu blīvumu

$$\Delta t_n = \frac{1}{\sigma}. \quad (1.9)$$

Precīzu attālumu starp divām nolasēm Δt_n var noteikt analītiski priekš konkrēta gadījuma. Ņemot par pamatu vienas harmonikas signālu (3) un ja signāls šķērso līmeni l_k laika momentā t_n un līmeni l_{k+1} laika momentā t_{n+1} , tad attālumu Δt_n var aprēķināt

$$\Delta t_n = \frac{1}{2\pi f_0} \left| \arcsin\left(\frac{l_k}{A}\right) - \arcsin\left(\frac{l_{k+1}}{A}\right) \right|. \quad (1.10)$$

Ekstrēmu tuvumā signāls vienu un to pašu līmeni šķērso divas reizes un attālums starp nolasēm ir

$$\Delta t_n = \frac{1}{\pi f_0} \left(\frac{\pi}{2} - \arcsin\left(\frac{|l_{\min|\max}|}{A}\right) \right). \quad (1.11)$$

Ja attālumu starp divām nolasēm Δt_n nevar noteikt analītiski, tad augstāko un zemāko laika intervālu robežas var noteikt balstoties uz signāla parametriem. Minimālo attālumu var noteikt ar

$$T_{\min} \geq \frac{\Delta l_{\min}}{\max(|x'(t)|)}, \quad (1.12)$$

kur Δl_{\min} – minimālais attālums starp diviem kvantēšanas līmeņiem, $x'(t)$ – ir signāla pirmās kārtas atvasinājums.

Gadījumā, kad signāls šķērso vienu un to pašu līmeni divas reizes T_{\min} var sasniegt nulli, jo $\Delta l=0$. Laika intervāls starp divām nolasēm Δt_n ir bezgalīgs, jo līmeņu-šķērsojošā diskretizācija nevar kvantēt signālu, kas atrodas starp diviem līmeņiem. Lai novērstu šo problēmu, attālumam starp kvantēšanas līmeņiem ir jābūt mazākiem nekā signāla amplitūda.

2. Signāla rekonstrukcija no līmeņu šķērsojuma notikuma nolasēm

Veicot signāla līmeņu-šķērsojuma diskretizāciju, vispārīgā gadījumā iegūst laikā nevienmērīgi izvietotas signāla nolases, kuru skaitu ietekmē paša signāla īpašības – jo straujāk tas mainās, jo vairāk nolašu. Signāla rekonstrukcija ir pretējs process diskretizācijai – zinot signāla nolases $s(t_n)$, $n = 0, 1, \dots, N-1$, tiek iegūta tā sākotnējā analogā forma $s(t)$. Problēmas būtība ir izstrādāt metodes un algoritmus, kas šos procesus realizē.

2.1. Eksistējošo metožu izpēte, to trūkumi un priekšrocības

Līdz šim jau publicētās signāla atjaunošanas metodes:

- ✓ *Sinc* funkciju metode,
- ✓ Splainu metode,
- ✓ Iteratīvas signāla atjaunošanas metodes:
 - Voronova metode,
 - Lineāro funkciju metode,
 - Marvasti metode,
 - Adaptīvo koeficientu metode.

***Sinc* funkciju** metode ir veidota signāla atjaunošanai no tā vienmērīgi izvietotām nolasēm. Saskaņā ar Šenona-Koteļņikova teorēmu visu informāciju par signālu ar ierobežotu spektru $f \in [0, f_{\max}]$ satur tā nolases, kas fiksētas vienmērīgi izvietotos laika momentos ar soli $T \leq \frac{1}{2f_{\max}}$, kur f_{\max} ir maksimālā frekvence signāla spektrā[14].

Ierobežota skaita N nolasēm, atjaunotā signāla izteiksme ir:

$$s(t) = \sum_{n=0}^{N-1} s(nT) \cdot \text{sinc}\left(\frac{\omega_0}{2}(t - nT)\right), \quad (2.1)$$

kur $s(nT)$ - signāla nolases, bet $\omega_0 = 4\pi f_{\max}$. Izteiksmi (2.1) var vispārināt un pielietot nevienmērīgas diskretizācijas gadījumam. Zinot signāla $s(t)$ nevienmērīgi izvietotās nolases $s(t_n)$, $n = 0, 1, \dots, N-1$, atjaunotā signāla izteiksmi raksta:

$$\hat{s}(t) = \sum_{n=0}^{N-1} s(t_n) \cdot \text{sinc}\left(\frac{\omega_0}{2}(t - t_n)\right) \quad (2.2)$$

Ja $s(t_n)$ vietā izteiksmē (2.2) liek koeficientus D_n , tad iegūst (2.1) otru vispārinājumu, kur D_n tādā gadījumā atrod, atrisinot lineāru N vienādojumu sistēmu, ko iegūst, ja izvirza nosacījumu $s(t_n) = \hat{s}(t_n)$.

Sinc funkciju metodes trūkums ir tas, ka tā ir veidota vienmērīgi diskretizēta signāla atjaunošanai. Pēc līmeņu-šķērsojuma principa diskretizēta signāla nolases bieži vien būs nevienmērīgi izvietotas laikā, tāpēc atjaunotais signāls būs nekorekts.

Splainu metode līdzīgi *sinc* funkciju metodei ir veidota signāla atjaunošanai no tā vienmērīgi izvietotām nolasēm. Atjaunotā signāla izteiksme ierobežota skaita K nolasēm:

$$\hat{s}(t) = \sum_{k=0}^{K-1} s(kT) \cdot \eta_1^n(t - kT), \quad (2.3)$$

kur $\eta_1^n(t)$ ir n -tās kārtas pamat-splains[15]. Pieaugot pamat-splaina kārtai n , tas sāk vairāk līdzināties *sinc* funkcijai, un to galvenā priekšrocība ir īpašība strauji norimt. Izteiksmi (2.3) var vispārināt un pielietot nevienmērīgas diskretizācijas gadījumam, līdzīgi kā (2.2). Atjaunotā signāla izteiksmi var rakstīt:

$$\hat{s}(t) = \sum_{k=0}^{K-1} s(t_k) \cdot \eta_1^n(t - t_k), \quad (2.4)$$

kur $s(t_k)$ vietā varētu likt arī koeficientus D_k , kurus atrod, izvirzot nosacījumu $\hat{s}(t_k) = s(t_k)$.

Splainu metodes priekšrocība ir splainu īpašība strauji norimt, tāpēc atjaunotā signāla momentāno vērtību ietekmē mazāks skaits signāla nolašu. Savukārt trūkums ir tāds, ka arī šī metode ir veidota vienmērīgi diskretizēta signāla atjaunošanai.

Iteratīvas signāla atjaunošanas metodes. Jebkuru frekvenču joslā ierobežotu signālu iespējams atjaunot no tā nevienmērīgi izvietotām nolasēm, ja maksimālais attālums T_{\max} starp divām nolasēm nepārsniedz Naikvista soli $T = \frac{1}{2f_{\max}}$ [16]. Signāla atjaunošana norit pēc iteratīva algoritma. Zinot nepārtraukta signāla $s(t)$ nolases $s_n = s(t_n)$, $n = 0, 2, \dots, N - 1$, tiek konstruēts kontinuāls aproksimācijas signāls

$$s_a(t) = A[s_n], \quad (2.5)$$

kur A apzīmē aproksimācijas operatoru. Tālāk $s_a(t)$ tiek filtrēts, vājinot liekās spektra komponentes, kas neiekļaujas sākotnējā signāla $s(t)$ spektrā. Filtra izejas signāls (atjaunotais signāls pēc nulltās iterācijas)

$$\hat{s}_0(t) = \Pi[s_a(t)] = \Pi[A[s_n]], \quad (2.6)$$

kur Π apzīmē projekcijas operatoru, no sākotnējā atšķirsies par $\varepsilon_0(t) = s(t) - \hat{s}_0(t)$, turklāt eksistē tāds pozitīvs koeficients $q < 1$, kuram

$$\|s(t) - \hat{s}_0(t)\| < q \cdot \|s(t)\| \quad (2.7)$$

Ja tālāk aproksimācijas un projekcijas operatorus pielieto kļūdas signāla nolasēm, tad atjaunotais signāls pēc m -tās iterācijas:

$$\hat{s}_m(t) = \hat{s}_{m-1}(t) + \Pi[A[s_n - \hat{s}_{m-1}]], \quad (2.8)$$

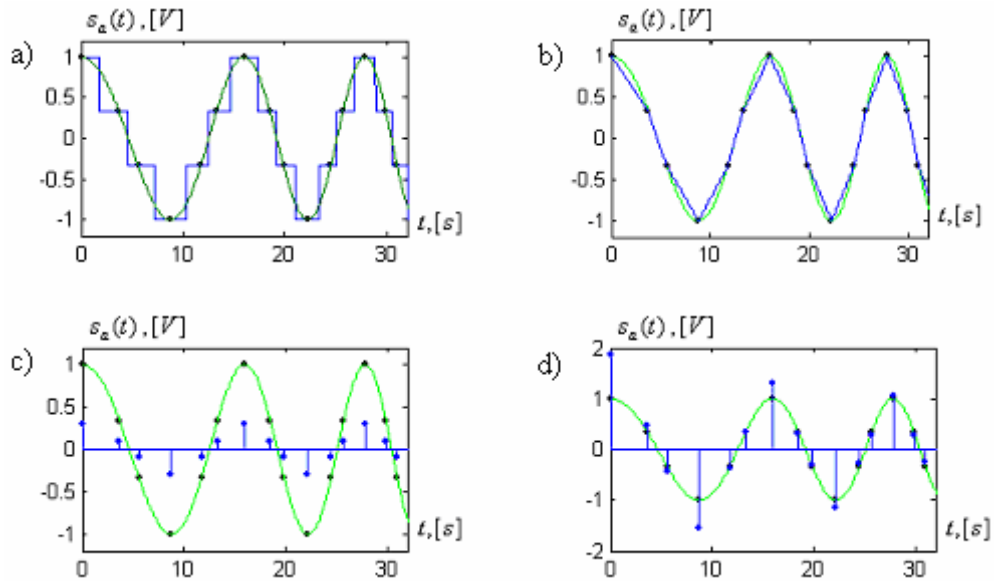
bet kļūdas signāls

$$\|s(t) - \hat{s}_m(t)\| < q^{m+1} \cdot \|s(t)\|, \quad (2.9)$$

pieaugot m , tiecas uz nulli, tāpēc

$$\lim_{m \rightarrow \infty} \hat{s}_m(t) = s(t) \quad (2.10)$$

Atkarībā no tā, kā tiek konstruēts aproksimācijas signāls (kāds ir aproksimācijas operators A), izšķir dažādas signāla atjaunošanas metodes (2.1. zīm.), kas savā starpā atšķiras ar konverģences ātrumu, aprēķinu sarežģītību un citiem parametriem[17].



Zīm. 2.1. Signāla aproksimācija: a) Voronova, b) Lineāro funkciju, c) Marvasti, d) Adaptīvo koeficientu metode

Voronova metodes gadījumā signālu aproksimē ar taisnstūrveida funkcijām. Lineāro funkciju metodes gadījumā signāls tiek aproksimēts ar taisnēm. Marvasti metodei aproksimācijas signālu veido signāla nolases, kas pareizinātas ar globālu relaksācijas koeficientu, kurš nosaka signāla rekonstrukcijas algoritma konverģenci vai diverģenci. Adaptīvo koeficientu metodei aproksimācijas signālu veido tās pašas signāla nolases, tikai katra no tām ir pareizināta ar tai atbilstošu svara koeficientu, kuru nosaka attālums starp blakus nolāsēm.

Iteratīvo metožu priekšrocība ir spēja veikt nevienmērīgi diskretizēta signāla precīzu atjaunošanu. Kā trūkums ir jāmin ierobežojums attiecībā uz maksimālo attālumu starp signāla nolāsēm, kas nedrīkst pārsniegt Naikvista soli. Pēc līmeņu šķērsojuma principa diskretizētam signālam attālums starp blakus nolāsēm var pārsniegt Naikvista soli vairākas reizes.

2.2. Sinc funkcijas parametru adaptācija atbilstoši lokālam nolašu blīvumam

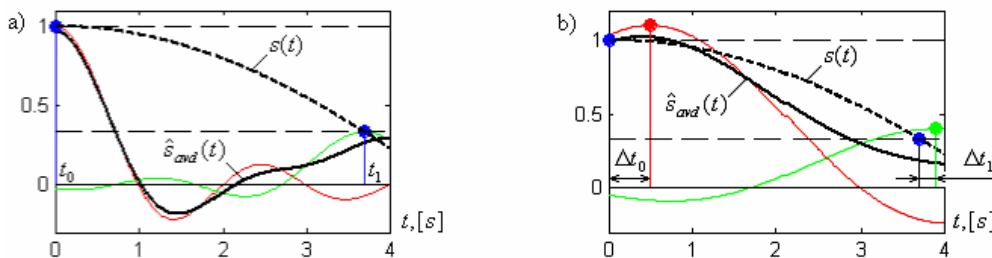
Sinc funkciju metodes gadījumā atjaunotā signāla izteiksme ierobežota skaita N nolāsēm bija:

$$\hat{s}(t) = \sum_{n=0}^{N-1} s(t_n) \cdot \text{sinc}\left(\frac{\omega_0}{2}(t - t_n)\right), \quad (2.11)$$

kur lielumu $\omega_0 = 2\omega_{\max}$ noteica signāla spektra maksimālā frekvence ω_{\max} . Ja tagad katrai signāla nolasei piemeklē savu sinc funkciju ar noteiktu svara koeficientu, galvenās vāles platumu un laika nobīdi, tad atjaunotā signāla izteiksme būs:

$$\hat{s}(t) = \sum_{n=0}^{N-1} C_n \cdot \text{sinc}\left(\frac{\omega_n}{2}(t - t'_n)\right), \quad (2.12)$$

kur parametru C_n , ω_n un t'_n vērtības jāatrod tā, lai atjaunotais signāls no sākotnējā atšķirtos pēc iespējas mazāk. Parastās un uzlabotās *sinc* funkciju metodes atšķirību var novērot 2.2. zīmējumā, kurā ilustrēta signāla rekonstrukcija no tā divām līmeņu-šķērsojuma nolasēm.



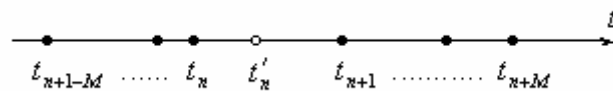
Zīm.2.2. Signāla rekonstrukcija ar a) parasto un b) uzlaboto *sinc* funkciju metodi

Kā redzams, tad uzlabotās *sinc* funkciju metodes gadījumā atjaunotais signāls $\hat{s}_{avd}(t)$ no sākotnējā $s(t)$ atšķiras ar mazāku kļūdu.

Pēc līmeņu šķērsojuma principa diskretizēta signāla nolašu skaitu ietekmē paša signāla īpašības – jo straujāk tas mainās (augstāka frekvence), jo vairāk nolašu, tāpēc loģiski būtu izteiksmē (2.12) *sinc* funkciju parametru ω_n saistīt ar **lokālu signāla nolašu blīvumu**, kuru, vadoties pēc 2.3. zīmējuma, definēsim:

$$\rho_M(t'_n) \Big|_{t'_n \in (t_n, t_{n+1})} = \frac{2M}{t_{n+M} - t_{n+1-M}} \quad \text{vai} \quad \rho_M(t'_n) \Big|_{t'_n = t_n} = \frac{2M}{t_{n+M} - t_{n-M}} \quad (2.13)$$

kur $2M$ – vērā ņemamo nolašu skaits, kas izvietotas uz abām pusēm no t'_n .



Zīm. 2.3. Signāla nolašu laika momenti

Zinot $\rho_M(t'_n)$, var aprēķināt *sinc* funkciju parametru ω_n :

$$\omega_n = 2\pi \cdot \rho_M(t'_n) \quad (2.14)$$

Atkarībā no M iegūst dažādas ω_n vērtības, bet, lai saglabātu lokālā nolašu blīvuma ideju, šim skaitlim nevajadzētu būt pārāk lielam.

Simetrisku *sinc* funkciju amplitūdu aprēķins. Vienkāršojot uzdevumu, pieņem $t'_n = t_n$ un izvirza nosacījumu $\hat{s}(t_n) = s(t_n)$, no kura izriet:

$$\mathbf{C} = \mathbf{\Omega}^{-1} \cdot \mathbf{S}, \quad (2.15)$$

kur matrica \mathbf{C} satur nezināmos koeficientus C_n , matrica \mathbf{S} satur signāla nolašu vērtības $s(t_n)$, bet $\mathbf{\Omega}$ matricas i -tās rindas un m -tās kolonnas elements būs:

$$\Omega_{i,m} = \text{sinc} \left(\frac{\omega_{m-1}}{2} (t_{i-1} - t_{m-1}) \right) \quad (2.16)$$

Nesimetrisku *sinc* funkciju amplitūdu aprēķins. Šajā gadījumā matricas $\mathbf{\Omega}$ i -tās rindas un m -tās kolonnas elements būs divu *sinc* funkciju summa:

$$\Omega_{i,m} = \alpha_{(m-1)l} \cdot \text{sinc}\left(\frac{\omega_{(m-1)l}}{2}(t_{i-1} - t_{m-1})\right) + \alpha_{(m-1)r} \cdot \text{sinc}\left(\frac{\omega_{(m-1)r}}{2}(t_{i-1} - t_{m-1})\right), \quad (2.17)$$

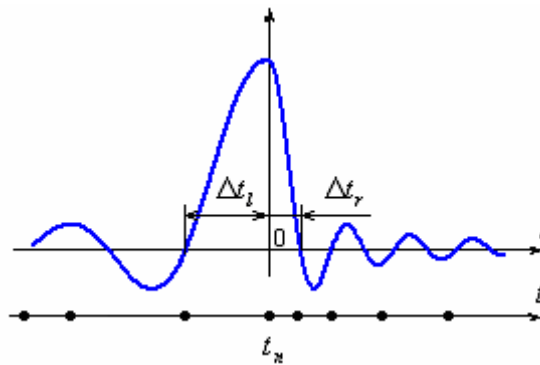
kur

$$\alpha_{nl} = \begin{cases} 1, & \text{ja } t \leq t_n \\ 0, & \text{ja } t > t_n \end{cases}, \quad \alpha_{nr} = \begin{cases} 0, & \text{ja } t \leq t_n \\ 1, & \text{ja } t > t_n \end{cases}, \quad (2.18)$$

bet ω_{nl} un ω_{nr} atrod saskaņā ar (2.14), izmantojot lokālos nolašu blīvumus:

$$\rho_{Ml}(t'_n)|_{t'_n=t_n} = \frac{M}{t_n - t_{n-M}}, \quad \rho_{Mr}(t'_n)|_{t'_n=t_n} = \frac{M}{t_{n+M} - t_n} \quad (2.19)$$

Šāds funkciju summas piemērs ilustrēts 2.4. zīmējumā.

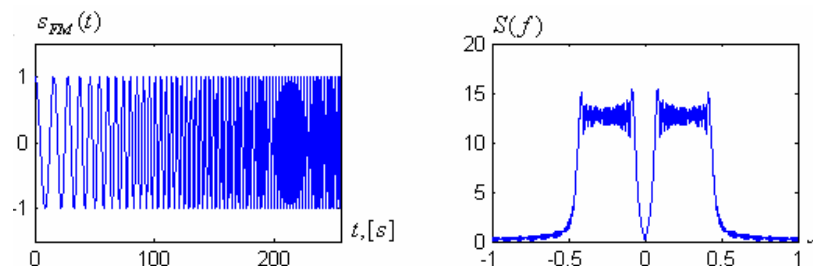


Zīm.2.4. Divu *sinc* funkciju summas piemērs

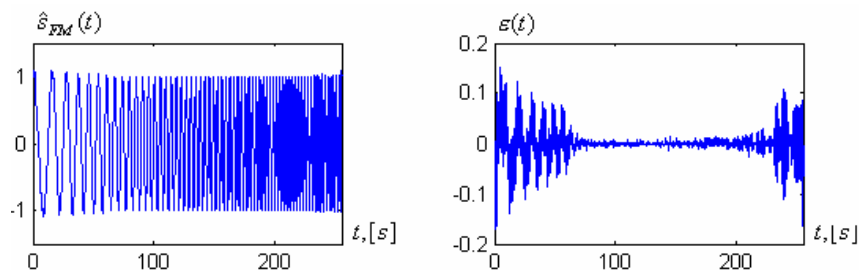
Pamatojums šādai nesimetrisku *sinc* funkciju izvēlei ir tāds, ka nolašu blīvums pa labi un pa kreisi no t_n var būt visai atšķirīgs.

2.3. Metožu darbības modelēšana un salīdzinošā analīze

Kad ir apgūtas metožu galvenās pamatidejas, var veikt signāla atjaunošanas simulāciju *MATLAB* vidē. Kā testa signāls ir izvēlēts frekvences modulēts (FM) signāls, kura frekvence lineāri pieaug no 0.05 Hz līdz 0.45 Hz. Pamatojums FM signālam ir tāds, ka tā frekvence laikā mainās, tāpēc mainās arī pēc līmeņu šķērsojuma principa iegūto nolašu blīvums. Testa signāls attēlots 2.5. zīmējumā, bet atjaunotais signāls, kas iegūts ar simetrisku *sinc* funkciju metodi, ir parādīts 2.6. zīmējumā.



Zīm.2.5. Testa frekvences modulēts signāls $s_{FM}(t)$ un tam atbilstošais spektrālais blīvums $S(f)$



Zīm.2.6. Atjaunotais signāls $\hat{s}_{FM}(t)$ un kļūda $\varepsilon(t)$ simetrisku *sinc* funkciju gadījumā

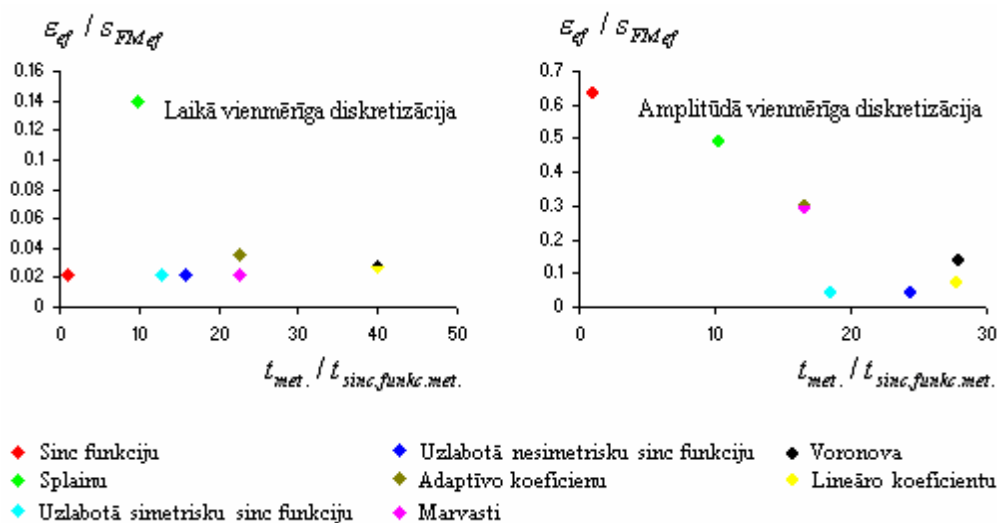
Veicot frekvences modulēta signāla atjaunošanas simulāciju *MATLAB* vidē ar katru no apskatītajām metodēm, iegūtie rezultāti apkopoti tabulā, bet to grafisks laika-kvalitātes ilustrējums dots 2.7. zīmējumā.

Tabula 2.1. Metožu rezultātu salīdzinājums

Metode	$ \varepsilon(t) _{\max}, [V]$		$\varepsilon_{ef} / s_{FM_{ef}}$		$t_{met.} / t_{sinc.funkc.met.}$	
	L.v.d.	A.v.d.	L.v.d.	A.v.d.	L.v.d.	A.v.d.
<i>Sinc</i> funkciju	0.24	1.25	0.022	0.64	1	1
Uzlabotā simetrisku <i>sinc</i> funkciju	0.24	0.17	0.022	0.044	12.67	18.52
Uzlabotā nesimetrisku <i>sinc</i> funkciju	0.24	0.19	0.022	0.042	15.84	24.41
Splainu	0.58	1.04	0.14	0.49	9.8	10.26
Voronova	0.23	1.23	0.027	0.14	40	27.90
Lineāro koeficientu	0.23	0.74	0.027	0.074	39.96	27.81
Marvasti	0.24	1.08	0.036	0.30	22.51	16.71
Adaptīvo koeficientu	0.24	1.10	0.022	0.30	22.62	16.71

L.v.d. – laikā vienmērīga diskretizācija
A.v.d. – amplitūdā vienmērīga diskretizācija

Metožu salīdzināšanai tiek izmantotas kļūdas signāla $\varepsilon(t) = s_{FM}(t) - \hat{s}_{FM}(t)$ maksimālās vērtības $|\varepsilon(t)|_{\max}$ un efektīvo vērtību attiecības $\varepsilon_{ef} / s_{FM_{ef}}$, kur $s_{FM_{ef}} = 0.71V$ ir paša FM signāla vidējā kvadrātiskā vērtība. Salīdzināts tiek arī laiks, kas nepieciešams atjaunotā signāla aprēķinam ar izvēlēto metodi. Lai izvairītos no datora veikspējas ietekmes uz laika novērtējumu, tas tiek dots attiecībā pret to laiku, kas nepieciešams signāla aprēķinam ar *sinc* funkciju metodi.



Zīm. 2.7. Metožu laika-kvalitātes novērtējums

Signāla atjaunošana ar visām metodēm tika realizēta gan laikā vienmērīgi, gan amplitūdā vienmērīgi diskretizētam FM signālam. Visu četru iteratīvo metožu rezultāts ir iegūts pēc devītā cikla. Aplūkojot 2.7. zīmējumu, izriet šādi secinājumi:

- 1) Visas apskatītās metodes ar vienlīdz labu rezultātu ļauj atjaunot vienmērīgi diskretizētu signālu, būtiski atšķiras tikai algoritmu izpildes laiki. Ar spline metodi iegūtais kļūdas signāls ir lielāks, kas skaidrojams ar pārāk zemas kārtas pamat-splaina izmantošanu signāla rekonstrukcijā.
- 2) Vismazāko signāla atjaunošanas kļūdu dod tieši uzlabotā *sinc* funkciju metode, kas to padara visai interesantu no turpmāko nākotnes pētījumu viedokļa.
- 3) Labs rezultāts ir sasniegts ar lineāro funkciju metodi, kas skaidrojams ar samērā veiksmīgu signāla aproksimāciju no tā līmeņu-šķērsojuma nolasēm nulltajā iterācijā. Tāpēc būtu interesanti nākotnē apskatīt arī citus signāla aproksimācijas paņēmienu, piemēram, izmantojot otrās kārtas polinomus. Vēl pastāv iespēja dažādos ciklos izmantot atšķirīgus signāla aproksimācijas paņēmienu.
- 4) Parasto *sinc* funkciju un B-spline metodes tomēr nav piemērojamas amplitūdā vienmērīgi diskretizēta signāla atjaunošanai, jo kļūdas signāls ir samērojams ar pašu signālu.

3. Nestacionāru signālu apstrādes beztaktētāja sistēmas modelēšana

Klasiskais paņēmiens, kā pārveidot nepārtraukto signālu ciparu formā ir izdarīt nolases noteiktos laika momentos un pēc tam nokodēt šīs nolases amplitūdas vērtību ar noteikto bitu skaitu. Laika momenti ir novietoti pa laika ass vienmērīgi, respektīvi, starp visām nolāsēm ir viens un tāds pats attālums, kuru sauc par diskretizācijas soli τ . Lai realizētu signāla vienmērīgo diskretizāciju ir vajadzīgs pulkstenis, jeb taktētāja sistēma. Taktētāja sistēmas klātbūtne ierīcē pirmkārt prasa samērā lielu enerģijas patēriņu, kas nav īpaši labi, ja ierīce darbojas autonomā režīmā ar baterijas palīdzību, otrkārt, ir vajadzīga stabila un precīza pulksteņa darbība. Cita pieeja ir izdarīt nepārtraukta signālu diskretizāciju nevienmērīgos laika momentos. Viens no paņēmieniem, kā to var realizēt ir līmeņu-šķērsojuma princips. Šis princips tiek realizēts beztaktētāja sistēmās, jo nolase tiek ņemta, laika momentā kad signāls sasniedz kādu iepriekš izvēlēto vērtību, bet nevis takts impulsa pienākšanas brīdī. Apskatīsim šīs darbības principu.

3.1. Nestacionāru signālu analogs-ciparu pārveidošana izmantojot līmeņu-šķērsojuma principu

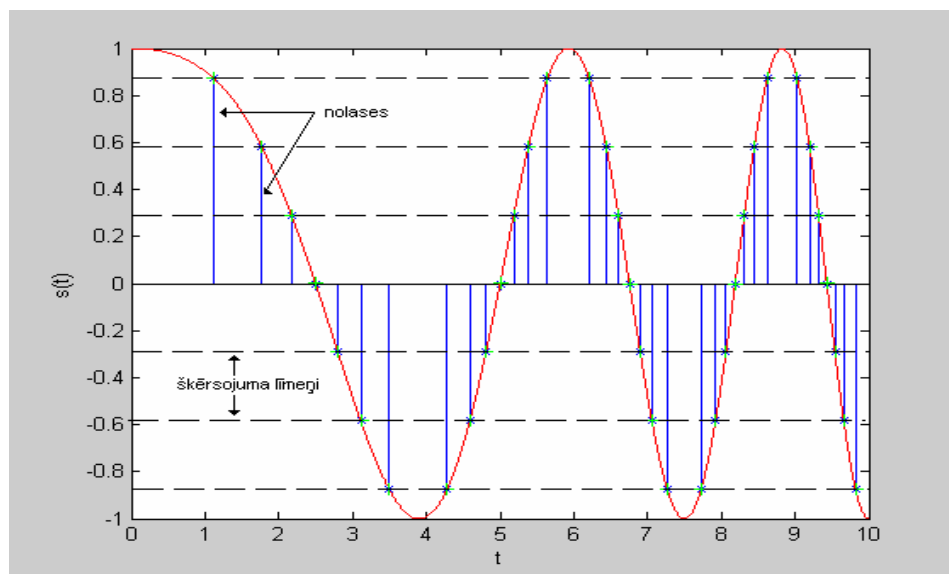
Uz frekvences modulēta signāla piemēra apskatīsim, kā notiek pāreja no analoga nestacionāra signāla uz signālu ciparu formā, kad nolases tika ņemtas ar līmeņu-šķērsojuma principu.

Mūsu izvēlēta signāla atkarība no laika var būt izteikta ar formulu

$$s(t) = \cos(2\pi(0.05 + t)t) \quad (3.1)$$

Sākuma fāze signālam $s(t)$ ir nulle un amplitūda ir 1.

Zīm. 3.1. ir dots grafiks, kur ir parādīts signāls un izdarītas nolases. Kā redzams, nolases ir novietotas nevienmērīgi uz laika ass.



Zīm. 3.1. Frekvenču modulēta signāla nevienmērīga diskretizācija ar līmeņu-šķērsojuma principu.

3.2. Laika-frekvenču reprezentāciju modelēšana

Pēc diskretizācijas mums ir nolašu plūsma, kura atbilst nestacionāram signālam, respektīvi, procesam ar laikā mainīgu spektrālo blīvumu. Lai iegūtu informāciju par diskrēto signālu mums ir jāpielieto kāda no speciālām signāla transformācijām. Klasiskā Furjē transformācija (nepārtrauktā un diskrēta) ir viens no pamatinstrumentiem signāla analīzē, bet tā nav pietiekoši efektīva nestacionāru signālu apstrādē, jo ir balstīta uz stacionaritātes koncepta. Tāpēc mēs šeit pievērsīsim uzmanību signāla laika-frekvenču reprezentācijai, kura raksturojas ar punktu kopumu laika-frekvenču plaknē. Apskatīsim trīs klasiskos laika-frekvenču reprezentācijas veidus: 1. Īssintervāla Furjē transformācija, 2. vilnīšanalīze, 3. Vignera sadalījumu.

3.2.1. Īssintervāla Furjē transformācijas pieeja

Vēsturiski pirmais piedāvātais veids kā analizēt nestacionārus signālus ir Īssintervāla Furjē transformācija (*STFT*). Tā ir bāzēta uz Furjē transformācijas. Pamatideja ir ievest laika logu, kurš tiek bīdīts gar aplūkoto signālu. Matemātiskā izteiksme šai transformācijai ir sekojoša [24, 25]

$$STFT(t, f) = \int_{-\infty}^{\infty} x(\tau) w^*(t - \tau) \exp(-j2\pi f \tau) d\tau \quad (3.2)$$

kur, $w^*(t - \tau)$ ir laika logs un zvaigzne apzīmē kompleksi saistīto lielumu. Patvaļīgas diskretizācijas gadījumā, kad nolašu skaits ir galīgs $x_n = x[t_n]$, kur $n = \overline{1, N}$ (N ir nolases skaits laikā intervālā θ), *STFR* var būt izteikta sekojoši [26]

$$STFT[k, m] = \left| \sum_{n=1}^N x_n w^* \left(\frac{k}{2\Omega} - t_n \right) \exp \left(\frac{-j2\pi t_n m}{\theta} \right) \right| \quad (3.3)$$

kur, Ω ir signāla spektra platums un m nosaka vienmērīgu soli frekvenču apgabalā. Pieņemsim, ka mūsu uzdevums ir pārveidot ar STFT nevienmērīgi diskretizētu frekvenču modulētu kosinuss funkciju

$$s(t) = \cos \left(2\pi \left(0.05 + \frac{0.2}{256} t \right) t \right) \quad (3.4)$$

Par loga funkciju izvēlamies “*Hamminga*” logu, kurš ir definēts sekojoši [25]

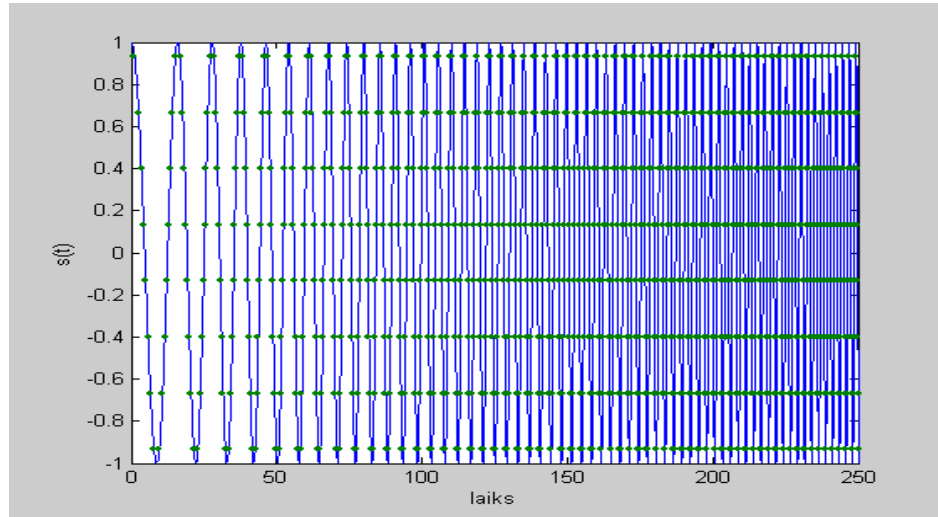
$$w(n) = \begin{cases} 0.54 - 0.46 \cos \left(2\pi \frac{n}{T-1} \right), & 0 \leq n < T \\ 0, & \text{citur} \end{cases} \quad (3.5)$$

kur T ir loga garums. Veicot attiecīgus aprēķinus ir iegūti sekojoši rezultāti: Zīm. 3.2. ir parādīts signāls un nolases; Zīm. 3.3. ir dots nevienmērīgi diskrēta signāla spektrs un 3.4 attēlā ir *STFT* nevienmērīgi diskrētam signālam ar līmeņa-šķērsojuma principu palīdzību. Līmeņa-šķērsojuma analogs-ciparu pārveidojumam ir izvēlēti 8 līmeņi. Lai novērstu artefaktus, kuri ir saistīti ar to, ka $\exp(-j2\pi t_n m / \theta)$ pie nevienmērīgi izvietotiem t_n

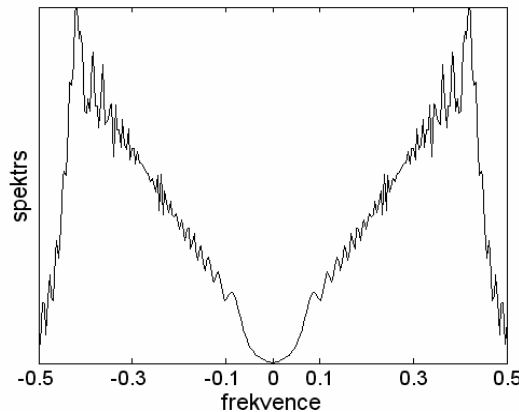
momentiem nedod ortogonālu transformāciju, ir izmantota neortogonāla transformācija [26]. Tas nozīmē to, ka aprēķinos matrica $E_{nm} = \exp(-j2\pi t_n m / \theta)$ ($n = \overline{1, N}; m = \overline{1, M}$) ir aizvietota ar matricu [26]

$$\mathbf{G} = (\mathbf{E}\mathbf{E}^T)^{-1}\mathbf{E} \quad (3.6)$$

kur \mathbf{E}^T ir transponēta matrica un $()^{-1}$ apzīmē inverso matricu.

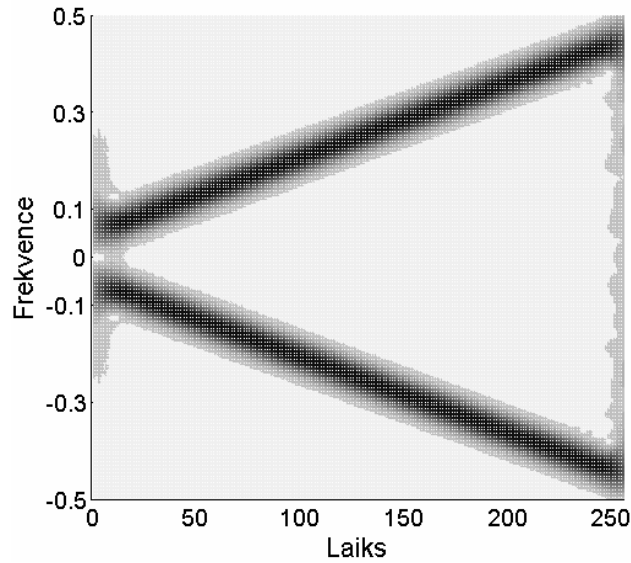


Zīm. 3.2. FM modulēts reāls signāls. Ar punktiem ir atzīmētas vietas, kur signāls krusto attiecīgo līmeni.



Zīm.3.3. Aplūkots signāla spektrs. Frekvence ir atliekta normētas vienības.

Zīm. 3.2. ir redzams, ka jo lielāk ir signāla frekvence, jo lielāk ir nolases skaits. Zīm.3.3. mums neko nevar pateikt par to, kā ar laiku maiņas spektrālais blīvums aplūkotai funkcijai un jo sarežģītāk ir funkcija, jo grūtāk kaut ko pēc tā spektra spriest par signāla sastāvdaļām. Zīm. 3.4. ir dota laika-frekvenču reprezentācija. Kā redzams, tajā neparādās artefakti, kas ir tipiski nevienmērīgai diskretizācijai, jo ir izmantota neortogonāla transformācija. Laika intervālā sākumā un beigās, spektrs ir neskaidri izplūdis gandrīz pa visu frekvenču skalu, kas izskaidrojams ar gala efektiem, kad analīzei ir nepietiekams nolašu skaits. Tas, ka *STFT* ir gan pie pozitīvām un negatīvām frekvencēm ir saistīts ar to, ka reālajam signālam spektrāla blīvuma funkcija ir simetriska pret frekvences nulles punktu.



Zīm. 3.4. STFT laika-frekvenču reprezentācija.

3.2.2. Vilnīša analīzes pieeja

Pazīstama STFT problēma ir sakarība starp izšķirtspēju pēc laika un frekvencēm. Loga garuma paplašināšana uzlabo frekvenču izšķiršanu, bet tajā pašā laikā samazina laicisko izšķirtspēju. Lai novērstu šāda veida problēmu, var tikt pielietota vilnīšanalīze.

Laikā nepārtraukta signāla vilnīštransformācija ir izteikta ar integrāli [25, 26]

$$WT(t, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(\tau) h^* \left(\frac{t - \tau}{a} \right) d\tau \quad (3.7)$$

kur a ir mērogošanas faktors un $h(t)$ ir analīzes vilnīša funkcija. Mērogošanas faktors ir saistīts ar frekvencēm ar sakarību $a = \frac{f_0}{f}$. Vilnīšiem piemīt tāda īpatnība, ka frekvenču izšķirtspēja samazinās pie lielām frekvencēm un izšķiršana laikā samazinās pie mazām frekvencēm.

Praksē izmanto daudz dažādu $h(t)$ vilnīša funkciju. Mēs savā analīzē pielietosim tā saucamu Morleta vilnīti [25]

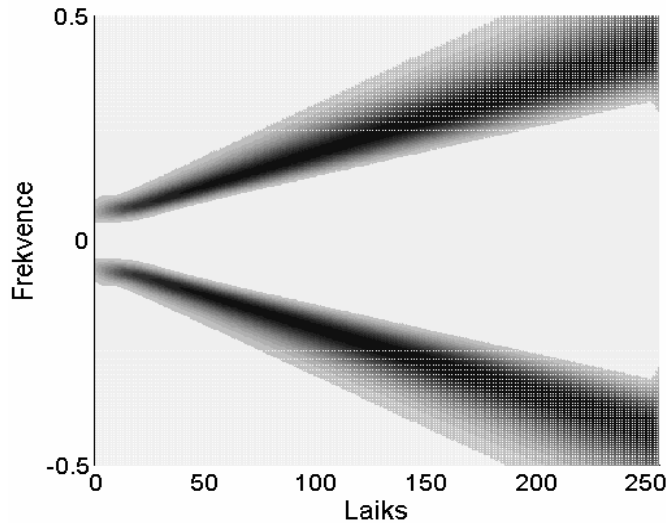
$$h(t) = \exp\{j\omega_0 t\} \exp\left\{-\frac{t^2}{2}\right\} \quad (3.8)$$

kur ω_0 ir dominantā frekvence, ar kuras palīdzību var variēt bāzes selektivitāti.

Diskrētā signāla gadījumā, vilnīšanalīze var būt izteikta ar summas formulu [26]

$$WT[k, m] = \left| \frac{1}{\sqrt{f_0 \theta / m}} \sum_{n=1}^N x[t_n] h^* \left(\frac{k / 2\Omega - t_n}{f_0 \theta / m} \right) \right| \quad (3.9)$$

Šī izteiksme ir spēkā gan vienmērīgi, gan nevienmērīgi novietotam nolasēm.



Zīm. 3.5. Laika-frekvenču reprezentācija lineāri FM kosinuss funkcijai. Signāls bija diskretizēts ar 8 līmeņu palīdzību.

Zīm.3.5. ir parādīts modelēšanas rezultāts, kad punktā 3.2.1. aplūkotais signāls ir diskretizēts ar līmeņa-šķērsojuma principa palīdzību. Ir izmantoti 8 kodēšanas līmeņi. Kā ir redzams, pie lielām frekvencēm samazinās spektrāla izšķirtspēja, savukārt pie zemām frekvencēm samazinās laika izšķirtspēja.

3.2.3. Vignera sadalījuma funkcija

Viena no plaši pielietotām pieejām laika-frekvenču signāla reprezentācijā ir Vignera sadalījuma integrālis, kurš ir sastāvdaļa kopējai laika-frekvenču pārveidošanas pieejai.

Vignera integrālis laika nepārtrauktai funkcijai ir [24, 25, 27]

$$WVD(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \exp(-j2\pi f\tau) d\tau \quad (3.10)$$

Par kodolu, mūsu gadījumā sauc $\exp(-j2\pi f\tau)$. Šis integrālis dod lielu izšķišanas spēju laikā un frekvencēm mono-komponentu signāliem. Tomēr, papildus interferences produkti var parādīties, ja signālam ir vairākas komponentes. Lai cīnīties ar šo efektu, ir izstrādātas citas kodola funkcijas, bet tas tipiski noved pie izšķirtspējas samazināšanas gan laikā gan frekvencēm.

Runājot par nevienmērīgi izvietotām nolasēm, paradās sekojoša problēma. Pieņemsim, ka katra nolase ir Dirāka delta funkcija. Tad plūsmu no vairākām nolasēm varām uzrakstīt kā delta funkciju summu, ierakstot argumentā nobīdes laiku. Tad Vignera integrālim būs forma

$$WVD[t, f) = \int_{-\infty}^{\infty} \left\{ a_0 \delta\left(t + \frac{\tau}{2} - t_0\right) + \dots + a_N \delta\left(t + \frac{\tau}{2} - t_N\right) \right\} \cdot \left\{ a_0^* \delta\left(t - \frac{\tau}{2} - t_0\right) + \dots + a_N^* \delta\left(t - \frac{\tau}{2} - t_N\right) \right\} \cdot \exp(-j2\pi f\tau) d\tau \quad (3.11)$$

kur indeksi $n = \overline{1, N}$ apzīmē attiecīgu laika momentu un nolases vērtību šajā brīdī un $a_{0...N}^*$ ir kompleksie koeficienti, gadījumā, ja nepārtrauktais signāls ir komplekss. Atverot vaļā

iekavas, iegūstam delta funkciju reizinājumu summu. Apskatīsim divu delta funkciju reizinājumu nolasēm laika momentos t_i un t_j .

$$\delta\left(t + \frac{\tau}{2} - t_i\right)\delta\left(t - \frac{\tau}{2} - t_j\right) \neq 0 \quad (3.12)$$

Lai šis funkcijas reizinājums nebūtu vienāds ar nulli, vienmērīgas diskretizācijas gadījumā, integrēšanas solim τ jābūt

$$\tau = t_i - t_j = t_0(i - j) = kt_0 \quad (3.13)$$

kur $k = 0, \pm 1, \pm 2, \dots$ un t_0 ir diskretizācijas solis.

Nevienmērīgas diskretizācijas gadījumā, starpībai starp j -to nolasi un i -to nolasi jābūt tik mazai, kā delta funkcijas eksistēšanas argumenta robežai

$$|t_i - t_j| \leq \varepsilon \quad (3.14)$$

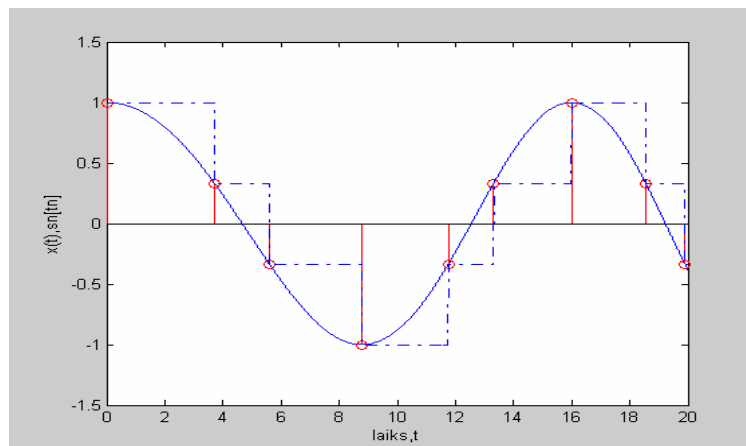
Tas nozīme, ja koeficienti a_n un a_n^* ir atšķirīgi no nulles, tad lai Vignera integrālis būtu korekti atrisināts, nevienmērīgas diskretizācijas gadījumā, nolasēm jāseko vienai aiz otram tik tuvu, ka nevar izšķirt vai signāls ir kontinuāls vai diskrets.

Šī doma noved pie secinājuma, ka diskrēta Vignera sadalījuma funkcija ir pielietojama tikai vienmērīgi diskretizētiem signāliem. Ja signāls ir nevienmērīgi diskrets pastāv iespēja nolases interpolēt ar nepārtrauktu funkciju un pēc tam rēķināt Vignera integrāli interpolētai funkcijai.

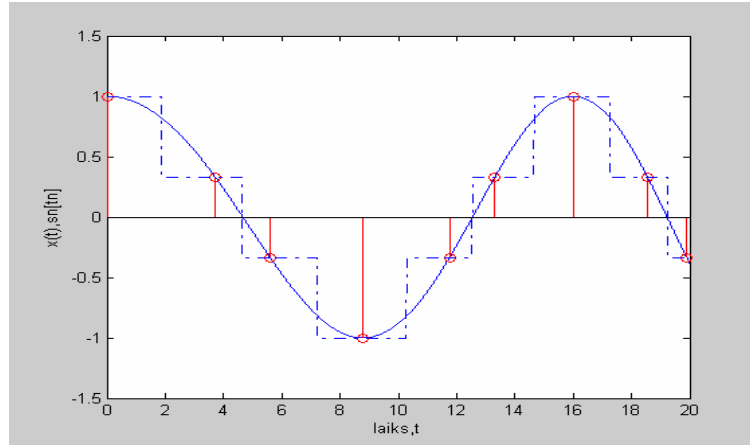
Zīm. 3.6. ir parādīts signāla, kurš ir uzdots ar funkciju (3.4), fragments līdz 20 sekundēm. Nepārtraukta līnija atbilst sākuma funkcijai $s(t)$, bet punktaina – aproksimācijas funkcijai. Šeit ir izmantots princips, kad aproksimācijas līnija sākas no nolases vērtības un uztur to līdz nākošajai vērtībai.

Zīm. 3.7. ir dota uzlabota aproksimācijas pieeja. Tagad, līnijas nesākas no nolases vērtības (nolases laika momenta), bet nolases vērtība atrodas pa vidu interpolācijas līnijas gabalam. Tīri vizuāli šī aproksimācija izskatās tuvāka signālam $s(t)$.

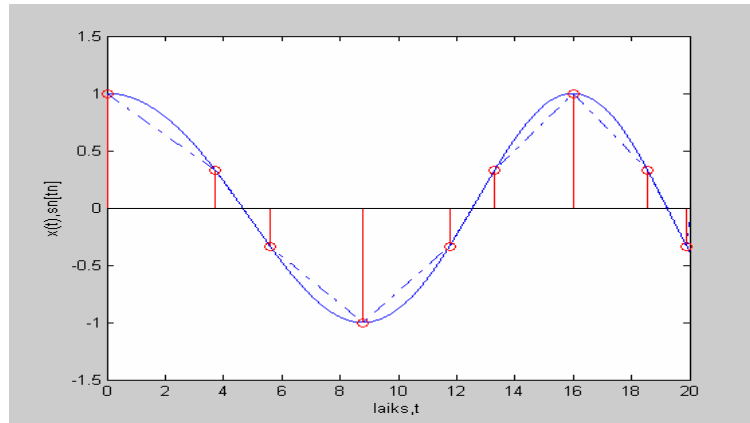
Un pēdējais piedāvātais variants, ir interpolācija ar lineārām funkcijām. Respektīvi, nolases vērtības ir savienotas ir nākošo ar taisnas līnijas palīdzību. Neskatoties uz šīs interpolācijas vienkāršību, tā ir diezgan tuva īstam signālam $s(t)$, ja nolašu blīvums ir liels.



Zīm. 3.6. Aproksimācijas piemērs ar taisnām līnijām, kuri ir paralēli laika ass un sakas no iepriekšējās nolases vērtības un tas laika momenta.



Zīm.3.7. Aproximācijas piemērs ar taisnām līnijām, kuri ir paralēli laika ass un nolases laika momenti ir novietoti apmēram pa vidu aproksimācijas nogriežņiem.



Zīm.3.8. Aproximācija ar lineāru funkciju palīdzību.

Jebkurš aproksimācijas veids pilnībā neatbilst sākuma signālam $s(t)$, vienmēr pastāv kāda aproksimācijas kļūda, bet, lai novērtētu laika-frekvenču analīzes rezultātu tas var būt pietiekošs tuvinājums. Literatūrā nav atrodami risinājumi kā izskatīsies Vignera integrāļa rezultāts nevienmērīgi novietotām nolāsēm, kā arī speciālam gadījumam, kad nolases ir novietotas ar līmeņa-šķērsojuma principa palīdzību. Projektā nākamajos etapos ir plānots izveidot efektīvu algoritmu Vignera sadalījuma funkcijas aprēķināšanai nevienmērīgi izvietotām nolāsēm, pirmkārt apskatot līmeņu šķērsojuma diskretizācijas gadījumu. Tas dos iespēju salīdzināt šīs pieejas rezultātus ar īssintervāla Furjē transformācijas un vilnīšanalīzes laika-frekvenču sadalījumiem.

Jo vairāk metodes būs zināmas laika-frekvenču analīzē nevienmērīgi diskretizētiem signāliem ar līmeņa-šķērsojuma principa izmantošanu, jo adaptīvāk mēs varēsim izveidot ierīci, kura veic signāla apstrādi vai uzrakstīt attiecīgo signālu apstrādes programmatūru.

4. DFT-veida pārveidojumu izstrāde, kas atbilst un ņem vērā ar LŠ-ACP iegūto laikā nevienmērīgo nolašu specifiku.

Klasiskā metode ciparu signālu apstrādē ir Furjē transformāciju pāris. Tiešā diskretā transformācija ļauj no N laikā vienmērīgi izvietotām signāla nolasēm iegūt frekvencēs vienmērīgi izvietotus spektrālos koeficientus:

$$X(m) = \sum_{n=0}^{N-1} x(n) \exp(-j2\pi mn / N) \quad (4.1)$$

Inversā DFT ļauj no vienmērīgi spektrā izvietotiem koeficientiem iegūt signāla nolases vienmērīgi izvietotas laikā:

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) \exp(j2\pi mn / N) \quad (4.2)$$

Praktiskos pielietojumos tipiski eksistē divi ierobežojumi: signālu frekvenču joslas platums nepārsniedz Ω , un signāla novērojuma laiks ir ierobežots ar Θ . Ciparu signāla nolases $x_n = x(t_n)$, $n = 0, N-1$ tiek iegūtas ar diskretizācijas palīdzību un vispārējā gadījumā tās var būt izvietotas nevienmērīgi uz laika ass.

4. 1. Vispārinātā DFT izteiksme

Vispārinātā DFT izteiksme, kas pieļauj nevienmērīgi izvietotu nolašu apstrādi var tikt izteikta kā:

$$X_m = \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi \frac{m}{\Theta} t_n\right). \quad (4.3)$$

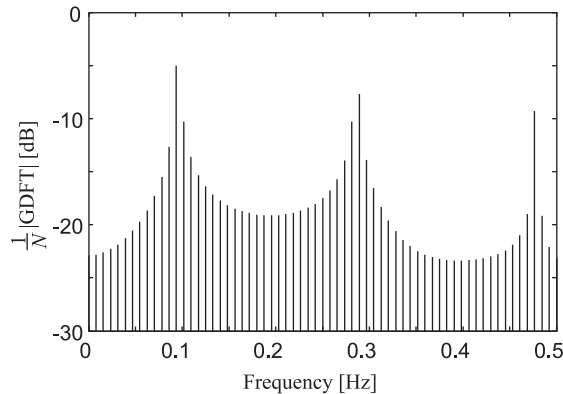
Frekvenču joslas ierobežojums nosaka prasību - $m : \left| \frac{m}{\Theta} \right| \leq \Omega$.

Vienmērīgi izvietotām nolasēm izteiksme (4.3) sakrīt ar formulu (4.1). Ja ir izmantota piemērota apzināti nevienmērīgas diskretizācijas plūsma, tad ar (4.3) var iegūt labus spektrālos novērtējumus gadījumos, kad nolašu blīvums nav mazāks par to ko nosaka Naikvista kritērijs vienmērīgai diskretizācijai. Vēl vairāk, ir iespējams apmierinoši novērtēt signāla spektrālo blīvumu arī tad, ja blīvums ir mazāks par Naikvista limitu un plūsmas īpašības ir izvēlētas tā, ka frekvenču uzklāšanās efekts tiek nospiests. Taču ja izteiksmi (4.3) pielieto nolasēm, kas iegūtas pēc līmeņu-šķērsojuma diskretizācijas principa, tad iegūtie rezultāti ne vienmēr ir adekvāti pat neskatoties uz to, ka vienmēr signāls ir virsdiskretizēts (*oversampled*). Ilustrēsim to ar vienkāršu piemēru – mono harmonisks signāls ar frekvenci f_0 ir diskretizēts pielietojot 3-bitu LŠ-ACP (7 līmeņi). Ar izteiksmi (4.3) iegūtais spektrālais novērtējums ir parādīts 4.1. zīm. Papildus patiesajai spektrālajai komponentei ar frekvenci ~ 0.0955 Hz, parādās arī viltus izsitieni uz augstākām nepāra harmonikām. Tas nav dēļ nepietiekama nolašu blīvuma, bet gan dēļ LŠ-ACP pieejas īpatnībām [5]. Līdz ar to rodas vajadzība pēc uzlabotiem DFT-veida pārveidojumiem, kas spētu novērst šāda veida artefaktu klātbūtni.

Balstoties uz Furjē rindu, signāla laika formu no spektrāliem koeficientiem var atjaunot sekojošā veidā:

$$\hat{x}(t) = \sum_{m=-M}^M X_m \exp(j2\pi t f_m), \quad t \in [0, \Theta], \quad (4.4)$$

kur $f_m = \frac{m}{\Theta}$.

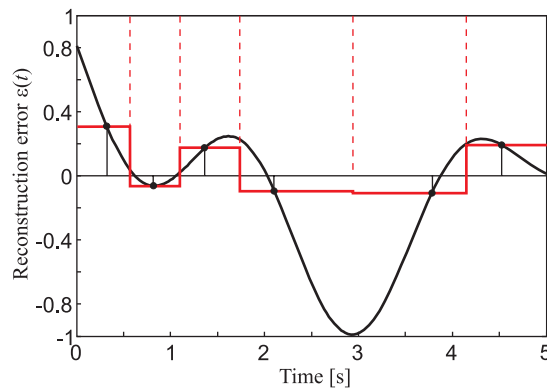


4.1. zīm. Spektrālais novērtējums sinusoidālam signālam ar $f_0 = 12.22/\Theta$ ($\Theta = 128$ s.), kurš diskretizēts izmantojot 7 līmeņu šķērsojumu momentus.

Izmantojot informāciju par signālu $x(t)$, ir iespējams konstruēt atjaunošanas kļūdas funkciju kā $\varepsilon(t) = x(t) - \hat{x}(t)$. Šīs funkcijas sākumapgabals ir parādīts 4.2. zīm., kurā $\varepsilon_n = \varepsilon(t_n)$ vērtības – atjaunošanas kļūda diskretizācijas laika momentos t_n , ir ilustrētas ar treknākiem punktiem. Var redzēt, ka atjaunošanas kļūda starp signāla nolašu laika momentiem mēdz būt ievērojami lielāka nekā ε_n vērtības. Ja diskretizācija ir veidota pēc taktētāja izmantošanas principa, tad principā nav nekādu konkrētu nosacījumu par to, kas ar signāla vērtībām notiek starp tā nolasēm. Līmeņu-šķērsojuma diskretizācijas gadījumā katra nolase ir definēta kā noteikta līmeņa šķērsojums un ir zināms, ka starp tām signāla vērtība atrodas ierobežotā diapazonā. Tas ļauj izteikt ideju, ka signāla atjaunošanas kļūdu vajag minimizēt ne tikai nolašu punktos, bet arī starp tiem ar tik pat lielu svarīgumu. Minimizācijas uzdevums

$$\int_0^{\Theta} \varepsilon^2(t) dt \rightarrow \min \quad (4.5)$$

var tik definēts paturot prātā, ka tā atrisināšanai ir zināmas tikai diskrētas oriģinālā signāla vērtības un ka signāla atjaunošanai tiek izmantota izteiksme (4.4). Minimizācija ir jāveic attiecībā uz spektrāliem koeficientiem $\{X_m\}$.



4.2. zīm. Atjaunošanas kļūdas funkcija (melnā līnija) un kļūdas nolašu ε_n (treknie punkti) interpolācija ar gabaliem konstantu funkciju (sarkanā līnija).

Uzdevuma (4.5) risinājums var tikt balstīts uz divām pieejām:

- 1) signāla nolases $\{x_n\}$ tiek interpolētas intervālā $[0 \Theta]$ un atjaunošanas kļūda tiek izteikta kā $\varepsilon^{(x)}(t) = \tilde{x}(t) - \sum_m X_m \exp(j2\pi f_m t)$, kur $\tilde{x}(t)$ ir interpolētais signāls;
- 2) kļūdas nolases $\varepsilon_n = x_n - \hat{x}_n$ tiek interpolētas intervālā $[0 \Theta]$, ņemot vērā, ka $\hat{x}_n = \sum_m X_m \exp(j2\pi f_m t_n)$.

4.2. DFT uzlabojums ar signāla nolašu interpolācijas palīdzību.

Kā efektīvs signāla interpolācijas paņēmiens ir minams tā nolašu savienošana ar k -tās kārtas polinomiem $p_n^k(t)$, vai frekvenču joslā ierobežota interpolācija, kas tiek veikta ar laikā nobīdītu sinc funkciju palīdzību.

Pārrakstīsim izteiksmi (4.5) ņemot vērā, ka signāla nolases ir interpolētas:

$$\int_0^{\Theta} \left(\tilde{x}(t) - \sum_{m=-M}^M X_m^{(x)} \exp(j2\pi f_m t) \right)^2 dt \rightarrow \min \quad (4.6)$$

Lai atrastu minimumu, visus parcīālos atvasinājumus pēc X_m ir nepieciešams pielīdzināt nullei. Tādejādi iegūstam $2M + 1$ lineārus vienādojumus

$$2 \int_0^{\Theta} \left(\tilde{x}(t) - \sum_{k=-M}^M X_k^{(x)} \exp(j2\pi f_k t) \right) \cdot \exp(j2\pi f_m t) dt = 0 \quad (4.7)$$

kur $m = -M, M$. Ņemot vērā, ka $f_m = \frac{m}{\Theta}$, $\{\exp(j2\pi f_m t)\}$ ir ortogonālu funkciju kopa intervālā $[0 \Theta]$. Līdz ar to varam rakstīt, ka:

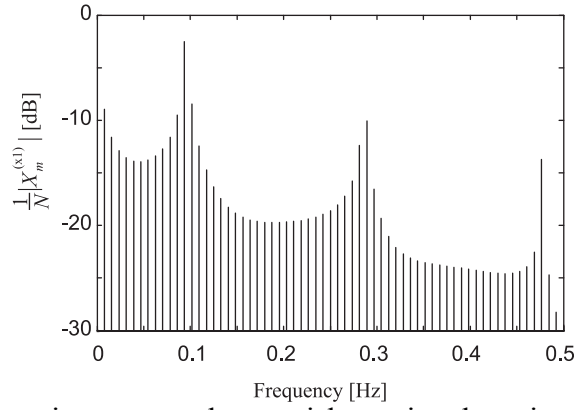
$$X_m^{(x)} = \frac{1}{\Theta} \int_0^{\Theta} \tilde{x}(t) \exp(j2\pi f_m t) dt \quad (4.8)$$

Izteiksme (4.8) ir līdzīga ar Furjē rindas koeficientu aprēķināšanas formulu signālam $\tilde{x}(t)$.

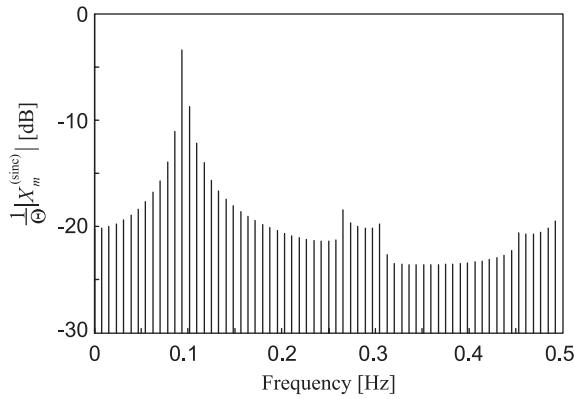
Ja nolases tiek interpolētas ar nulltās kārtas polinomiem (gabaliem konstanta funkcija, līdzīga tai kas atainota 4.2. zīm. ar sarkanu līniju), iegūstam

$$\begin{aligned} X_m^{(x0)} &= \sum_{n=0}^{N-1} x_n \int_{\frac{t_n+t_{n-1}}{2}}^{\frac{t_n+t_{n+1}}{2}} \exp(j2\pi f_m t) dt = \\ &= \frac{j}{2\pi f_m} \sum_{n=0}^{N-1} x_n \exp(j2\pi f_m t_n) (1 - \exp(-j2\pi f_m \Delta t'_n)), \end{aligned} \quad (4.9)$$

$$\text{kur } \Delta t'_n = \frac{t_{n+1} - t_{n-1}}{2}.$$



4.3. zīm. Spektra novērtējums mono-harmoniskam signālam, ja pielietota pirmās kārtas polinomu interpolācija.



4.4. zīm. Spektra novērtējums mono-harmoniskam signālam, ja pielietota interpolācija ar sinc funkcijām.

Gabaliem lineārai interpolācijai starp signāla nolasēm var izmantot polinomus $p_n^1 = \alpha_n(t - t_n) + x_n$, kur $\alpha_n = \frac{\Delta x_n}{\Delta t_n}$, $\Delta x_n = x_n - x_{n-1}$, $\Delta t_n = t_n - t_{n-1}$, kas dod:

$$X_m^{(x1)} = X_m^{(x0)} + \frac{1}{(2\pi f_m)^2} \cdot \sum_{n=0}^{N-1} \alpha_n \exp(j2\pi f_m t_n) (1 - \exp(-j2\pi f_m \Delta t_n)) + \frac{i}{2\pi f_m} \sum_{n=0}^{N-1} \alpha_n \Delta t_n \exp(j2\pi f_m t_n) \exp(-j2\pi f_m \Delta t_n). \quad (4.10)$$

Cita veida pieeja ir veikt $\{x_n\}$ interpolāciju ar sinc funkcijām. Tā kā signāla frekvenču joslas platums ir ierobežots ar Ω , varam rakstīt:

$$\tilde{x}^{(\text{sinc})}(t) = \sum_{k=0}^{K-1} c_k \text{sinc}(2\Omega(t - kT)), \quad (4.11)$$

kur $T = \frac{1}{2\Omega}$, $K : kT < \Theta$. Koeficientus c_k varam atrast no sekojošas vienādojumu sistēmas:

$$x_n = \tilde{x}^{(\text{sinc})}(t_n). \quad (4.12)$$

Šajā gadījumā DFT-veidīgā transformācija ir nosakāma ar

$$X_m^{(\text{sinc})} = \sum_{k=0}^{K-1} c_k \exp(-j2\pi f_m kT). \quad (4.13)$$

Jāatzīmē, ka pēdējais apskatītais gadījums papildus DFT aprēķinu sarežģītībai pieprasa arī lineāras vienādojumu sistēmas ar N nezināmajiem un $2M+1$ vienādojumiem atrisināšanu.

4.3. DFT uzlabojums ar atjaunošanas kļūdas nolašu interpolācijas palīdzību.

Līdzīgi signāla nolašu interpolācijai, arī atjaunošanas kļūdas laika funkcija $\tilde{\varepsilon}(t)$ var tikt iegūta no tās diskrētām nolasēm $\varepsilon_n = x_n - \hat{x}_n$ laika momentos $\{t_n\}$. Atgādināsim, ka atjaunotā signāla nolases tiek aprēķinātas kā $\hat{x}_n = \sum_m X_m \exp(j2\pi f_m t_n)$. Uzdevums (4.5) šajā gadījumā var tikt interpretēts kā laukuma zem funkcijas $\tilde{\varepsilon}^2(t)$ minimizācija. Ja ε_n^2 ir interpolēts ar nulltās kārtas polinomiem, tad minimizācijas uzdevums ir pierakstāms kā:

$$\sum_{n=0}^{N-1} \left(x_n - \sum_{m=-M}^M X_m^{(\varepsilon_0)} \exp(j2\pi f_m t_n) \right)^2 \cdot \Delta t'_n \rightarrow \min. \quad (4.14)$$

Pēc atvasinājumu pielīdzināšanas nullei, iegūstam

$$2 \sum_{n=1}^N \left(x_n - \sum_{k=-M}^M X_k^{(\varepsilon_0)} \exp(j2\pi f_k t_n) \right) \exp(j2\pi f_m t_n) \Delta t'_n = 0 \quad (4.15)$$

Pārrakstot matricu formā, iegūstam

$$\Psi \mathbf{x} = \mathbf{X}^{(\varepsilon_0)} \cdot \Phi \cdot \Psi^T, \quad (4.16)$$

kur $\varphi_{mn} = \exp(j2\pi f_m t_n)$ un $\psi_{mn} = \varphi_{mn} \cdot \Delta t'_n$. Izteiksmes (4.16) risinājums var tikt atrasts kā

$$\mathbf{X}^{(\varepsilon_0)} = (\Psi \cdot \mathbf{x}) \cdot (\Phi \cdot \Psi^T)^{-1}, \quad (4.17)$$

kur $(\cdot)^T$ un $(\cdot)^{-1}$ attiecīgi apzīmē transponēšanas un invertēšanas operācijas.

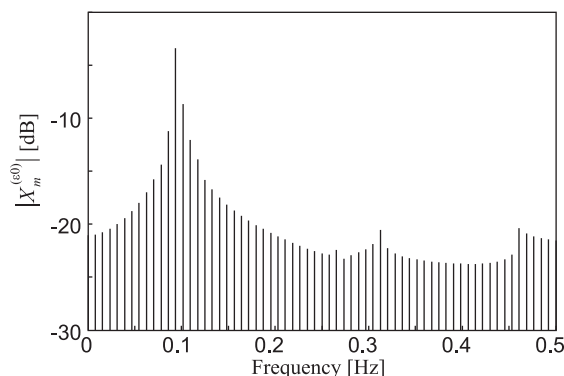
Interpolācija ar pirmās kārtas polinomiem noved pie sekojoša minimizācijas uzdevuma noformulēšanas:

$$\frac{1}{2} \left(\sum_{n=0}^{N-2} \varepsilon_n^2 \cdot \Delta t_n + \sum_{n=1}^{N-1} \varepsilon_n^2 \cdot \Delta t_{n-1} \right) \rightarrow \min. \quad (4.18)$$

Tas ir uzskatāms līdzīgi kā divu nulltās kārtas problēmu summa. Risinājuma gaita šajā gadījumā ir līdzīga iepriekš aprakstītajai un varam iegūt::

$$\mathbf{X}^{(\varepsilon_1)} = (\Psi'' \cdot \mathbf{x}' + \Psi'' \cdot \mathbf{x}'') (\Phi'' \cdot \Psi''^T + \Phi'' \cdot \Psi''^T)^{-1}, \quad (4.19)$$

kur $\Phi', \Psi', \mathbf{x}'$ un $\Phi'', \Psi'', \mathbf{x}''$ matricas ir veidotas no Φ, Ψ, \mathbf{x} lietojot attiecīgi indeksāciju $n = \overline{0, N-2}$ un $n = \overline{1, N-1}$.



4.5. zīm. Spektra novērtējums mono-harmoniskam signālam, ja atjaunošanas kļūdas funkcijai pielietota nulltās kārtas polinomu interpolācija.

4.4. Uzlaboto DFT-veidīgo pārveidojumu modelēšanas rezultāti.

Iepriekš aprakstīto algoritmu darbība tika novērtēta veicot datormodelēšanu. Šajā sakarībā valodā MATLAB ir uzrakstīta programmatūra, kuras izejas teksti ir pievienoti 7.D pielikumā. Kā testa signāls tika izmantota mono-harmoniska signāla nolašu kopa, kas iegūta no $\Theta = 128$ [sec] gara novērojuma izmantojot 7 līmeņu šķērsojuma notikumus. Testa signāla frekvence tika izvēlēta $12.22 / \Theta \approx 0.0955$ [Hz], kas nozīmē, ka tā neatrodas uz Furjē analīzes frekvenču režģa. Ar LŠ-ACP pieeju tika iegūtas $N = 172$ nolases.

Ar vispārināto DFT (izteiksme (4.3)) iegūtais rezultāts parādīts 4.1. zīm. Tas var tikts izmantots kā reference, lai salīdzinātu uzlabojumu kvalitāti. 4.3 zīm. rāda spektrālo novērtējumu, kas iegūts ar pirmās kārtas polinomu signāla interpolācijas algoritmu (izteiksme (4.10)). Lai gan viltus komponentu amplitūdas ir samazinājušās (apmēram par 5dB), to klātbūtne spektrā tomēr ir jūtami ievērojama. Aprēķinu sarežģītība ir dažas reizes lielāka par DFT. Nulltās kārtas polinomu interpolācijas izmantošana dod nedaudz sliktākus rezultātus.

Frekvenču joslā ierobežota interpolācija ar sinc funkcijām (izteiksme (4.13)) sniedz uzlabotāku spektrālo novērtējumu, kas ilustrēts 4.4. zīmējumā. Viltus komponentu klātbūtne ir pilnībā novērsta. Jāatzīmē, ka šī algoritma sarežģītība ir būtiski lielāka kā iepriekš minēto, jo ir nepieciešamība pēc lineāras vienādojumu sistēmas atrisināšanas.

Līdzīgu kvalitāte un sarežģītība piemīt arī metodēm, kas balstās uz atjaunošanas kļūdas nolašu interpolāciju. Abos gadījumos (0-tās un 1-ās kārtas polinomi) iegūtie rezultāti ir līdzīgi, tāpēc te 4.5. attēlā ir parādītas tikai nulltās kārtas polinomu interpolācijas gadījums.

Dažādu DFT-veidīgo pārveidojumu kvalitāte kvantitatīvi tikai novērtēta izmantojot integrālās atjaunošanas kļūdas kritēriju. Iegūtie rezultāti apkopoti 4.1. tabulā.

4.1. Tabula. Integrālā atjaunošanas kļūda dažādiem DFT-veida algoritmiem.

Metode	Vispārinātais DFT	Nulltās kārtas x_n interpolācija	Pirmās kārtas x_n interpolācija	x_n interpolācija ar sinc	Nulltās kārtas ε_n interpolācija	Pirmās kārtas ε_n interpolācija
$\int_0^{\Theta} \varepsilon^2(t) dt$	835	811	803	3.00	2.35	2.29

Visu DFT-veidīgo pārveidojumu kopīga iezīme ir tas, ka tie dod rezultātus uz Furjē frekvenču režģa. Tas ierobežo spektrālo izšķiršanu. Lai to palielinātu var izmantot sarežģītākas metodes, piemēram, signālatkarīgo pārveidojumu [7]. Ar to būtu saistāmi nākamajos projekta etapos veicamie pētījumi, kuros apskatīt kā var paaugstināt rezultātu kvalitāti, ja izmanto gan signālatkarīgu ACP, gan arī signālatkarīgas apstrādes metodes.

5. DASP virtuālo instrumentu attīstīšana izmantojot modificētos algoritmus

Projekta izpildes laika tiek radīti jauni, oriģināli algoritmi. To veidošana nav pašmērķis, bet gan tie ir domāti kā līdzeklis dažādu signālu apstrādes uzdevumu risināšanai. Tāpēc iegūtos algoritmus ir jācenšas ieviest praktiski pielietojamās sistēmās. Viens no šādiem novirzieniem ir saistīts ar algoritmu implementēšanu LabVIEW. Tā ir ērti lietojama grafiskā programmēšanas vide, kas ir paredzēta beigu lietotājam, kura programmēšanas zināšanas nav tik augstas. Iepriekšējo ciparotāju programnodrošinājums tika veidots MatLab un LabWindows vidēs, kas prasa padziļinātas zināšanas programmēšanā. Pielietojot LabVIEW ir iespējams veidot programmu izmantojot gatavos LabVIEW blokus, kas ir sadalīti divās kategorijās: grafiskā lietotāja interfeisa izveidei un algoritmu izveidei. Turpmāk ir apskatīts platjoslas ciparotāja vadības bloka izveide, izmantojot gatavos blokus, kā arī jaunradīto algoritmu implementācijas iespējas.

5.1. Platjoslīga un plaša dinamiskā diapazona ciparotāja vadība LabVIEW vidē

LabVIEW jau ir iebūvēti vairāki gatavi bloki, kas darbojas ar standarta iekārtām un lietotājam, kas pielieto standarta risinājumus nav nepieciešams veidot savus vadības blokus. DIGITIZER B042 ar datoru tiek saslēgts caur paralēlo portu, kas darbojas EPP 1.9 režīmā, diemžēl LabVIEW neuztur EPP 1.9 režīmu. Šī iemesla dēļ ir izveidota dinamisko saišu bibliotēka (DDL), kura ir importēta LabVIEW un ir izveidoti vadības bloki port_in un port_out. Pielietojot šos blokus ir iespējams datus nosūtīt un saņemt no ārējās iekārtas, kas strādā EPP 1.9 režīmā.

Pirms sākt ciparot datus ar DIGITIZER B042 ir nepieciešams iekārtu inicializēt, ko panāk nosūtīt sekojošas komandas:

Tabula 5.1. Iekārtas parametru nomešana

<i>Darbības Nr.</i>	<i>Darbības nosaukums</i>	<i>Adrese</i>	<i>Dati</i>
1	Iestāda rakstīšanas režīmu	BASE+2	0x01
2	Iestāda rakstīšanas režīmu	BASE+2	0x05
3	Iestāda visu parametru nomešanas režīmu	BASE+3	0x01
4	Visu iekārtas parametru nomešana	BASE+4	0x00
5	Nolasa status reģistru	BASE+1	

Tabula 5.2. Raksta datus FIFO atmiņā

<i>Darbības Nr.</i>	<i>Darbības nosaukums</i>	<i>Adrese</i>	<i>Dati</i>
1	Iestāda EPP rakstīšanas režīmā	BASE+2	0x05
2	Iestāda visu parametru nomešanas režīmu	BASE+3	0x01
3	Visu iekārtas parametru nomešana	BASE+4	0x00
4	Iestāda diskretizācijas režīmu Uniform NonUniform	BASE+3	0x0A 0x02
5	Sāk datu ciparošanu	BASE+4	0x00

Tabula 5.3. Lasīt datus no FIFO atmiņas uz datoru

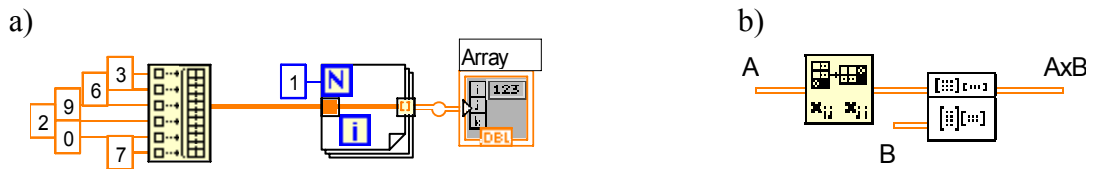
<i>Darbības Nr.</i>	<i>Darbības nosaukums</i>	<i>Adrese</i>	<i>Dati</i>
1	Iestāda EPP rakstīšanas režīmā	BASE+2	0x05
2	Iestāda iekārtu FIFO lasīšanas režīmā	BASE+3	0x40
3	Iestāda EPP lasīšanas režīmā	BASE+2	0x04
4	Nolasa datus no FIFO atmiņas	BASE+4	

Tā kā DIGITIZER B042 ir 12bitu ACP, bet paralēlais ports ir tikai 8 bitu, tad dati tiek speciāli formēti, lai tos noraidītu uz datora. No sākuma noraida pirmos 8 bitus un pēc tam atlikušos 4, bet atlikušos 4 bitus aizpilda ar 0. Kad dators ir saņēmis datus, tad ir nepieciešams datus konvertēt atpakaļ uz 12 bitu datiem. Pēc visām šīm darbībām notiek signāla attēlošana un apstrāde.

Apstrādes algoritmu implementācija un grafiskā lietotāja interfeisa izveide ir aprakstīta nākošajās apakšnodaļās.

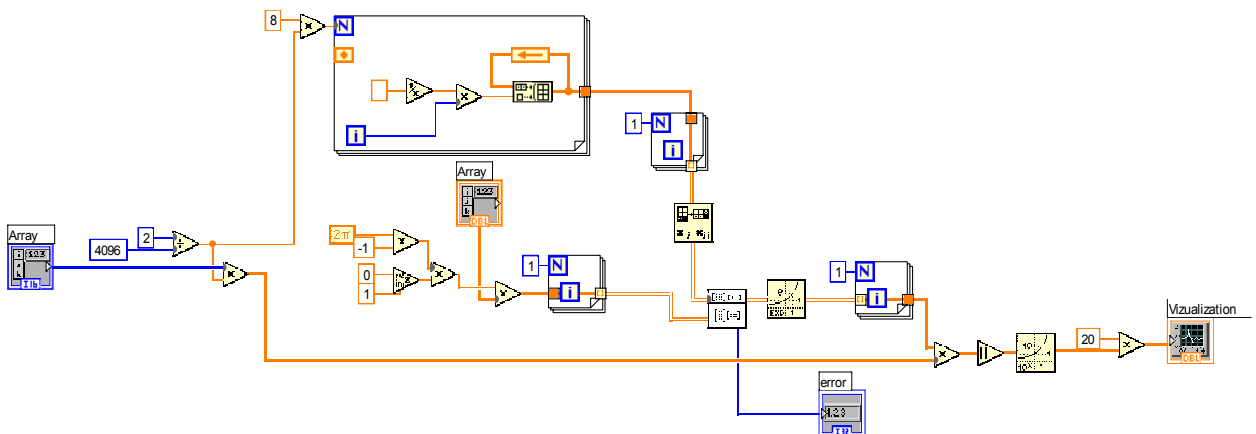
5.2. Jaunradīto signālu apstrādes algoritmu implementācija LabVIEW

LabVIEW ir plašs matemātisko darbību bloku skaits, ar kuru palīdzību var realizēt jebkuru matemātisku darbību. Signālu apstrādes algoritmu pamatā ir darbības ar vektoriem, masīviem un matricām. Lai LabVIEW sareizinātu divus vektorus tie ir jānoformē kā matricas, kuru viena dimensija ir 1. To var panākt viendimensiju masīvu laižot cauri *for* ciklam, kurš atkārtojas tikai vienu reizi (Skat zīm. 5.1.a). Tā kā *for* cikls vektoram piešķir indeksus, tad cikla izejā ir matrica. Pēc pārvēršanas pirmo matricu transponē un sareizina ar otro matricu, izmantojot $A \times B$ bloku (Skat. zīm. 5.1.b). Lai matricu pārvērstu atpakaļ vektorā, matricu nepieciešams laist cauri *for* ciklam, kuram izejā ir izslēgti indeksi.



Zīm.5.1. a) vektora pārvēršana par viendimensiju matricu, b) matricu reizināšana.

Pilnu DFT algoritma implementācijas grafisko shēmu var redzēt zīm. 5.2.

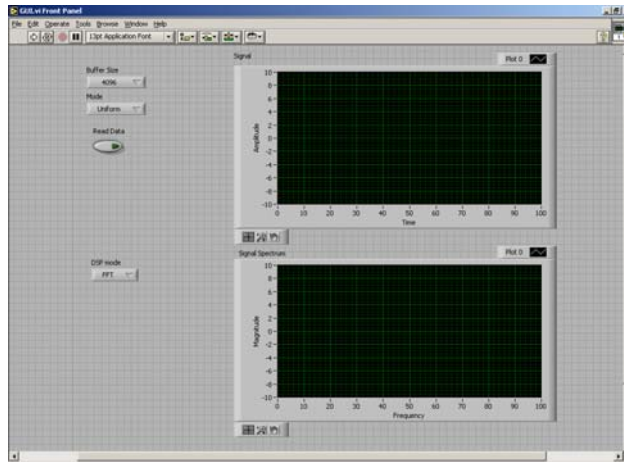


Zīm. 5.2. DFT grafiskā implementācija LabVIEW.

Ar LabVIEW palīdzību grafiskā veidā var realizēt jebkuru algoritmu. Diemžēl šis process ir darbietilpīgāks, ja salīdzina tāda paša algoritma izveidi C/C++ valodā. Šai gadījumā gan jāpiemin cits pozitīvais aspekts. Gala lietotājam nav jāpatērē daudz laika, lai izprastu algoritma darbības principu un nepieciešamības gadījumā viņš var viegli un ērti uzlabot algoritmu.

5.3. Grafisko Lietotāju Interfeisu izveide

LabVIEW ir ērti un vienkārši veidot grafisko lietotāja interfeisu. Atverot jaunu virtuālā instrumenta priekšējo paneli un nospiežot labo peles pogu parādās izvēlne, kurā var izvēlēties grafiskās ierīces, tas ir, kontroles, indikatorus, izvēlnes utt. Izvietojot tās uz darba virsmas grafiskā vide ātri un ērti ir izveidota. Zem grafiskās vides slēpjas visu grafisko elementu savienojumi ar vadības apstrādes algoritmu blokiem.



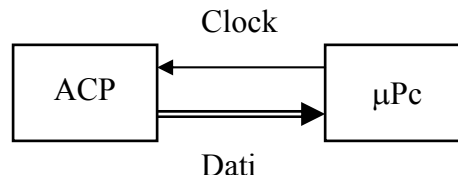
Zīm. 5.3. Virtuālā instrumenta grafiskais interfeiss.

6. Asinhrono mikroprocesoru pielietojamība ar beztaktētāja analogs-ciparu pārveidotāju iegūtu datu energoekonomiskai apstrādei

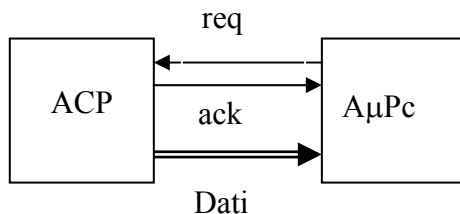
6.1 Enerģijas patēriņa minimizācijas modeļi

Enerģijas patēriņa minimizācijai vienkāršā signālu apstrādes sistēmā (zīm.6.1), iespējamas vairākas alternatīvas:

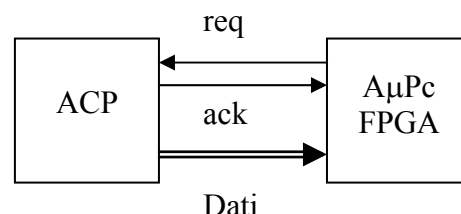
1. nepārtraukta periodiska datu plūsmas nolase no datu avota katrā analogo datu pārveides ciklā, izmantojot katru ārējo procesora sinhronizācijas periodu, un datu apstrāde ar sinhrono (standarta) mikroprocesoru;
2. datu nolase atbilstoša p.1., bet datu apstrāde ar programmējamu DVS un DM mikroprocesoru;
3. stohastiska datu nolase no datu avota momentos, kad ieejas signāla U_a vērtības izmaiņa $|\Delta U_a| > \delta$, izmantojot asinhrona procesora rokspiešanas (*handshaking*) porta signālus req – ack, un datu apstrādi šajā procesorā, zīm. 6.1.a;
4. datu nolase atbilstoša p.3. bet rokspiešanas un datu apstrāde organizācija asinhronā rekonfigurējamā FPGA, zīm. 6.1.b.



Zīm. 6.1



Zīm. 6.1.a



Zīm. 6.1.b

Lai aptuveni novērtētu un salīdzinātu šās sistēmas pieņem, ka nosacījums $|\Delta U_a| > \delta$ izpildās vienmērīgi izkliedētos laika intervālos $\tau = 1, 2, 3, 4, 5$ ms, bet katra pieprasījuma apstrāde $t_E = 1$ ms, kurās sistēmā 1. tiek patērēta enerģija 10×1 . Sistēmā 1. un 2. ieejas signāla aptauja notiek ar 0,1ms garu strobējošu signālu pēc laika intervāla $t_s = 1$ ms, t.i., pie katras strobēšanas patērējot enerģiju $10 \times 0,1$. Vienā regulārā ciklā 1.sistēmā aprēķinam tiek patērēta enerģija $10 \times 5 = 50$ un strobēšanai $1 \times 15 = 15$ $\Sigma E = 65$. Otrā tipa sistēmās izmantojot DVS, pamatojoties uz [18] veiktajiem aprēķiniem, var pieņemt ka laika intervālā τ_1 enerģijas patēriņš $10 \times 1 = 10$, $\tau_2 = 4 \times 2 = 8$, $\tau_3 = 1 \times 3 = 3$, $\tau_4 = 0,5 \times 4 = 2$, $\tau_5 = 0,1 \times 5 = 0,5$ ($E_p = 23,5$), strobēšanai $E_s = 15$: $\Sigma E = 38,5$.

Sistēmā 3., kurā izmantots asinhronais procesors, nav ieejas strobēšanas signālu un enerģija strobēšanai netiek tērēta $E_s = 0$, bet instrukcijas izpildei patērētā enerģija aptuveni var būt $\times 5$ mazāka par enerģijas patēriņu analogā sinhronā procesorā [19].

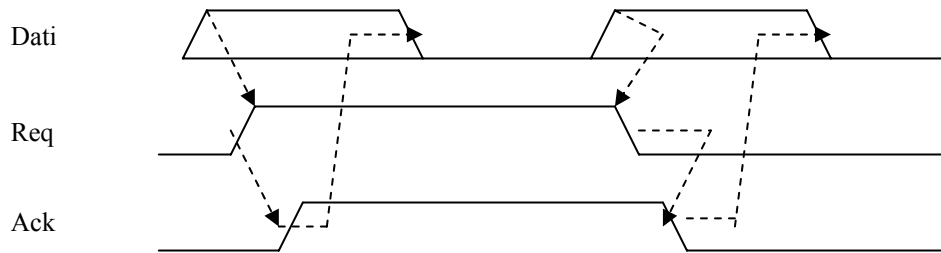
Tāpēc var uzskatīt, kopējais enerģijas patēriņš $\Sigma E = 0 + 5 \times 10/5 = 10$.

Sistēma 4. strobēšanai enerģija netiek tērēta $E_s = 0$, bet pārējais pagaidām miglā tīts.

6.2 Asinhronā procesora enerģijas taupības iespējas

A. Asinhronās funkcionālās ierīces patērē enerģiju tikai darot lietderīgu darbu. Turpretim sinhronajā procesorā takts frekvenci jāizvēlas tā, lai procesors varētu izpildīt uzdevumu pie maksimālās slodzes. Tāpēc procesora takts frekvence bieži ir augstāka kā nepieciešams, un rezultātā tiek lieki tērēta enerģija.

B. Asinhronās funkcionālās ierīces apmaiņas ar datiem izmantojot asinhrono „Micropipelines” rokspiešanas protokolu (zīm. 6.2.)



zīm. 6.2.

Realizējot šo protokolu:

1. Sūtītājs (Sender) uz kopnes uzliek datus (Data)
2. Sūtītājs ģenerē signālu Req (*Request*)
3. Saņēmējs (Receiver), kad ir tam gatavs, akceptē datus
4. Saņēmējs ģenerē un noraida Sūtītājam signālu Ack (*Acknowledge*)
5. Pēc *Acknowledge* saņemšanas Sūtītājs var noņemt no kopnes datus un ir gatavs jaunu datu nosūtīšanai

Asinhronajam „Micropipelines” protokolam ir divi realizācijas varianti:

- a) Variantā, kas parādīts zīm.6.2. signālu ģenerēšana notiek pie sprieguma līmeņu augošas vai krītošas izmaiņas, bet, kā redzams, *Req* un *Ack* var saglabāties arī laikā, kad datu pārraide nenotiek (lieks enerģijas patēriņš!) [20]
- b) Variantā, kas ieguvis nosaukumu ‘līmeņa signalizācija’, izmantojot krītošās un augošās spriegumu maiņas, tiek veidoti īsi *Req* un *Ack* signāli, kas ir aktīvi tikai datu pārraides laikā un lieku enerģiju nepatērē [21].

C. Visi asinhronā procesora mezgli (bloki) funkcionē pēc konveijera principa (*pipeline*), bet šos konveijerus iespējams apstādināt jebkurā darbības posmā (fāzē), pēc kuras enerģija vairs netiek patērēta.

D. Parasti netiek pielietota kešatmiņa, kuras darbināšana saistīta ar ievērojamu enerģijas patēriņu reģistros un statiskajā brīvpiekļuves atmiņas (RAM) šūnās.

E. Veidojot ātrdarbīgus (32 bitu) asinhronus RISC procesoru par bāzi izvēlēta populārā ARM arhitektūra, kas pateicoties izdevīgajai instrukciju kopai, nodrošina visai mazu enerģijas patēriņu.

6.3 Asinhrono procesoru raksturojumi un lietojums

AMULET1 izstrāde Oksfordas Universitātē parādīja iespēju izveidot pilnīgi asinhronu ARM arhitektūras procesoru. Tomēr enerģijas patēriņš 83mW pārsniedza sinhronā ARM6 patēriņu 75mW [20].

AMULET2e tiek pilnīgāk nekā -AMULET1 izmantotas asinhronās kopnes priekšrocības (augstāk apskatītais *Micropipeline b*) variants). Enerģijas patēriņš tika samazināts [21].

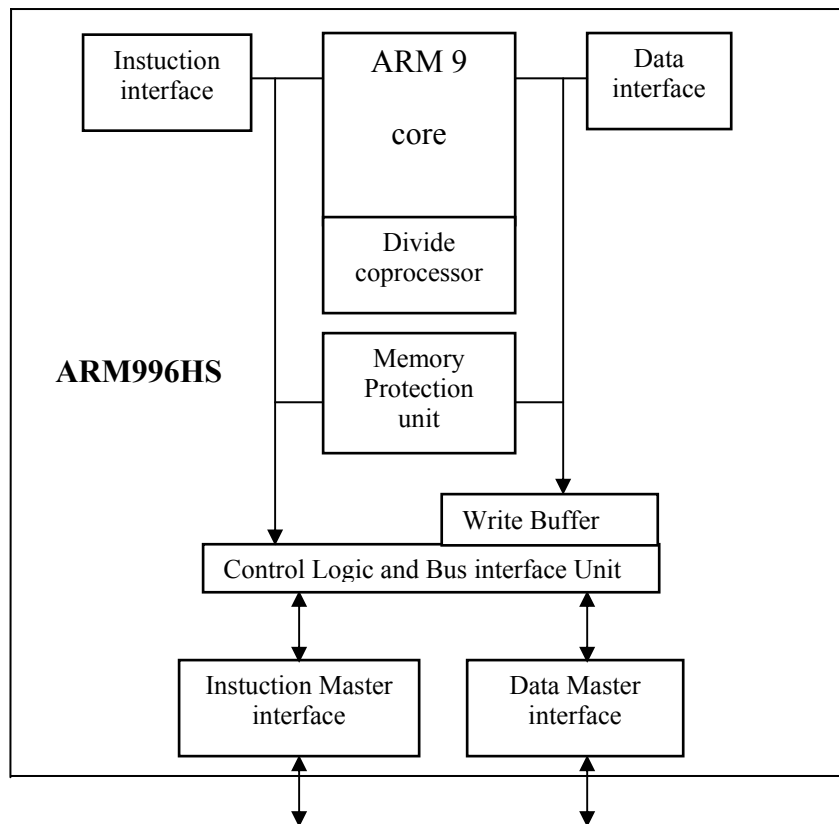
AMULET3 kodolā izmantota ARM4T versija ar jaunu iekšējo mikroarhitektūru. Izpildes kārtības maiņas bufera ieviešana atļauj paaugstināt ātrdarbību līdz 100MIPS. Enerģijas patēriņa ietaupījumu panāk:

1. izmantojot speciālu HALT instrukciju, kas momentā pārtrauc instrukciju priekšnolasīšanu (prefetching) un likvidē izpildmezglā enerģijas patēriņu,
2. sadalot atmiņu mazos 1k blokos,
3. ARM 32 bitu instrukciju vietā, izmantojot vienkāršotu 16 bitu Thumb instrukciju kopu [22].

AMULET3i izveidota saglabājot AMULET3 kodolu un balstoties uz asinhrono kopni MARBLE. Paredzēts uz AMULET3i bāzes uzsākt ražot ierīci Draco - Radio Communsiation Controller.

Praktiskus risinājumus, kas novesti līdz ražošanas uzsākšanai devis firmas Philips uzņēmums „Handshake Solution”. Te uzsākta divu veidu asinhronu procesoru ražošana.

1) RISC arhitektūras 32 bitu ARM996HS:



Zīm. 6.3

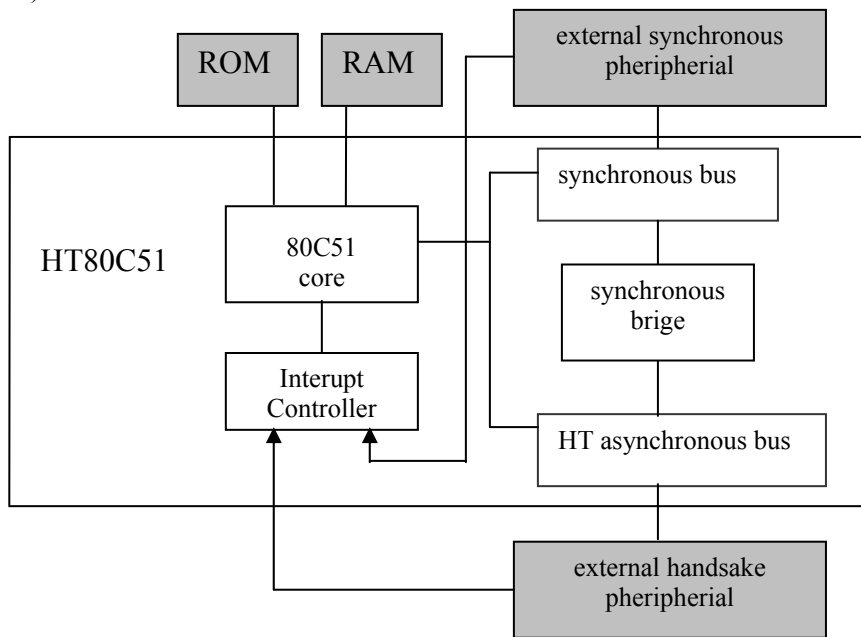
ARM996HS arhitektūra: 32 bitu un 16 bitu Thumb instrukciju kopas u.c. rādītāji atbilst populārajam sinhronajam procesoram ARMT9E, kas pats par sev ir ARM arhitektūras procesors ar mazāko enerģijas patēriņu un ir piemērots reālā laika lietojumiem[23].

ARM996HS spēj automātiski adaptēties apkārtējās vides (temperatūras, barošanas sprieguma) maiņām.

Saskarne ar ārējām ierīcēm tiek nodrošināta izmantojot ar enerģiju maztērējošo interfeisa moduli (master) AMBA un rokspiešanas principu.

Rokspiešanas testprogrammas realizācijas ir parādījuši, ka pie vienādas ātrdarbības ARM996HS enerģijas patēriņš ir bijis 0.045mW, bet sinhronā procesora ARMT9E sasniedzis 0.13 mW [23].

2) Mikrokontrollera HT80C51:



Zīm. 6.4

HT80C51 atbilst vienam no populārākajiem mikrokontrolieriem 80C51 un var pildīt visas 80C51 izveidotās programmas, kā arī 80C51 atmiņas un ārējo iekārtu interfeisu.

Bez zīm.4. parādījām komponentēm HT80C51, sastāvā ir arī 2 taimeri, kā arī UART, SPI un I²C kontrolleri [19].

HT80C51 var darbināt arī sinhronā režīmā.

Instrukcijas veikšanai HT80C51 patērē 89pJ, bet sinhronais 80C51 patērē 500pJ.

Miera stāvoklī (idle mode) HT80C51 enerģiju nepatērē.

7. Pētījumi datu apstrādes sistēmu realizācijai beztaktētāja elektroniskās iekārtās

7.1 Transformāciju ar taisnstūra funkcijām pielietojums

Tradicionāli signāla spektra aprēķināšanai izmantojot Furjē transformāciju (FT). Šeit spektra aprēķināšanai lieto divas sinusoīdas, vienu no otras nobīdītu fāzē par $\frac{\pi}{2}$. FT aprēķins ir relatīvi sarežģīts un prasa daudz laika. Elektroniskajās shēmās FT izmantošana prasa lielu strāvās patēriņu, pie tam ir jāveic reizināšanas operācija, kas ir diezgan grūti realizējama. Ja FT izmantotās sin un cos funkcijas aizstāj ar taisnstūra funkcijām, var saīsināt spektra aprēķina algoritmu, kā arī samazināt elektroniskās iekārtās enerģijas patēriņu, kas ir svarīgi iekārtām, kas darbojas ar enerģiju uzkrājošiem elementiem (baterijām, akumulatoriem utt.). Taisnstūrfunkciju FT koeficientus var novērtēt kā [8]:

$$a_n = \frac{1}{t_b - t_a} \int_{t_a}^{t_b} \text{sign}(t) \cdot \sin(2\pi \cdot f \cdot t) dt \quad (7.1)$$

kur t_a - laiks aplūkojamā signāla sākuma momentā
 t_b - laiks aplūkojamā signāla beigu momentā
 $\text{sign}(t)$ – aplūkojamais signāls

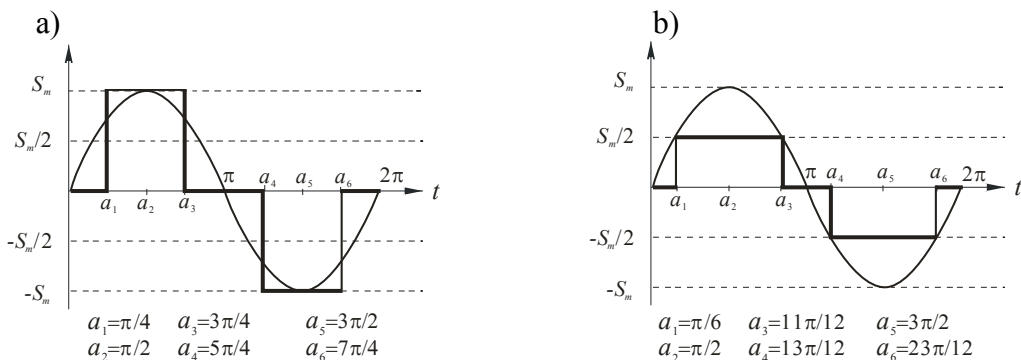
Šeit apskatīts tikai viens FT koeficients, otrs ir identisks tikai nobīdīts fāzē par $\frac{\pi}{2}$.

Izmantojot spektra aprēķināšanai taisnstūra funkciju, kura pieņem vērtību ± 1 , šī paša koeficienta izteiksmi var pierakstīt [1]:

$$a_n = \frac{1}{t_b - t_a} \int_{t_a}^{t_b} \text{sig}(t) \cdot R(f, t) dt = \frac{1}{t_b - t_a} \left[\int_{t_a}^{t_1} \text{sign}(t) dt - \int_{t_2}^{t_1} \text{sign}(t) dt + \int_{t_3}^{t_2} \text{sign}(t) dt - \int_{t_3}^{t_2} \text{sign}(t) dt + \dots \right] \quad (7.2)$$

kur $R(f, t)$ – taisnstūra funkcija

Kā redzams no izteiksmes, ja izmanto taisnstūra funkciju kā filtrējošo funkciju, tad reizināšanas darbība tiek aizstāta ar saskaitīšanu, kas ir daudz vienkāršāk realizējama operācija.



Zīm. 7.1.1. Taisnstūra funkcijas a) ar vienādu samēru, b) ar nevienādu samēru

Ja kā taisnstūra funkciju izmanto funkciju ar vērtībām ± 1 un 0 , tad attiecīgajām pētāmā signāla vērtībām jāpiešķir pluss vai mīnus zīme, bet tās vērtības, kuras atrodas nulles intervālā ir nulle un vispār nav jāņem vērā. Spektra aprēķināšanai tiek izmantotas divas identiskas taisnstūra funkcijas – viena attiecībā pret otru nobīdīta fāzē par $\frac{\pi}{2}$.

Kā bāzes funkcijas tika izmantotas:

Klasiskais meandrs

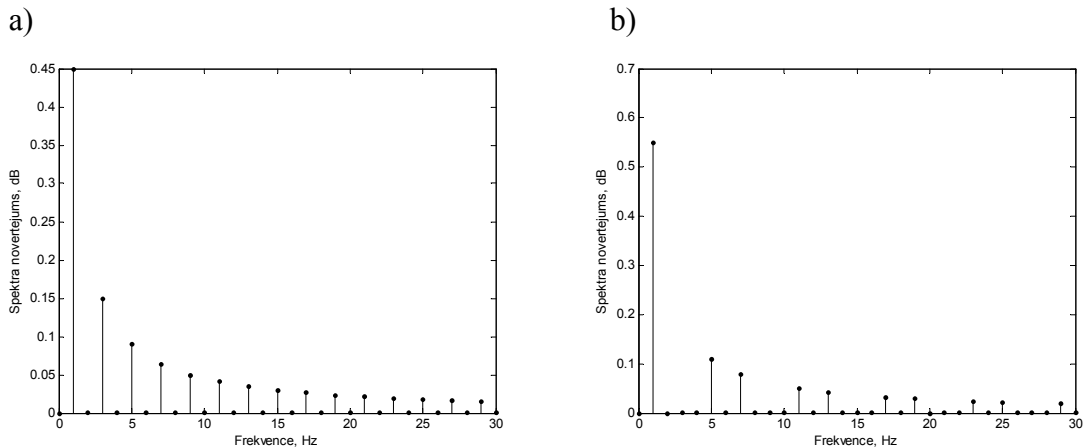
Taisnstūra funkcijas ar vienādu samēru un vērtībām ± 1 un 0 (zīm. 7.1.1 a)

Taisnstūra funkcijas ar nevienādu samēru un vērtībām ± 1 un 0 (zīm. 7.1.1 b)

Trepveidīgas funkcijas ar vērtībām $\pm 1; \pm 2$ un 0

Volša funkcijas

Kā bāzes funkcijas izmantojot Volša funkcijas, signāla spektrā ir daudz viltus komponentes. Tas skaidrojams ar to, ka Volša funkciju spektrs ir lēni rimstošs, pie tam no visas Volša funkciju saimes dažām funkcijām ir gandrīz vienādas spektra blakus komponentes. Arī meandra un trepveidīgo taisnstūra funkciju ar vienādu samēru izmantošana par bāzes funkcijām dod sliktus rezultātus dēļ lēni rimstošā spektra. Veicot taisnstūra funkcijas Furjē analīzi, redzams, ka šo funkciju spektrā ir nepāra komponentes (zīm. 7.1.2 a), kas iespaido pētāmā signāla spektru (zīm.7.1.3 a).



Zīm. 7.1.2 Taisnstūra funkciju spektrs a) ar vienādu samēru; b) ar nevienādu samēru

Zīm. 7.1.3 a attēlots spektrs sinusoīdai ar frekvenci 15 Hz, kas diskretizēta ar vienmērīgu diskretizāciju [9], spektra aprēķinam izmantojot taisnstūra funkciju ar vienādu samēru un vērtībām ± 1 un 0 (zīm. 7.1.1 a). Zīm. 7.1.3 b attēlots spektrs sinusoīdai, kas diskretizēta ar līmeņu šķērsošanas ACP, izmantojot 4 bitu līmeņus un iepriekš minēto taisnstūra funkciju. Šeit spektrā parādās daudzas viltus komponentes, kuras nav vienmērīgi diskretizēta signāla spektrā. Dažas no spektra viltus komponentēm var samazināt, izmantojot taisnstūra funkcijas ar nevienādu samēru (zīm. 7.1.1 b). Ja taisnstūra funkcijai ar vērtībām ± 1 un 0 vieninieka attiecība (pret visu funkcijas viena perioda garumu) ir $2/3$, bet nulles attiecība $1/3$, var iegūt spektru, kurā būs pazudušas 3, 9, 15 ... komponentes (zīm. 7.1.2 b) [8]. Izmantojot šādu funkciju kā bāzes funkciju signāla spektra aprēķinam, pazūd attiecīgās viltus komponentes gan vienmērīgi diskretizēta signāla spektrā (1 un 5 komponente (zīm. 7.1.3 c)), gan ar līmeņu šķērsošanu diskretizētā signāla spektrā (zīm. 7.1.3 d). Bet pie līmeņu šķērsošanas diskretizācijas paliek visas pārējās viltus komponentes, gan nedaudz samazinājušās. Tas liek domāt, ka bāzes funkcijas izvēle maz ietekmē tās viltus komponentes, kuras atrodas aiz signāla frekvences (šajā gadījumā 15 Hz). Palikušās viltus komponentes var vēl samazināt, ņemot vērā laukumu starp divām blakus esošajām diskretizētā signāla nolasēm (delta -

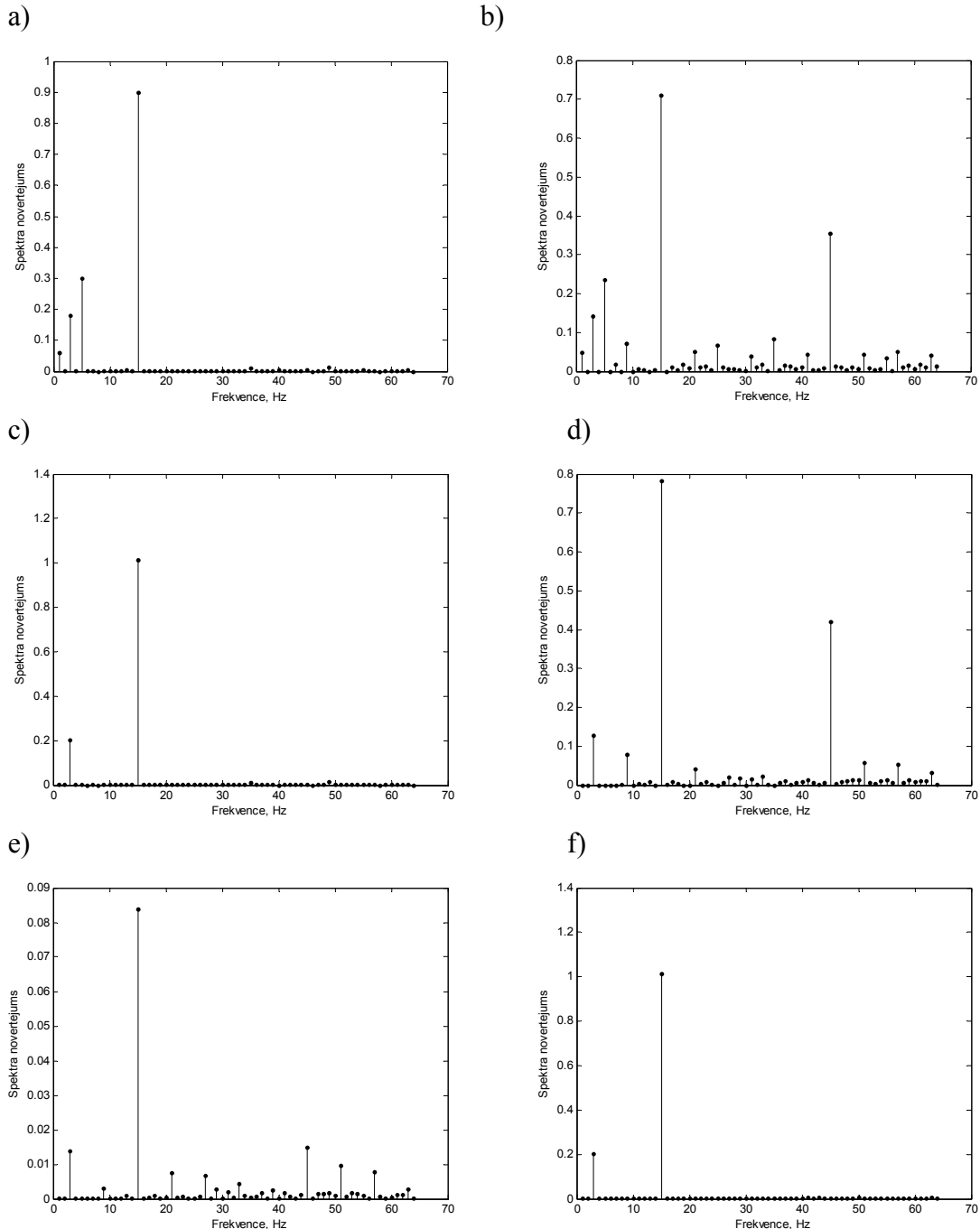
izmaiņa). Tas nozīmē: no pašreizējās nolases laika vērtības atņem iepriekšējās nolases laika vērtību un iegūto starpību reizina ar pašreizējās nolases amplitūdu:

$$S = (t_t - t_v) \cdot A \quad (7.3)$$

kur t_t – tekošais laika moments,

t_v – iepriekšējais laika moments;

A – diskretizētā signāla amplitūda laika momentā t_t



Zīm. 7.1.3 Spektrs a) vienmērīgai diskretizācijai ar vienādu taisnstūra funkcijas samēru; b) līmeņu šķērsošanas diskretizācijai ar vienādu taisnstūra funkcijas samēru; c) vienmērīgai diskretizācijai ar nevienādu taisnstūra funkcijas samēru; d) līmeņu šķērsošanas diskretizācijai ar nevienādu taisnstūra funkcijas samēru; e) līmeņu šķērsošanas diskretizācijai ar nevienādu taisnstūra funkcijas samēru, ņemot vērā laukumu pirms spektra komponentes (delta-izmaiņa); f) nevienmērīgai diskretizācijai ar nevienādu taisnstūra funkcijas samēru.

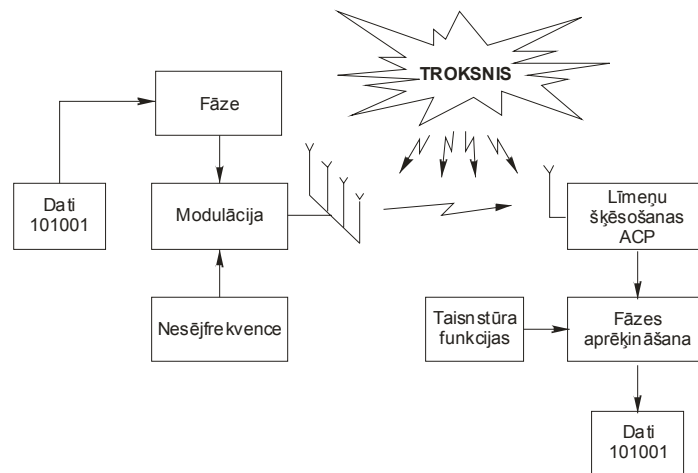
Izmantojot šādu paņēmienu, ievērojami samazinās kā viltus komponentes, tā arī signāla īstās komponentes (zīm. 7.1.3 e), pie kam viltus komponentu samazināšanās ir daudz ievērojamāka. Tas ļauj izdalīt no visa spektra īstās signāla komponentes.

Veicot signāla (šajā gadījumā sinusoīdas) nevienmērīgo diskretizāciju, tas ir – laika momenti, kad signāls tiek diskretizēts tiek izvēlēti nejauši (zīm. 7.1.3 f), spektrā pazūd visas komponentes, kas nav saistītas ar taisnstūra funkcijas spektru. Spektrs sāk līdzināties vienmērīgi diskretizēta signāla spektram, ar atšķirību, ka netiek novērota spektra pārklāšanās, kas parādās vienmērīgi diskretizēta signāla spektrā dēļ diskretizācijas soļa periodiskuma. Nevienmērīgās diskretizācijas realizēšana elektroniskajā aparatūrā ir sarežģīta, tādēļ veiktajam pētījumam tā neder, jo mērķis ir radīt elektronisko aparatūru, kas patērētu mazāk enerģiju nekā pašreiz esošās un būtu ar ātrāku darbības algoritmu.

Daudzo viltus komponentu parādīšanās pie līmeņu šķērsošanas diskretizācijas skaidrojama ar to, ka nolases ir regulāras (ja diskretizējamais signāls ir regulārs). Tās liek secināt, ka līmeņu šķērsošanas diskretizācija ir efektīva, ja ieejas signāls ir nestacionārs. Stacionāru signālu gadījumā, taisnstūra funkcijas un līmeņu šķērsošanas diskretizāciju var pielietot, ja ir zināms, kāda ir uztveramā signāla frekvence (apskata signāla spektru šaurā joslā) un pārējās spektra komponentes tad ir atmetamas. Ja izmanto delta – izmaiņa spektra rēķināšanas metodi un taisnstūra funkcijas ar nevienādu samēru, līmeņu šķērsošanas diskretizāciju var lietot arī signāla uztveršanai ar iepriekš nezināmu frekvenci, jo viltus komponentes tiek ievērojami nospiesta.

7.2 GSM signālu apstrādes beztaktētāja sistēmas modelis

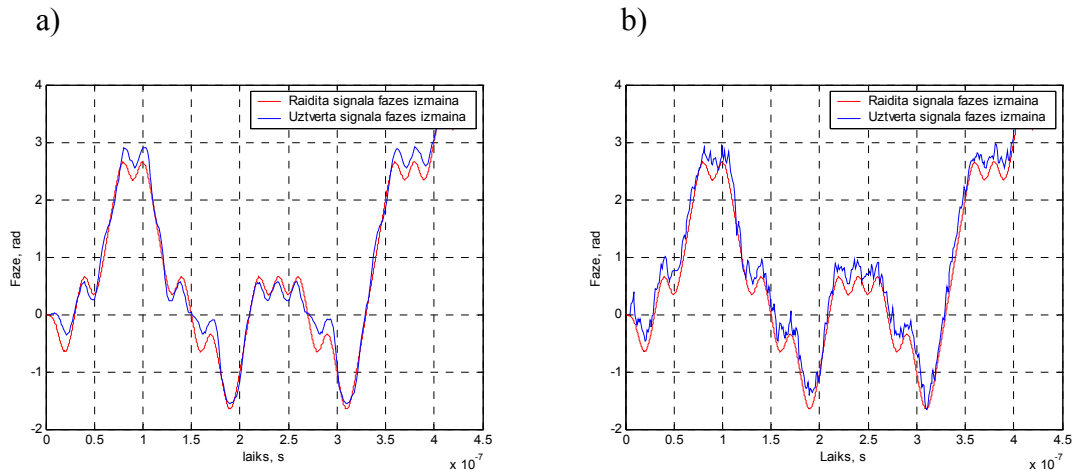
Darba mērķis ir izveidot Matlab vidē datu pārraides sistēmas modeli, kurš darbotos ar līmeņu šķērsošanas ACP un izmantotu taisnstūra funkcijas. Kā datu pārraides sistēma ir izvēlēta GSM sistēma, kurā izmanto GMSK modulāciju. Programmas ieejā ir dati binārā veidā (zīm. 7.2.1).



Zīm. 7.2.1. Beztaktētāja sistēmas modeļa shēma

Atkarībā no ieejas datu (nuļļu un vieninieku) secības, tiek ģenerēta fāzes izmaiņa. Lai fāzes izmaiņai (līdz ar to arī nesēj signāla izmaiņai) nebūtu asu lēcienus, tiek izmantots Gausa filtrs. Fāzes izmaiņa tiek pieskaitīta nesēj signāla fāzei (tiek veikta fāzes modulācija). Nesēj signāla frekvence ir 890.2 MHz. Viena simbola pārraides laiks ir $1 \cdot 10^{-8}$ sekundes. Šeit ir atšķirība no GSM sistēmas, kur viena simbola pārraides laiks ir 3,6923 mikrosekundes [10]. Samazinot simbola garumu, var iegūt daudz lielākus datu pārraides ātrumus. Šajā

gadījumā datu pārraides ātrums aptuveni nekā 360 reizes lielāks, kā GSM sistēmā. Lai modeli pietuvinātu reālajiem apstākļiem, modulētajam signālam pārraides laikā tiek pievienots baltais troksnis. Uztvērējā signāls tiek diskretizēts ar līmeņu šķērsošanas ACP. Diskretizētā signāla fāzes aprēķināšanai izmanto taisnstūra funkcijas ar vienādu samēru. Taisnstūra funkcijas pieņem vērtības ± 1 . Fāzi aprēķina nelielam signāla gabalam, kuru no visa kopējā signāla izgriež ar logu. Tad logu pārbīda tālāk un atkal aprēķina signāla fāzi. Iegūtās fāzes vērtības salīdzina ar signālu, kuram ir tāda pati frekvence kā nesējsignālam, bet nemainīga fāze. Atkarībā no fāzes izmaiņas, tiek atkodēti pārraidītie dati. Izmantojot līmeņu šķērsošanas ACP ar 5 bitiem, uztvertā signāla fāzes izmaiņa ir identiska raidītā signāla fāzes izmaiņai (zīm. 7.2.2a).

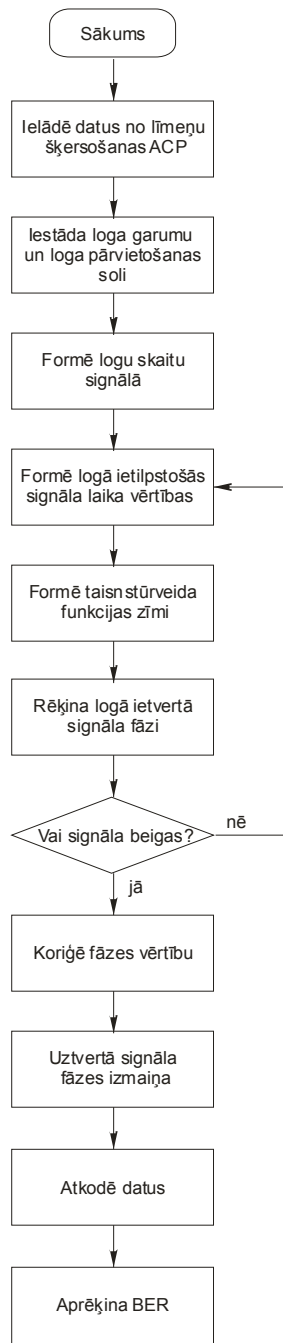


Zīm 7.2.2 Uztvertā un raidītā signāla fāzes izmaiņa a) ideālā kanālā b) kanālā ar troksni

Fāzes izmaiņu neietekmē taisnstūra funkcijas veida izvēle, tādēļ ir izdevīgi izvēlēties funkciju, kura pieņem tikai divas vērtības (tas ir ± 1). Rezultātā diskretizētā signāla amplitūdas vērtībām piešķir “+” vai “-” zīmi un visas vienā logā ietvertās vērtības saskaita (nav nepieciešams izmantot reizināšanas operāciju). Šāds algoritms ievērojami var ietaupīt iekārtas elektroenerģijas patēriņu, kā arī samazināt datu apstrādes laiku. Minimālais bitu skaits, kas nepieciešams diskretizējot signālu ar līmeņu šķērsošanas diskretizāciju, ir trīs. Izmantojot šādu ACP, signāla fāzes izmaiņu ir iespējams samērā precīzi aprēķināt. Taču krietni labāki rezultāti ir, ja izmanto 4 bitu līmeņu šķērsošanas ACP. Ja kanālam pievieno balto troksni ar SNR ir 67,7 dB (zīm. 7.2.2.b), fāzes izmaiņa tiek izkropļota, taču pārraidītos bitus ir iespējams atpazīt nekļūdīgi. Jāņem vērā, ka modeļa ieejā netiek izmantots filtrs, kas daļēji izfiltrētu uztveramo signālu no trokšņa.

Turpmākie darba mērķi:

- 1) Daudzkanālu GSM beztaktētāja sistēmas modeļa izstrāde.
- 2) Izveidot uztverto datu atpazīšanu ar koriģējošajiem kodiem. Pašreiz izstrādātajā modelī atkodējot kādu bitu kļūdai, tālāk sekojošie biti tiek nepareizi atpazīti.
- 3) Izveidot modeli, kura ieejā ir filtrs, kas izfiltrē signālu, tādējādi samazinot trokšņa līmeni.



Zīm 7.2.3. Fāzes atpazīšanas programmas shēma

Izmantotās literatūras saraksts

- [1] Proceedings of the 5th IEEE International Workshop on System-on-Chip for Real-Time Applications (IWSOC 2005), 20-24 July 2005., Banff, Alberta, Canada. IEEE Computer Society 2005, ISBN 0-7695-2403-6.
- [2] Hauck S. Asynchronous Design Methodologies: An Overview // Proceedings of the IEEE, Vol. 83, No. 1, pp. 69-93, January, 1995.
- [3] Logan B. Information in zero crossings of bandpass signals // Bell Sys. Tech. J. 56, 487-510 (1977).
- [4] Nazario MA. Saloma C. Signal recovery in sinusoid-crossing sampling by use of the minimum-negativity constraint // Applied Optics. 37(14):2953-2963, 1998
- [5] F. Aeschlimann, E. Allier, L.Fesquet, M.Renaudin. Spectral analysis of level-crossing sampling scheme // Proc. of the 2005 International Conference on Sampling Theory and Application (SampTA 2005), July 10-15, 2005, Samsun, Turkey (publ. on CD)
- [6] F. A. Marvasti. A unified approach to zero-crossings and nonuniform sampling // Department of Electrical Engineering, Illinois Insitute of Technology, Chicago, USA, 1987.
- [7] M. Greitans. Spectral analysis based on signal dependent transformation // The 2005 International Workshop on Spectral Methods and Multirate Signal Processing, (SMMSPP 2005), June 20-22, 2005, Riga, Latvia. pp. 179-184.
- [8] I.Bilinski , Yu. F. Borovik. and A.Mikelson (1982), “Use of rectangular periodic functions for computing discrete Fourier transforms”, Autom. Control Comput. Sci., 2 81-86 (in Russian)
- [9] I.Bilinski, Yu. F.Borovik and A.Mikelson (1982), “Implementation of a DFT method based on the use of rectangular functions”, Autom. Control Comput. Sci., 6 81-88 (in Russian).
- [10] Michel Mouly, Marie – Bernadette Pautet (1992), The GSM system for mobile communications.
- [11] E. Allier, G. Sicard, L.F. and Renaudin, M (2003). A new class of asynchronous a/d converters based on time quantization. In *Proc. Of International Symposium on Asynchronous Circuits ans Systems ASYNC'03*, pages 196-205, Vancouver, Canada.
- [12] J. W.Mark, T.D.Todd, (1981). A nonuniform sampling approach to data compression. *IEEE Trans. On Comm.*, 29(1):24-32.
- [13] W. Kaster, Analog-Digital conversion, *Analog Devices, Inc.*, 2004.
- [14] E. Beķeris Signālu teorijas elementi. – Rīga: IU ‘‘Mācību apgāds’’, 1998. – 189. lpp.
- [15] M.Unser, A.Aldroubi, M.Eden B-Spline Signal Processing: Part I-Theory// IEEE Transactions on Signal Processing. – 1993. – Vol. 41. – pp. 821-833
- [16] T.Werther "Reconstruction from irregular samples with improved locality", Master's thesis, University of Vienna, Dec. 1999.
- [17] C. Cenker, Hans G. Feichtinger, Hermann M. Iterative algorithms in irregular sampling A first comparison of methods. – 1992.
- [18] A. Baums, Energy Consumption Optimization in Hard Real-Time System CMOS Processors, *Electronica and Electrical Engineering* , N4 (68) , Kaunas 2006, pp.19-22.
- [19] HT80C51 User Manual, Philips Electronics N.V., 2005, pp. 1-133.
- [20] Furber S.B. , Day P., GarsideJ.D, Amulet1: A Micropiplined ARM, GompCon 94, Sun Francisco, March 1994, pp.290-299.
- [21] Furber S.B. , GarsideJ.D, Temple S., Day P., Amulet2e: An Asyncheonous Embedded Controller, Proc. IEEE. 87(2), Febr, 1999, pp. 290-299.
- [22] Furber S.B. , Edvards D.A. , GarsideJ.D., Amulet3: Asyncheonous EmbeddedProcessor, Proc. Of the IEEE Intern.Conf on Computer Design, 2000.
- [23] www.handshakesolutions.com/Products_Services/ARM996HS/ - 20k..
- [24] M. Akay, Ed., Time-frequency and wavelets in biomedical signal processing, *IEEE*

Press, 1998.

- [25] Л.В. Новиков, Основы вейвлет анализа сигналов. Учебное пособие.
<http://neuroschool.narod.ru/>.
- [26] M. Greitans, Signal dependent STFT-like Time-frequency analysis applicable for non-uniform sampling. *Submitted to the journal “Sampling theory in signal and image processing”* (preprint)
- [27] E. Wigner, On the quantum correction for thermodynamics equilibrium. *Phys. Rev.*, vol. 40, pp.749-759, 1932

1. PIELIKUMS. PUBLIKĀCIJA ŽURNĀLĀ „ELECTRONICS AND ELECTRICAL ENGINEERING”,
ISSN 1392-1215, NO. 4 (68), 2006, M. GREITANS, I. HOMJAKOVS, ENHANCED DIGITAL
SIGNAL PROCESSING OF SIGNAL-DEPENDENTLY SAMPLED SIGNALS.

Enhanced Digital Signal Processing of Signal-dependently Sampled Data

M. Greitāns, I. Homjakovs

*Institute of Electronics and Computer Science, University of Latvia,
14 Dzerbenes str., Riga, LV1006, Latvia, phone: +371-7558203, e-mail: modris_greitans@edi.lv*

Introduction

The evolution of electronic circuits has been drastically advanced during the recent years. Introduction of new nano-materials and technologies allows to miniaturize devices considerably. As a result, engineers are developing much faster, more complex and smarter systems. This progress makes them attractive for expanded range of applications. A System-on-Chip (SoC) technology is considered as a revolutionary approach in electronics. It means the packaging of all the necessary electronic circuits and parts for a "system" on a single integrated circuit [1]. For example, a system-on-chip for a sound-detecting device might include an audio receiver, an analog-to-digital converter (ADC), a digital signal processor, necessary memory as well as the input/output logic control for a user. Consequently, SoC may contain digital, analog and often mixed-signal functions – all on one chip.

Most of today's electronics systems are based on synchronous paradigm - the work of a whole system is ruled by a clock. All the states in a design have to be changed at the same time. Since the systems constantly become more sophisticated and capacious, it is established that 10-30% of their resources (size, power consumption etc.) are wasted to circuits, which provide the synchronism of the device. Thus, it is relevant to explore other paradigms for development of future microelectronics based on alternative and innovative solutions.

One of the alternative approaches, which was introduced in the mid 50s and now receives increasing interest, is asynchronous logic. The key benefits of the asynchronous system are lower power consumption, absence of the clock screw, reduced heat elimination, lower EMI, automatic adaptation to physical properties, etc. [2]. Asynchronous design is characterized by the absence of a global clock, instead of that the system is driven by events, which arise during the execution of actual task. Such a change of paradigm leads to a drastic rearrangement in electronic designs by complete rethinking their architecture and signal processing methods.

Typically the information obtained from sensors has continuous time nature, while modern signal processing

techniques are based on digital methods. The conversion from analog (continuous) to digital (discrete) presentation of the source data is being performed by ADC. In systems designed according to synchronous paradigm, ADC is driven by system clock. If uniform sampling is used the ADC clock frequency is determined by Nyquist theorem. If non-uniform sampling is involved, the mean sampling density can be below the Nyquist rate, however the sampling point flow is still derived from the clock. The asynchronous systems do not have a clock at all. That leads to the question - how to manage the analog-to-digital conversion in a case without clock? The asynchronous circuits on its merits are event-driven, so also the signal sampling process should be organized by events, which are got from the information presented in signal. New class of ADCs based on signal-dependent sampling schemes has to be developed.

The three most popular types of signal-dependent sampling – zero-crossing [3], reference signal crossing [4] and level-crossing [5] are illustrated in Fig. 1. Naturally, each of them has its own advantages and limitations, however there are some important joint features, which have to be taken into account performing the processing of signal-dependently sampled data:

1. in general case the signal samples are spaced non-uniformly,
2. it is impossible to determine the sampling time instants in advance,
3. local sampling density depends on local statistical characteristics of signal.

In this paper the samples obtained by level-crossing approach are chosen as an example of input data for digital signal processing. However, the developed methods are useful also for other cases of signal-dependently sampled data.

Non-uniformly spaced data

The techniques of non-uniform sampling are discussed in literature for many years [6]. In most cases the problem is stated from the following point of view - how to calculate the sampling time instants in a way, which allows

to gain some advantages during the processing. The suppression of frequency aliasing effect can be quoted as an example. In this case the sampling point flow is deliberately pseudo-randomized, and it has to satisfy certain requirements. The statistical characteristics of such a sampling process are independent of input signal. In electronic design that leads to the task to clock the “classical” ADC at predetermined, non-uniformly spaced time instants with high accuracy.

The signal-dependent sampling also provides signal samples at the non-uniformly spaced time moments. Their nature and properties can considerably differ from deliberately pseudo-randomized case. Let us discuss, as an example, the level-crossing sampling approach. Higher frequency in a spectral presentation of signal provides faster changes in its waveform. If sampling is organized as events of levels crossing that leads to the higher density of signal samples (see Fig.1.c). Statistical characteristics of obtained sampling point flow directly depend on input signal. Furthermore, if a signal waveform has some regularity, the event flow of level-crossing sampling has the same regularity as well. As a result of this effect, the following problem can be stated - the methods developed for processing of deliberately non-uniformly sampled data can be impracticable for signal-dependently non-uniformly sampled data. This situation will be illustrated in the next section.

General Discrete Fourier transform

The classical method for digital signal analysis is a pair of Discrete Fourier Transform (DFT). The direct DFT allows to obtain the spectral values from N signal samples if they are equidistantly spaced in time:

$$X(m) = \sum_{n=0}^{N-1} x(n) \exp(-j2\pi mn / N). \quad (1)$$

The inverse DFT allows to reconstruct signal values in time from signal spectral values at equally spaced frequencies:

$$x(n) = \frac{1}{N} \sum_{m=0}^{N-1} X(m) \exp(j2\pi mn / N). \quad (2)$$

In practical applications typically exist two restrictions – the bandwidth of signal spectrum is limited to some higher frequency Ω , and the observation of signal waveform is limited to some duration Θ . The discrete signal samples $x_n = x(t_n)$, $n = \overline{0, N-1}$ are obtained by sampling procedure and they can be located arbitrary along the time axis. The general form of DFT, that enables the processing of non-equidistantly spaced data, can be expressed as:

$$X_m = \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi \frac{m}{\Theta} t_n\right). \quad (3)$$

The band-limited nature of the signal provides the restriction for m : $\left|\frac{m}{\Theta}\right| \leq \Omega$.

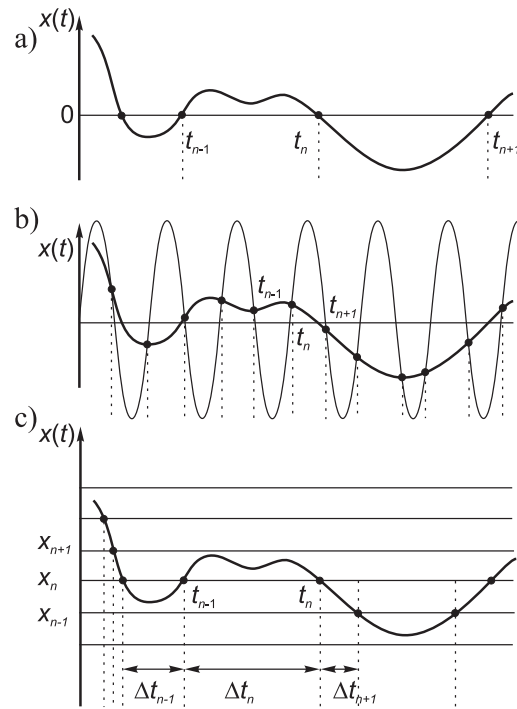


Fig. 1. Signal-dependent sampling schemes: a) zero-crossing; b) reference signal crossing; c) level-crossing

In the uniform sampling case expression (3) conforms to the formula (1). If appropriate deliberately non-uniform sampling scheme is used, expression (3) provides quite good spectral estimation for the cases, where sampling density is equal or higher than the Nyquist rate. Moreover, it is possible satisfactorily to estimate the spectral content of signal even if sampling density is below the Nyquist rate. The sampling point flow has to suppress the frequency aliasing for that. However, if formula (3) is applied to the data acquired by level-crossing sampling, the obtained result not always is adequate, despite the fact that signal always is oversampled. Let us illustrate this statement by a simple example. The mono-harmonic signal with frequency f_0 is sampled by 3-bit level-crossing ADC (7 levels). The spectral estimate obtained by formula (3) is shown in the Fig. 2. In addition to the spectral component at the true frequency (~ 0.0955 Hz) spurious components at the higher odd harmonics appear as well. Note, the issue of spurious components is not due to the frequency aliasing effect connected with insufficient sampling density, but due to specifics of signal and non-uniform samples obtained by level-crossing approach [5]. It leads to the conclusion that specialized enhancement of DFT should be derived to process the data acquired in such a way.

On the base of the Fourier series we can reconstruct the signal waveform from its spectral estimates by the following formula

$$\hat{x}(t) = \sum_{m=-M}^M X_m \exp(j2\pi t f_m), \quad t \in [0, \Theta], \quad (4)$$

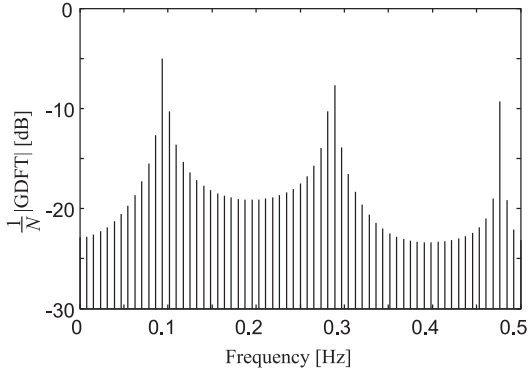


Fig. 2. Spectral estimate of sinusoidal signal with $f_0 = 12.22 / \Theta$ ($\Theta = 128$ s.) sampled by 7 levels-crossings

where $f_m = \frac{m}{\Theta}$. Using the information about original signal $x(t)$, we can construct a reconstruction error as $\varepsilon(t) = x(t) - \hat{x}(t)$. The initial part of it is illustrated in the Fig. 3. The values $\varepsilon_n = \varepsilon(t_n)$ of reconstruction error at the sampling point instants t_n are shown by bold points. We can observe that the reconstruction error between samples used to be considerable higher than ε_n values. If sampling is based on the classical scheme, where sampling points are determined in advance, there are no restriction to the behavior of reconstructed signal between samples. In the case of level-crossing sampling the situation differs. Each level crossing is defined as an event and characterized by signal sample. Between them the waveform of reconstructed signal should not cross any level. Consequently the reconstruction error has to be minimized not only at sampling time instants, but also between them with the same accuracy. The minimization task

$$\int_0^{\Theta} \varepsilon^2(t) dt \rightarrow \min \quad (5)$$

can be defined on the understanding that the signal values are known only at sampling points and the reconstructed signal is described by the expression (4). The minimization problem has to be solved with respect to spectral estimates $\{X_m\}$.

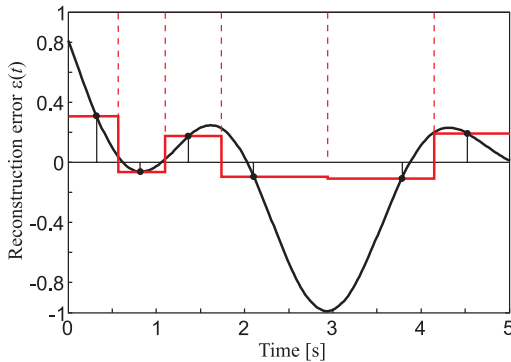


Fig. 3. Reconstruction error function (black curve) and zero-order interpolation (red piece-wise constant line) of ε_n (bold dots)

The solution of (5) can be based on two different approaches:

- 1) signal samples $\{x_n\}$ are interpolated within the time interval $[0 \Theta]$ and the reconstruction error is expressed as $\varepsilon^{(x)}(t) = \tilde{x}(t) - \sum_m X_m \exp(j2\pi f_m t)$, where $\tilde{x}(t)$ is interpolated signal;
- 2) error samples $\varepsilon_n = x_n - \hat{x}_n$ are interpolated within the time interval $[0 \Theta]$, and $\hat{x}_n = \sum_m X_m \exp(j2\pi f_m t_n)$.

Enhanced DFT based on signal interpolation.

Signal interpolation easily can be done by connecting the sample points with polynomials $p_n^k(t)$ of order k , or a band-limited interpolation can be performed as a sum of time-shifted sinc functions.

Let us rewrite (5) taking into account that signal samples are interpolated:

$$\int_0^{\Theta} \left(\tilde{x}(t) - \sum_{m=-M}^M X_m^{(x)} \exp(j2\pi f_m t) \right)^2 dt \rightarrow \min. \quad (6)$$

To find the minimum, all the individual derivatives of X_m have to be considered as being equal to zero. We obtain $2M + 1$ linear equations

$$2 \int_0^{\Theta} \left(\tilde{x}(t) - \sum_{k=-M}^M X_k^{(x)} \exp(j2\pi f_k t) \right) \cdot \exp(j2\pi f_m t) dt = 0 \quad (7)$$

for $m = -M, M$. Taking into account that $f_m = \frac{m}{\Theta}$, the $\{\exp(j2\pi f_m t)\}$ is a set of orthogonal functions into interval $[0 \Theta]$. Thus we can write:

$$X_m^{(x)} = \frac{1}{\Theta} \int_0^{\Theta} \tilde{x}(t) \exp(j2\pi f_m t) dt. \quad (8)$$

The expression (8) is similar with the formula of calculation of Fourier series coefficients for signal $\tilde{x}(t)$.

If signal samples $\{x_n\}$ are interpolated with zero-order polynomials (like a piece-wise constant line in the Fig. 3), we obtain

$$\begin{aligned} X_m^{(x_0)} &= \sum_{n=0}^{N-1} x_n \int_{\frac{t_n+t_{n-1}}{2}}^{\frac{t_n+t_{n+1}}{2}} \exp(j2\pi f_m t) dt \\ &= \frac{j}{2\pi f_m} \sum_{n=0}^{N-1} x_n \exp(j2\pi f_m t_n) (1 - \exp(-j2\pi f_m \Delta t')), \end{aligned} \quad (9)$$

where $\Delta t'_n = \frac{t_{n+1} - t_{n-1}}{2}$.

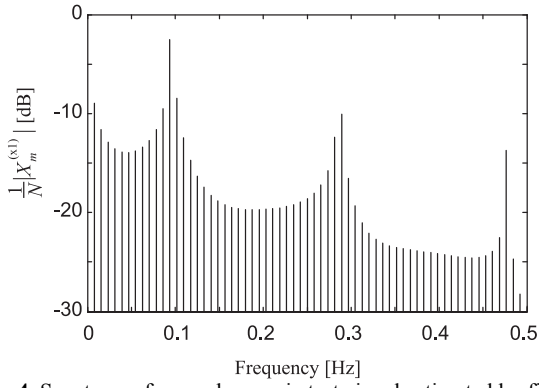


Fig. 4. Spectrum of mono-harmonic test-signal estimated by first-order signal interpolation algorithm

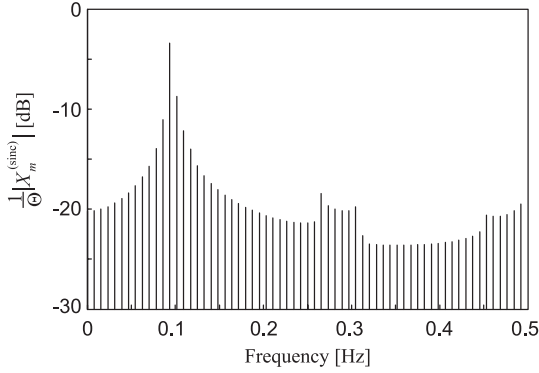


Fig. 5. Spectrum of mono-harmonic test-signal estimated by sinc interpolation algorithm

For piece-wise linear interpolation between the samples we can use polynomial $p_n^1 = \alpha_n(t - t_n) + x_n$, where $\alpha_n = \frac{\Delta x_n}{\Delta t_n}$, $\Delta x_n = x_n - x_{n-1}$, $\Delta t_n = t_n - t_{n-1}$, which gives:

$$X_m^{(x1)} = X_m^{(x0)} + \frac{1}{(2\pi f_m)^2} \cdot \sum_{n=0}^{N-1} \alpha_n \exp(j2\pi f_m t_n) (1 - \exp(-j2\pi f_m \Delta t_n)) + \frac{i}{2\pi f_m} \sum_{n=0}^{N-1} \alpha_n \Delta t_n \exp(j2\pi f_m t_n) \exp(-j2\pi f_m \Delta t_n). \quad (10)$$

Another approach is based on interpolation of $\{x_n\}$ with sinc functions. As the bandwidth of signal is limited to Ω , we can write:

$$\tilde{x}^{(\text{sinc})}(t) = \sum_{k=0}^{K-1} c_k \text{sinc}(2\Omega(t - kT)), \quad (11)$$

where $T = \frac{1}{2\Omega}$, $K : kT < \Theta$. Amplitudes c_k can be found from a linear equation system:

$$x_n = \tilde{x}^{(\text{sinc})}(t_n). \quad (12)$$

In this case the enhanced discrete Fourier transform

becomes

$$X_m^{(\text{sinc})} = \sum_{k=0}^{K-1} c_k \exp(-j2\pi f_m kT). \quad (13)$$

Note, the last method, besides the DFT complexity of calculations, requires the solution of linear system with N equations and with $2M + 1$ unknowns.

Enhancement of Discrete Fourier transform based on interpolation of error samples.

Like the interpolation of signal samples the continues time reconstruction error function $\tilde{\varepsilon}(t)$ can be obtained from its values $\varepsilon_n = x_n - \hat{x}_n$ at time instants $\{t_n\}$. The estimates of reconstructed signal are calculated as $\hat{x}_n = \sum_m X_m \exp(j2\pi f_m t_n)$. The problem (5) in this case can be interpreted as minimization of area under the function $\tilde{\varepsilon}^2(t)$. If ε_n^2 is interpolated by zero-order polynomial the minimization task becomes as:

$$\sum_{n=0}^{N-1} \left(x_n - \sum_{m=-M}^M X_m^{(\varepsilon 0)} \exp(j2\pi f_m t_n) \right)^2 \cdot \Delta t_n \rightarrow \min. \quad (14)$$

After the derivation the system of linear equations is formed

$$2 \sum_{n=1}^N \left(x_n - \sum_{k=-M}^M X_k^{(\varepsilon 0)} \exp(j2\pi f_k t_n) \right) \exp(j2\pi f_m t_n) \times \Delta t_n = 0 \quad (15)$$

This system rewritten in matrix form is

$$\Psi \mathbf{x} = \mathbf{X}^{(\varepsilon 0)} \cdot \Phi \cdot \Psi^T, \quad (16)$$

where $\varphi_{mn} = \exp(j2\pi f_m t_n)$ and $\psi_{mn} = \varphi_{mn} \cdot \Delta t_n$.

Solution of (16) can be found as

$$\mathbf{X}^{(\varepsilon 0)} = (\Psi \cdot \mathbf{x}) \cdot (\Phi \cdot \Psi^T)^{-1}, \quad (17)$$

where $(\cdot)^T$ and $(\cdot)^{-1}$ denotes the transpose and inverse operation of matrix respectively.

The first-order polynomial interpolation of error function samples provides the following minimization task:

$$\frac{1}{2} \left(\sum_{n=0}^{N-2} \varepsilon_n^2 \cdot \Delta t_n + \sum_{n=1}^{N-1} \varepsilon_n^2 \cdot \Delta t_{n-1} \right) \rightarrow \min. \quad (18)$$

It looks like a sum of two zero-order interpolation problems. The solution in this case is similar to the previous one expressed as (17):

$$\mathbf{X}^{(\varepsilon 1)} = (\Psi' \cdot \mathbf{x}' + \Psi'' \cdot \mathbf{x}'') \left(\Phi' \cdot \Psi'^T + \Phi'' \cdot \Psi''^T \right)^{-1}, \quad (19)$$

where $\Phi', \Psi', \mathbf{x}'$ and $\Phi'', \Psi'', \mathbf{x}''$ matrices are formed from Φ, Ψ, \mathbf{x} by using indexes $n = \overline{0, N-2}$ and $n = \overline{1, N-1}$ respectively.

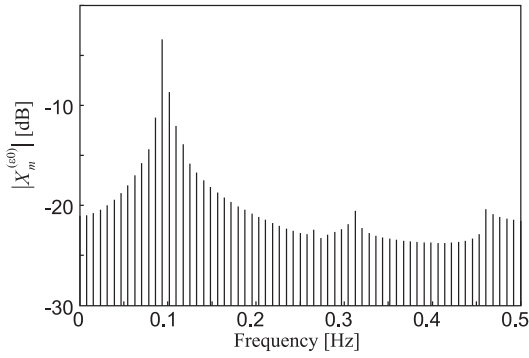


Fig. 6. Spectrum of mono-harmonic test-signal estimated by zero-order reconstruction error interpolation algorithm

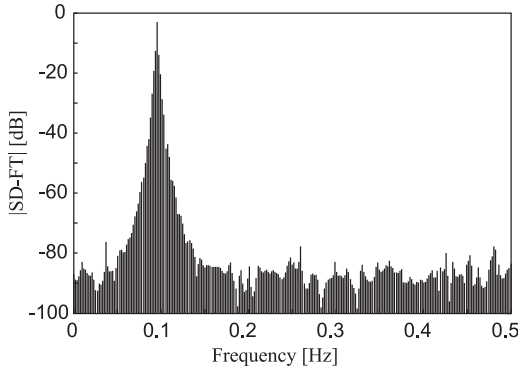


Fig. 7. Spectrum of mono-harmonic test-signal estimated by signal-dependent transformation

Simulation results

The performance of described algorithms has been investigated by computer simulations. The above mentioned single-tone sinusoid has been used as a test-signal. The input data are acquired from $\Theta = 128[\text{sec}]$ long observation as events of 7 levels-crossings. The frequency of test-signal is $12.22/\Theta \approx 0.0955[\text{Hz}]$ that does not lie on the frequency grid of Fourier series. The number of acquired samples is $N = 172$.

The spectral estimate obtained by General DFT (expression (3)) and illustrated in Fig. 2 can be used as a reference, which allows identifying the improvement of processing, when enhanced methods are exploited. The Fig. 4 shows spectral result obtained by first-order signal interpolation algorithm (expression (10)). Although the amplitudes of higher spurious harmonics are decreased for about 5dB, they are still considerable. The complexity of this algorithm is few times higher than DFT. The zero-order signal interpolation gives slightly worse result.

The band-limited interpolation with sinc functions (expression (13)) provides improved spectral presentation of test signal, which is shown in the Fig. 5. The spurious harmonics are suppressed. The drawback of this algorithm is that beside the DFT calculation it requires the solution of the linear equation system as well.

The similar mathematical complexity is also for methods, which use reconstruction error interpolation by

zero- or first-order polynomials. The results obtained in both cases are quite similar, and therefore we present here only the simplest case – zero-order error interpolation, in the Fig. 6.

The common feature of all presented algorithms is spectral analysis on the grid of Fourier frequencies $f_m = m/\Theta$. For the expressions (9) and (10) the motivation is to build the orthogonal basis of transformation. The expressions (17) and (19) can be considered as unorthogonal transformations. That allows to use also frequency grid with higher density, which can improve spectral resolution. However, the necessity to solve the equations system limits the grid – the number of analysis frequencies has to be equal or less than the number of samples. Note, the number of samples in signal-dependent sampling case is not known in advance, because it depends on actual signal properties. The method which overcomes this problem is so called signal-dependent Fourier transform described in [7]. The frequency grid of this algorithm is independent on number of samples and provides high spectral resolution. The Fig. 7 demonstrates the spectral estimate obtained by this method, where analysis grid is four times frequenter than in the DFT case.

The quality of simulated algorithms can be characterized by integral value of squared reconstruction error. The estimated values, averaged over different signal phases, are summarized in the Table 1. It shows that all methods decrease the reconstruction error in comparison with GDFT. The obtained improvement can vary from insignificant to several thousand times.

Conclusions

One of the basic tools in digital signal processing is the Discrete Fourier Transform. In this paper we presented several enhancements of DFT to make it convenient for analysis of signal-dependently sampled data, particularly paying attention to the level-crossing sampling. In this case the samples are not only spaced non-uniformly in time, but also its distribution depends on signal properties. The periodical signal provides regularities into sampling point flow, which leads to the appearance of spurious components (harmonics) in the spectrum estimated on the bases of conventional DFT analysis.

The level-crossing sampling technique determines not only signal value at sampling time instance, but also the rule, that signal values between two samples should not cross any quantization level. This condition can be taken into account during development of processing methods. The proposed idea is to minimize the error between original and reconstructed by Fourier series signals not only at sampling time instants, but also between them with the same accuracy. The problem lies in the fact that we know original signal values only at sampling instants. Two different approaches are considered. The first one is based on obtaining continues time signal by interpolation of known signal samples. The expressions for enhanced DFT calculation are proposed for zero-order and first order polynomial interpolation as well as for band-limited interpolation with sinc functions. The second approach to minimize continues time reconstruction error is based on

Table 1. Estimated reconstruction errors for different methods

Method	GDFT	Zero-order x_n interpolation	First-order x_n interpolation	x_n interpolation with sinc	Zero-order ε_n interpolation	First-order ε_n interpolation	Signal-dependent DFT-like method
$\int_0^{\Theta} \varepsilon^2(t) dt$	835	811	803	3.00	2.35	2.29	0.07

interpolation of error samples. The expressions for calculation of spectral coefficients are proposed for zero-order and first-order polynomial interpolation. Simulation results show the improvement of data processing if enhanced algorithms are used instead of classical General DFT. It should be mentioned that better results are achieved by algorithms, which require performing more mathematical operations. The best result can be reached by the signal-dependent DFT-like method, however the high computing complexity makes it unpractical for real-time applications.

References

1. **Proceedings** of the 5th IEEE International Workshop on System-on-Chip for Real-Time Applications (IWSOC 2005), 20-24 July 2005., Banff, Alberta, Canada. IEEE Computer Society 2005, ISBN 0-7695-2403-6.
2. **Hauck S.** Asynchronous Design Methodologies: An Overview // Proceedings of the IEEE, Vol. 83, No. 1, pp. 69-93, January, 1995.
3. **Logan B.** Information in zero crossings of bandpass signals

// Bell Sys. Tech. J. – 1977. – 56.– P. 487–510.

4. **Nazario MA, Saloma C.** Signal recovery in sinusoid-crossing sampling by use of the minimum-negativity constraint // Applied Optics. – 1988. – 37(14). – P. 2953–2963.
5. **Aeschlimann F., Allier E., Fesquet L., Renaudin M.** Spectral analysis of level-crossing sampling scheme // Proc. of the 2005 International Conference on Sampling Theory and Application (SampTA 2005).– Samsun, Turkey (publ. on CD), July 10-15, 2005.
6. **Marvasti F. A.** A unified approach to zero-crossings and nonuniform sampling / Department of Electrical Engineering, Illinois Institute of Technology, Chicago, USA, 1987.
7. **Greitans M.** Spectral analysis based on signal dependent transformation // The 2005 International Workshop on Spectral Methods and Multirate Signal Processing, (SMMSP 2005).– Riga, Latvia, June 20-22, 2005.– P. 179–184.

Presented for publication 2006 02 10

M. Greitans, I. Homjakovs. Enhanced Digital Signal Processing of Signal-Dependently Sampled Signals // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 4(68).– P. 9–14.

Asynchronous circuits receive now an increasing interest. Their promising advantages can play a significant role in future electronics' development. One of the factors, which delay the wide spread of the clock-less systems, is incompatibility with classical clock-driven analog to digital converters and processing algorithms. The new ADC approaches, which are based on signal-dependent sampling, are required, as well as convenient processing methods have to be developed. The paper presents the enhanced DFT-like algorithms derived from the idea to minimize the signal reconstruction error not only at sampling points, but also between them with the same accuracy. Two approaches for creating continuous time error function from its discrete values are investigated: 1) interpolation of signal samples; 2) interpolation of error samples. Level-crossing sampling has been used as an example for signal digitizing. Achieved advantages are demonstrated by simulations. Il.7, bibl. 7 (in English; summaries in English, Russian and Lithuanian).

М. Грейтанс, И. Хомяковс. Усовершенствованная цифровая обработка дискретизованных по уровню сигналов // Электроника и электротехника. – Каунас: Технология, 2006. – № 4(68). – С. 9–14.

В настоящее время все больше внимания уделяется асинхронным системам. Однако для асинхронных систем обработки сигналов существует потребность в новых видах аналого-цифрового преобразования и соответствующих им алгоритмах обработки. Рассматривается подход к аналого-цифровому преобразованию сигналов, основанный на их дискретизации по уровню, и предлагаются усовершенствованные, ДФТ-подобные алгоритмы, позволяющие значительно уменьшить погрешность результатов цифрового анализа сигналов. Ил.7, библи. 7 (на английском языке; рефераты на английском, русском и литовском яз.).

M. Greitans, I. Homjakovs. Patobulintas adaptyviai diskretizuotų signalų skaitmeninis apdorojimas // Elektronika ir Elektrotechnika.– Kaunas: Technologija, 2006. – Nr. 4(68). – P. 9–14.

Vis didesnį susidomėjimą kelia asinchroniniai grandynai. Dėl daug žadančių privalumų jie gali vaidinti reikšmingą vaidmenį ateityje, tobulinant elektroniką. Vienas iš veiksnių, lėtinančių platų sistemų, nenaudojančių taktinio dažnio, paplitimą, yra jų nesuderinamumas su klasikiniiais taktinio signalo valdomais analoginiais-kodiniais keitikliais ir apdorojimo algoritmais. Reikalingi naujų tipų keitikliai, kurių veikimas būtų grindžiamas adaptyviu (nuo signalo priklausančiu) diskretizavimu, taip pat turi būti sukurti patogūs apdorojimo metodai. Pateikiami patobulinti DFT algoritmai, kurie sukurti remiantis idėja minimizuoti signalo atkūrimo paklaidą ne tik diskretizavimo taškuose, bet ir tarp jų, tuo pačiu tikslumu. Nagrinėti du metodai, skirti tolygiai laikinei klaidos funkcijai sukurti iš jos diskretinių verčių: 1) diskretinių signalo verčių interpoliavimas; 2) diskretinių klaidos verčių interpoliavimas. Lygio kirtimo diskretizavimo būdas naudotas signalui skaitmenizuoti. Pasiiekti patobulinimai demonstruojami naudojant modelius. Il. 7, bibl. 7 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

2. PIELIKUMS. PUBLIKĀCIJA ŽURNĀLĀ „ELECTRONICS AND ELECTRICAL ENGINEERING”,
ISSN 1392-1215, NO. 4 (68), 2006, A. BAUMS, ENERGY CONSUMPTION OPTIMISATION IN
HARD REAL-TIME SYSTEM CMOS PROCESSORS.

Energy Consumption Optimization in Hard Real-Time System CMOS Processors

A. Baums

Institute of Electronics and Computer Science,

14 Dzerbenes st. LV-1006 Riga, Latvia; tel. +371 7558134, fax: +371 755337; e-mail: baum@edi.lv

Introduction

Portable system energy consumption reduction is of primary importance in mobile embedded systems and SoC design [1]. Dynamic voltage scaling (DVS) for CMOS chips at this time is the most effective energy reduction technology [2–11]. Usually there are no or few problems for DVS using it for *soft real-time* system design or task scheduling, where deadlines can be missed if the Quality of Service is kept [9, 10]. In *hard real-time* systems it is necessary to provide that the task never miss the deadline and the energy saving is maximized. The problems arise in *on-line* systems when at the ordinary task starting moment the task execution time is impossible to know and the best energy reduction regime can not be selected. In majority investigations and publications periodic task executions are analysed there with *off-line* determined execution times and determined priorities. Then the popular hard real-time task scheduling methods are earliest–deadlines first (EDF) or rate-monotonic scheduling (RMS) for DVS based energy consumption reduction is used [5, 6, 7, 8]. Few investigations are made in the field of event driven single stochastic hard real-time task energy consumption reduction. In [12] there is a method of dividing the task into multiple smaller sub-tasks with checkpoints for dynamic power management or voltage scaling. By using this method the energy consumption is reduced, but there are not guarantee that for the last sub-tasks execution step the deadline will not be reached. Method for the hard task energy consumption *on-line* minimisation which is named “*mini-max*” is proposed from author in article [11]. By using it the deadline is guaranteed for the task execution at the minimal power till the critical time t_{cr} , after which the execution is finished at maximal clock frequency.

Dynamic voltage scaling DVS

Power consumption of CMOS has two main components: P_{sw} – dynamic power consumption by charging and discharging the capacitive load and P_{st} – static power consumed, which is determined by leakage current I_{leak} and for CMOS technology below 0.1 μm can

not be taken into account. Dynamic or switching power can be estimated by: $P_{sw} \sim C_{ef} V^2 f_{sw} = C_{ef} V^2 / \tau_c$, where: V – supply voltage, C_{ef} – capacitive load, f_{sw} – switching frequency and τ_c – cycle time.

The time and energy consumption optimisation by using dynamic voltage-time scaling DVS is based on power ΔP_i consumption minimization for task step i , duration the *slack time* s_i . In cases when the real task execution time t_i is smaller then the worst case execution time WCET $_i$:

$$s_i = \text{WCET}_i - t_i. \quad (1)$$

For hard real-time system: $\text{WCET}_i \leq D$ (D deadline)

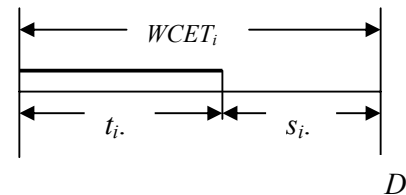


Fig. 1. Time connections for the task step i

For voltage or (and) frequency scaling different program controlled CMOS chips are designed and can be used: *Intel XScale Technology* (XA250, 80200) [14], *Transmeta Co. Crusoe* (TM5400÷TM5800) [15], *XILINX Co. Virtex-Pro* [16], *Altera Co. NoisII* [17].

Mini-max method for hard real-time task energy consumption on-line minimization

The hard real-time single event driven stochastic task energy consumption on-line minimization method is developed by consideration that CMOS processor has a wide supply voltage and clock frequency borders for voltage and frequency scaling. For the stochastic task it is necessary to assess the clock cycle τ_c number N_{max} at the WCET and at the best case execution time (BCET) τ_c number N_{min} . It can be made by simulation or executing

the task at some nominal frequency f_0 , clock cycle τ_{co} and nominal voltage V_0 . By using the power cycle time characteristic $P_c(\tau_c)$ the cycle time can be adjusted in region $\tau_{cmin} \div \tau_{cmax}$ and the cycle power consumption in region $P_{cmax} \div P_{cmin}$ (Fig. 2.), where P_{co} – cycle time power consumption at nominal frequency f_0 and nominal voltage V_0 .

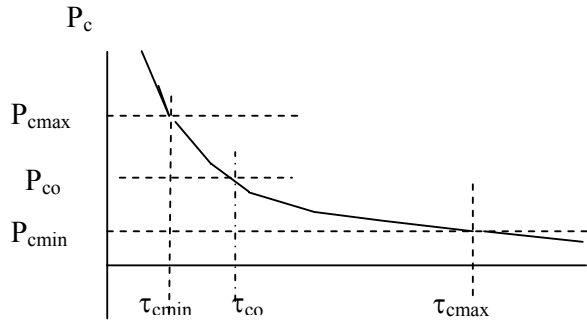


Fig. 2. Cycle time power consumption

Execution time characteristics at nominal voltage V_c and frequency f_{co} :

$$t_{max} = \tau_{co} N_{max} = WCET, \quad (2)$$

$$t_{min} = \tau_{co} N_{min} = BCET, \quad (3)$$

$$s_{max} = D - \tau_{co} N_{min}. \quad (4)$$

By using mini-max method in the worst case, when $t_i = WCET_i = D$ and $s_i = 0$, the task execution is started at minimal power P_{cmin} and continued till instant t_{cr} , after which in time interval $T_D = D - t_{cr}$ the execution is finished at maximal frequency and min clock time τ_{cmin} (Fig. 3).

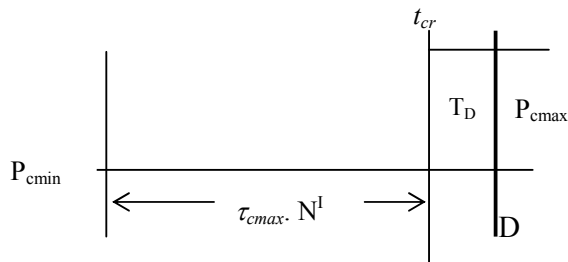


Fig. 3. The presentation of task execution critical time t_{cr}

In Fig. 3 by N^l the number of instructions executed at maximal long clock cycles is denoted. This number and time intervals t_{cr} and T_D can be calculated from equation (5):

$$\tau_{cmax} \cdot N^l + \tau_{cmin} (N_{max} - N^l) = D, \quad (5)$$

$$N^l = (D - \tau_{cmin} N_{max}) / (\tau_{cmax} - \tau_{cmin}), \quad (6)$$

$$t_{cr} = \tau_{cmax} N^l, \quad (7)$$

$$T_D = D - \tau_{cmax} \cdot N^l. \quad (8)$$

From here the energy consumption at WCET:

$$E_{max} = \tau_{cmax} P_{cmin} N^l + P_{cmax} T_D. \quad (9)$$

The task energy consumption E_{ci} at the dynamic (on-line) scheduling when $\tau_{cmax} N_i < D - T_D$:

$$E_{ci} = \tau_{cmax} P_{cmin} N_i, \quad (10)$$

$$E_{ci} = \tau_{cmax} P_{cmin} N^l + \tau_{cmin} P_{cmax} (N_i - N^l), \quad (11)$$

or

$$E_{ci} = \tau_{cmin} P_{cmax} N^l + (\tau_{cmax} P_{cmin} - \tau_{cmin} P_{cmax}) N_{max} (\tau_{cmax} - \tau_{cmin}) / (\tau_{co} - \tau_{cmin}). \quad (12)$$

To estimate the benefits of the mini-max method it is necessary to compare this energy consumption with the energy consumption at task static (off-line) scheduling, i.e. that at every instant of task start its execution time t_i is known (for example calculate at task compilation). For different hard deadline situations the execution at different t_i can be estimated by using simple expressions represented in Fig. 4.

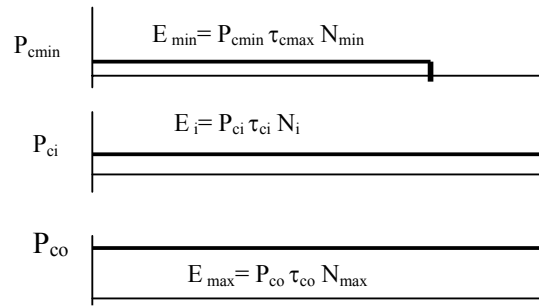


Fig. 4. Energy consumption at the task static scheduling

Uniform random distributed task execution time flow is proposed for the mini-max method investigation. For simplicity uniform distributed by constant step Δ_N clock cycle discrete number N_i and corresponding constant step Δ_s slot time s_{ico} values are used.

The single task energy consumption E_{m-m} includes two components E_1 and E_2 for the task execution times $t_i = t_{cr} \leq \tau_{cmax} N^l$:

$$E_1 = \sum_{i=1}^k P_{cmin} \tau_{cmax}, \quad (13)$$

else

$$E_2 = (n-k) P_{cmin} \tau_{cmax} + P_{cmax} \tau_{cmin} \sum_{k+1}^n (N_{max} - N^l - (n-k-1) \Delta_N) \quad (14)$$

Mini-max method was proposed in [13] and quantitative estimation of an *Example* was made. Experimental obtained $P(\tau_c)$ characteristic was used and work point $\tau_{co} = 1 \mu s$ ($f_{co} = 1 \text{ MHz}$), $P_{co} = 6 \text{ nJ}$; $\tau_{cmax} = 3 \mu s$, $P_{cmin} = 0.5 \text{ nJ}$; $\tau_{cmin} = 0.4 \mu s$, $P_{cmax} = 14 \text{ nJ}$, $N_{max} = 1000$, $N_{min} = 100$ and deadline $D = 1000 \mu s$ are selected. When $WCET = D$ and $BCET t_{min} = 100 \mu s$, then by using (6), (7) and (8): $N^l = 230$, $t_{cr} = 690 \mu s$ and $T_D = 310 \mu s$. The energy consumption is calculated for $\Delta_N = 100$ and for all corresponding slot times $s_{ico} = \{0, 100, 200, \dots, 800,$

900 μs) and the results of estimation as $E_{i \text{ din}}$ mini-max are represented in Fig. 5. For comparison there are represented curved lines calculated using the same workload: for task execution at nominal clock frequency f_{co} ($E_{i \text{ co}}$ at τ_{co}) and for task execution at off-line voltage scaling ($E_{i \text{ st}}$ off-line DVS).

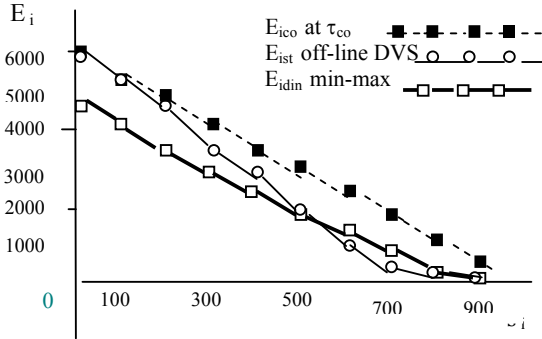


Fig. 5. The graphical presentation of the task energy consumption at different scheduling

The joint energy consumptions for task cycle including 10 tasks with different slot times $s_{i \text{ co}} = \{0, 100, 200, \dots, 800, 900 \mu\text{s}\}$ was calculated using (13) and (14): $E_{m-m} = E_1 + E_2 = 22000 \text{ nJ}$. For identical tasks and workloads the energy consumptions for the task execution at nominal clock frequency $E_{\text{co}} = 26000 \text{ nJ}$ and task execution at off-line voltage scaling, $E_{\text{st}} = 33000 \text{ nJ}$. The gain from the mini-max method using for energy consumptions in % can be estimated by using two characteristics

$$\delta_{\text{st}} = (E_{\text{st}} - E_{m-m}) 100 / E_{\text{st}}$$

and $\delta_{\text{co}} = (E_{\text{co}} - E_{m-m}) 100 / E_{\text{co}}$.

For the above example: $\delta_{\text{st}} = 15\%$ and $\delta_{\text{co}} = 33\%$.

The mini-max method benefit boundaries estimation

At first the influence of τ_{cmax} on E_{m-m} was investigated. The value of τ_{cmax} was adjusted from 1.5 μs to 5 μs and the energy economy results in % were compared with energy consumptions E_{st} and E_{co} . The results of estimation are presented in Table 1 and Fig. 6.

Table 1. Results of estimation

No.	τ_{cmax} , μs	δ_{st} , %	δ_{co} , %
1	1.5	3.5	13
2	2.0	8.8	22.5
3	2.5	12.5	29
4	2.8	16.5	34
5	3.0	15.5	33
6	3.5	12.5	31
7	4.0	6.0	25
8	4.5	-1.5	20
9	5.0	-19.0	6.0

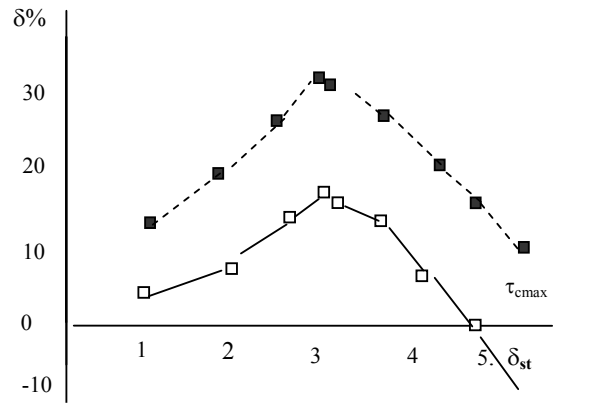


Fig. 6. Energy consumptions: $\delta_{\text{st}}(\tau_{\text{cmax}}) \%$ and $\delta_{\text{co}}(\tau_{\text{cmax}}) \%$

The value of $\tau_{\text{cmax}} = 3 \mu\text{s}$ which was used in Example is close to the optimal value $\tau_{\text{cmax}} = 2.8 \mu\text{s}$

At second the influence of τ_{cmin} on E_{m-m} was investigated. The value of τ_{cmin} was adjusted from 0.2 to 0.65 μs and the energy economy results in % compared with energy consumptions E_{st} and E_{co} . The results of estimation are presented in Table 2. and Fig. 7.

Table 2. Results of estimation

No.	τ_{cmin} , μs	δ_{st} , %	δ_{co} , %
1	0.2	31	75
2	0.3	24.5	40
3	0.4	15.5	33
4	0.45	13	31
5	0.5	10	28
6	0.6	-0.1	21
7	0.7	-4.0	20

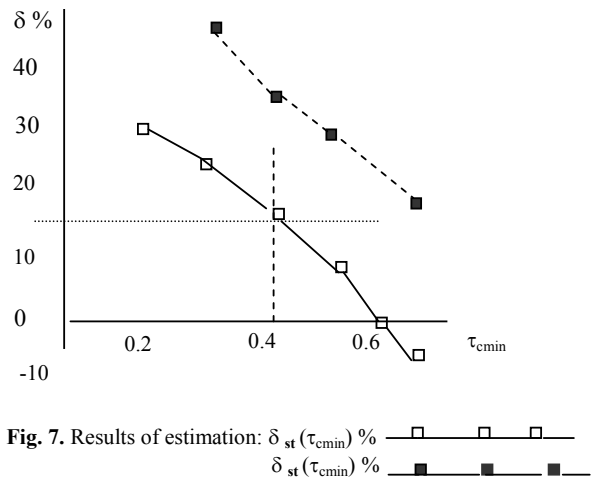


Fig. 7. Results of estimation: $\delta_{\text{st}}(\tau_{\text{cmin}}) \%$ and $\delta_{\text{co}}(\tau_{\text{cmin}}) \%$

It follows from Fig. 7, that by using mini-max method the energy economy can be a raised at task execution at small τ_{cmin} or by increasing the maximal clock frequency f_{cmax} .

Conclusion

To solve the *on-line* energy saving problem in the hard deadline task real-time systems *mini-max* method was proposed. By using this method at uniform random distributed task execution time workload, the energy saving exceed more then 15% compared to off-line DVS scheduling. The gain from *mini-max* using it for energy saving is estimated by using functions of τ_{cmax} and τ_{cmin} . The maximum energy saving can be obtained by selecting the optimal value of minimal frequency.

References

1. **Mudge T.** Power: a first-class architectural design constraint // Computer.– 2001.– 34(4).– P. 52–58.
2. **Yun H. S., Kim J.** On energy-optimal voltage scheduling for fixed-priority hard real-time systems - ACM Transactions on Embedded Computing Systems.– 2003.– V. 2.– P. 393–429.
3. **Ishihara T., Yasuura H** Voltage scheduling problem for dynamically variable voltage processors // Proc. International symposium on Low power electronics and design. – Monterey, 1998. – P. 197–202.
4. **Chandrasena L. H., Chandrasena P., Liebelt M. J.** An energy efficient rate selection algorithm for voltage quantized dynamic voltage scaling // Proc. International^{14th} symposium on Systems synthesis. – Montréal, 2001. – P. 124–129.
5. **Jejurikal R., Gupta R.** Energy Aware Task Scheduling with Task Synchronization for Embedded Real Time Systems // Proceedings of the international conference on Computers, architecture, and synthesis for embedded system. – Grenoble, 2002. – P.164–169.
6. **Kuo J. L., Chen T. F.** Dynamic voltage scheduling for real-time on low-power variable speed processors // Proc. International conference on Computers, architecture, and synthesis for embedded systems.– Grenoble, 2002. – P. 147–154.
7. **Choi K., Soma R., Pedram M.** Dynamic voltage and frequency scaling based on workload Decomposition // Proc. International symposium on Low power electronics and design. – California, 2004. – P. 174–179.
8. **Gruian F.** Hard real-time scheduling for low-energy using stochastic data and DVS processors // Proceedings of the 2001 international symposium on Low power electronics and design, Huntington, 2001.– P. 46–51.
9. **Hua. S., Qu G.** Energy-efficient dual voltage soft real-time system with (m,k) – firm deadline guarantee // Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems. – Washington, 2004. – P. 116–123.
10. **Mosse D., Aydin H., Childers B., Melhem R.** Compiler-assisted dynamic power-aware scheduling for real-time application // Workshop on Compilers and Operating Systems for Low Power. – 2000.– P. 11–22
11. **Rotenberg E.** Using Variable – MHz Microprocessors to Efficiently Handle Uncertainty in Real-Time Systems // Proceedings of the annual ACM/IEEE international symposium on Microarchitecture. – Austin, 2000. – P. 28–39.
12. **Baums A.** The Real-Time System Energy Consumption Estimation for the Task Dynamic and Statistic Realization Scheduling // Scientific proceedings of Riga Technical University, Computer Science, 5 series, 2005. – Vol. 20. – P. 36–41.

Presented for publication 2006 02 10

A. Baums. Energy Consumption Optimisation in Hard Real-Time System CMOS Processors // Electronics and Electrical Engineering. – Kaunas: Tehnologija, 2006. – No. 4(68). – P. 19–22.

Energy consumption reduction is of primary importance in mobile embedded systems and SoC design. Dynamic voltage scaling (DVS) for CMOS chips is the most effective energy reduction technology. For hard real-time system optimisation DVS based task on-line scheduling method “mini-max” was developed. By it using the deadline is guaranteed at the task execution at minimal power till critical time, after which task is finished at maximal clock frequency and speed. In paper the benefits of the mini-max method using for energy saving is estimated and the optimal frequency selection for hard deadline task execution is analysed. Ill. 7, bibl. 16 (in English; summaries in English, Russian and Lithuanian).

A. Баумс. Оптимизация энергопотребления CMOS процессоров в системах реального времени с жесткими временными ограничениями // Электроника и электротехника. – Каунас: Технология, 2006. – № 4(68). – С. 19–22.

Минимизация энергопотребления является первоочередной задачей при создании мобильных и однокристалльных CMOS систем. Наиболее эффективной технологией для этого является DVS (динамическое масштабирование напряжением). На ее основе создан метод оптимизации энергопотребления для CMOS процессоров „mini-max”. Согласно с этим методом при выполнении задач можно оперативно гарантировать жесткие временные ограничения и оптимальное потребление энергии. Это достигается реализацией задачи с минимальной частотой и энергопотреблением до определенного критического момента, после которого выполнение завершается на максимальной частоте. В работе представляются результаты анализа возможностей метода mini-max и принципы определения ее оптимального частотного режима. Ил. 7, библи. 16 (на английском языке; рефераты на английском, русском и литовском яз.).

A. Baums. Energijos suvartojimo optimizavimas realaus laiko sistemų KMOP procesoriuose su griežtais laiko apribojimais // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2006. – Nr. 4(68). – P. 19–22.

Energijos sąnaudų minimizavimas yra ypač svarbus veiksnys projektuojant mobilios paskirties įterptines sistemas ir vienkristales KMOP sistemas. Pati efektyviausia šiam tikslui technologija yra DVS (dinaminis įtampos keitimas). Jos pagrindu sukurtas CMOS procesorių energijos sąnaudų optimizavimo minimumų ir maksimumų metodas. Atliekant šiuo metodu užduotis galima operatyviai garantuoti griežtus laiko apribojimus ir optimalų energijos vartojimą. Tai pasiekama atliekant užduotį esant minimaliam dažniui ir energijos suvartojimui iki nustatyto kritinio laiko, po kurio vykdomas užbaigiamas maksimaliu dažniu. Darbe pateikti metodo „mini-max“ galimybių analizės rezultatai ir jo optimalaus dažnio režimo nustatymo principai. Il. 7, bibl. 16 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).

3. PIELIKUMS. PREZENTĀCIJA KONFERENCĒ „ELECTRONICS'2006, THE 10TH INTERNATIONAL CONFERENCE. KAUNAS UNIVERSITY OF TECHNOLOGY, MAY 24, 2006”
M. GREITANS, M.KALBERGS, ABILITIES AND APPLICATION OF LEVEL-CROSSING SAMPLING IN DIGITAL SIGNAL PROCESSING.

Abilities and application of level-crossing sampling in digital signal processing

M. Greitāns and M. Kalbergs

Institute of Electronics and Computer Science
kalbergs@edi.lv

OUTLINES

Introduction

Level-crossing sampling principle

Features – SQNR, sampling density, intervals, etc.

Potential applications

Conclusions

ELECTRONICS'2006 • Kaunas May 24 • #1

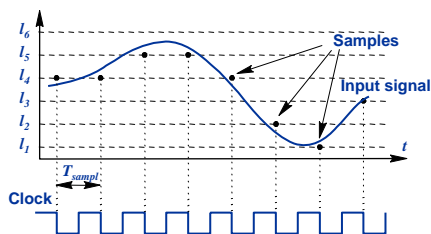
abilities and application of level-crossing sampling

INTRODUCTION

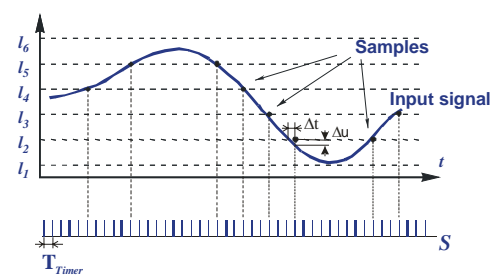
- ❑ **Chips become more bigger and complicated.**
- ❑ **Complicated to provide qualitative synchronization in high frequencies to all parts of chip.**
(10...30% of power consumption goes to distribute clock over chip)
- ❑ **Alternative – clock-less systems.**
(benefits: low power, low EMI, less heat elimination, etc.)
- ❑ **Clock-driven ADC can be substituted by level-crossing ADC.**
(event-based, signal-dependent sampling)

ELECTRONICS'2006 • Kaunas May 24 • #2

LEVEL-CROSSING SAMPLING PRINCIPLE



clock-driven ADC



signal-driven ADC

ELECTRONICS'2006 • Kaunas May 24 • #3

LC-sampling features

□ Signal-to-(quantization) Noise Ratio:

- classical ADC $SNR_{dB} = 1,76 + 6,02n$

where n – number of bits

- level-crossing sampling $SNR_{dB} = 10 \log \left(\frac{3P_x}{P_{x'}} \right) + 20 \log \left(\frac{1}{T_{timer}} \right)$

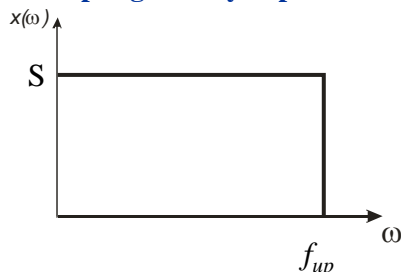
where T_{timer} – period of timer
 P_x – power of signal
 $P_{x'}$ – power of derivative signal

- Does not depend on the number of quantization levels
- Depends on precision of timer
- Depends on properties of input signal

ELECTRONICS'2006 • Kaunas May 24 • #4

LC-sampling features

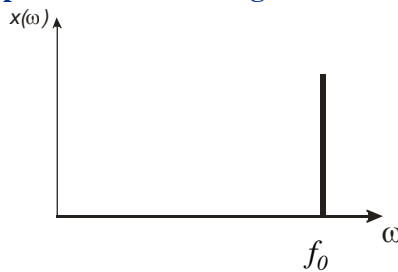
- Sampling density depends on spectral content of signal



Band-limited Gaussian process with zero mean and constant PSD:

$$\text{For level } l_1: E[\sigma_{l_1}] = \frac{2f_{UP}}{\sqrt{3}} \exp\left(\frac{-l_1^2}{4Sf_{UP}}\right)$$

$$\text{total } E[\sigma] = \sum_{k=1}^{2^L-1} \sigma_{l_k}$$



Single sinusoid signal

$$x(t) = A \sin(2\pi f_0 t + \varphi)$$

$$\sigma = 2R_{\Delta} f_0$$

where R_{Δ} – total number of different levels crossed by signal.

LC-sampling features

Time intervals between two adjacent samples:

Single sinusoid signal:
$$\Delta t_n = \frac{1}{2\pi f_0} \left| \arcsin\left(\frac{l_k}{A}\right) - \arcsin\left(\frac{l_{k+1}}{A}\right) \right|$$

around extremes

$$\Delta t_n = \frac{1}{\pi f_0} \left(\frac{\pi}{2} - \arcsin\left(\frac{|l_{\min|\max}|}{A}\right) \right)$$

General case – the upper and lower bounds:

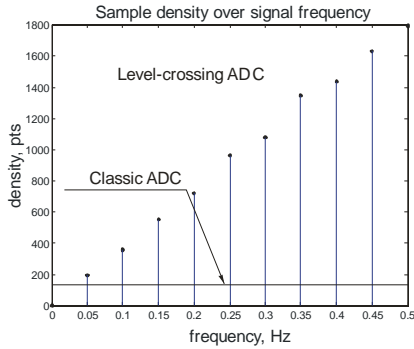
$$T_{\min} \geq \frac{\Delta l_{\min}}{\max(|x'(t)|)} \quad T_{\max} = \infty$$

where

Δl_{\min} – minimal distance between two quantization levels
 $x'(t)$ – derivative signal

abilities and application of level-crossing sampling

Simulation results I



Simulation parameters:

Bit count: 4 bit

Signal length: $\Theta=128s$

Frequency sweep from: 0 till 0.5 Hz step 0.05Hz

Sample density in level-crossing ADC depends from signal frequency, but in traditional ADC it remains constant.

Two types of signals: band-limited *noise* and *sinusoidal*

Signal length: $\Theta=128s$

Frequency: $30/\Theta$

Number of bits: 4

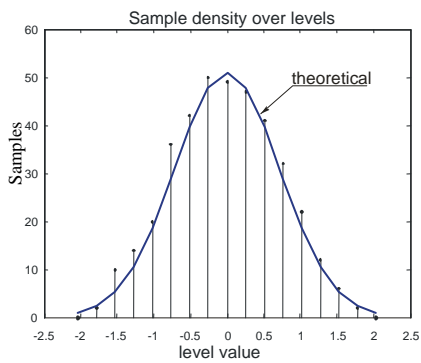
Amplitude of signal: 1V

ELECTRONICS'2006 • Kaunas May 24 • #7

abilities and application of level-crossing sampling

Simulation results II

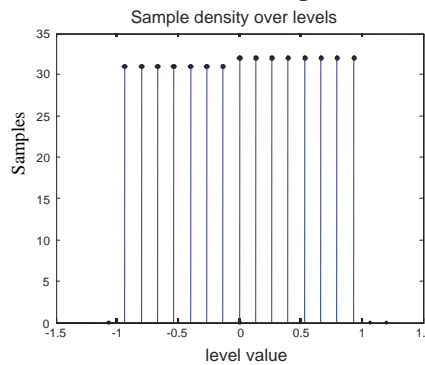
Band-limited noise signal



Increasing levels sample density is decreasing

Sample density of band-limited noise and sinusoidal signal using level-crossing sampling.

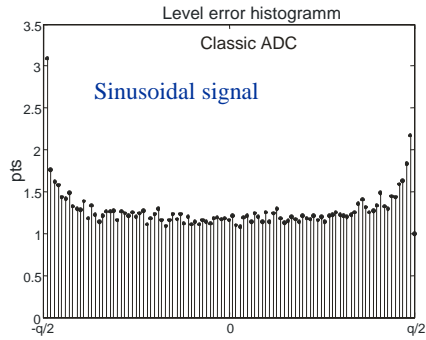
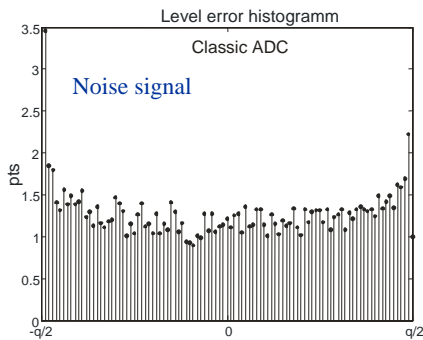
Sinusoidal signal



ELECTRONICS'2006 • Kaunas May 24 • #8

Simulation results III

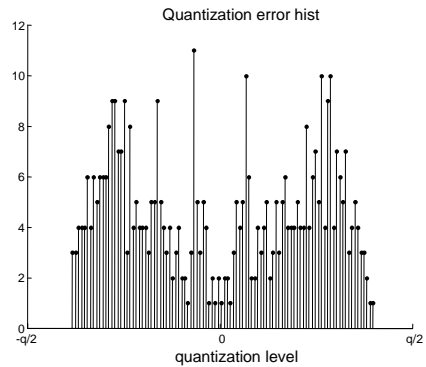
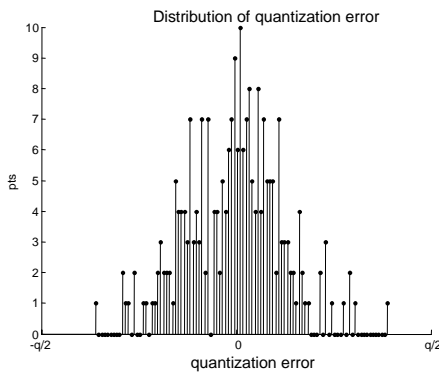
Error distributed constantly between quantization levels



Simulation results III

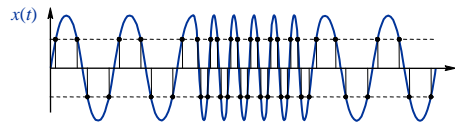
Timer resolution $T_{Times}=0.2s$

Increasing T_{Timer} resolution quantization error decreases and width of distribution decrease.



Potential applications

Non-stationary (in time) signals

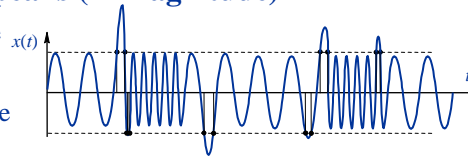


• Signal waveform changing rapidly samples are spaced closer.

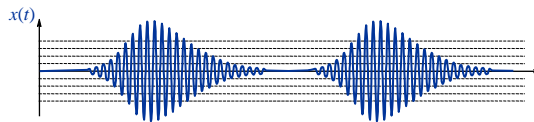
Signals with rare peaks (in magnitude)

In classic ADC case rare peaks introduce quantization error.

Level-crossing ADC have bigger distance between two samples.



Transient signals



CONCLUSIONS

- The main abilities of level-crossing sampling discussed.
- SQNR can be improved by increasing the precision of timer.
- Sampling density reflects the spectral contents of signal: (if signal waveform is changing rapidly, the samples are spaced closer)
- Samples are spaced non-equidistantly – specific DSP is required.
- Application for analysis of transient signals, non-stationary signals and signals with rare peaks

4. PIELIKUMS. PUBLIKĀCIJA KONFERENCES RAKSTOS, INTERNATIONAL REMOTE ENGINEERING VIRTUAL INSTRUMENT SYMPOSIUM, „REV2006”, 29-30 JUNE 2006, MARIBOR, SLOVENIA. M. GREITANS, VIRTUAL INSTRUMENTATION IN AN EXTENDED FREQUENCY RANGE USING DIGITAL ALIAS-FREE SIGNAL PROCESSING.

Virtual instrumentation in an extended frequency range using Digital Alias-free Signal Processing

Modris Greitans

Institute of Electronics and Computer Science

Key words: *Virtual instrumentation, data acquisition, non-uniform sampling, digital signal processing*

Abstract:

The paper presents an advanced Virtual instrumentation approach, which allows the handling of radio-frequency and microwave signals in real-time using standard PC interfaces for data transfer from data collection hardware. Signal acquisition is based on a non-traditional data capturing method – non-uniform sampling, which allows the suppression of frequency aliasing effect and, thus, the extension of the frequency range of processing. Appropriate algorithms of signal analysis have to be used to take into account non-uniformities of sampling and a priori information about signal. LabVIEW graphical development system can be used for VI software designs. The proposed approach could be effective for extending the use of virtual instrumentation principle to application areas, where this has not been possible so far.

1. Introduction

A digital measurement system typically consists of sensors, which deliver an electric signal, signal conditioning, data acquisition and processing, as well as result visualization and instrument control. Virtual instrumentation is a computer-based system that refers to the use of general purpose computer in combination with data collection hardware devices and virtual instrument software. This approach is an interdisciplinary area that includes sensing, hardware and software technologies in order to create flexible and sophisticated solutions. The typical structure of a personal computer based measurement system with virtual instruments is illustrated in Figure 1.

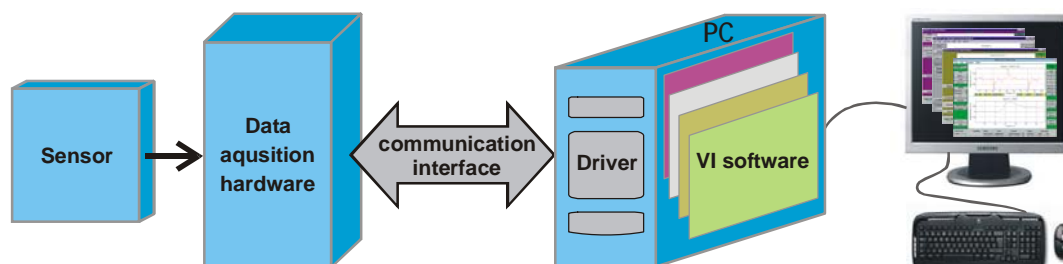


Figure1. Typical structure of virtual measurement system.

Table 1.

	Wired						Wireless		
Interface	RS232	EPP	PCI-E 16x	USB	Fire Wire - IEEE 1394b	Ultra 3 SCSI	802.11	Blue- tooth	ZigBee
Data rate	115kbps	1.8MB/s	250MB/s	480Mb/s	800Mb/s	320MB/s	54Mb/s	1Mb/s	250kbps
Distance	~15m	1.5m	onboard	5m	4.5m	1m	~50m	10m	up to 100m

A plug-in data acquisition board or externally connected data collection hardware transforms the PC into a digital device capable of collecting continuous time signals from sensors and sending commands to actuators. Computer monitor and keyboard (and/or mouse) replace the front panel of the traditional instruments and, in fact, with enhanced features. One can imagine that operating with more powerful PC, more sophisticated software tool and more improved hardware devices, engineers and scientists can create user-defined systems that meet their exact application needs. However, such a concept does not always comply with reality, because there is one more factor, which can influence the performance of the whole system. The data transfer rate of the interface for communication between the data collection hardware and the computer is limited.

Depending on the type of connection to the PC, interfaces can be classified as wired and wireless. Wireless interfaces are increasingly used due to their convenience. Typically they include 802.11 family of standards, Bluetooth and GPRS/GSM interface. Wired connections are serial buses (RS232, USB) or parallel interfaces, such as EPP, GPIB, SCSI and PC system buses (i.e. ISA, PCI, PCI-X). Therefore, the throughput between data collection hardware and CPU of computer can be up to few hundred MBps for integrated systems, up to several tens of MBps for distributed wired systems and up to few MBps for wireless systems. The Table 1 summarizes the basic parameters (data transfer rate and distance) of standard interfaces used by computers.

Traditional Digital Signal Processing (DSP) is based on analog-to-digital (A/D) conversion which provides uniformly spaced signal samples. The well-known Nyquist criterion states that the sampling rate must be at least twice as fast as the highest frequency component in the input signal. In real-time system the bandwidth of interface has to ensure the transfer of all captured data without loss. Consequently, the data transfer rate of the interface determines the possible sampling frequency of A/D converter. To prevent distortion due to frequency aliasing, the analog anti-alias filter preceding an ADC have to be used. In such a way the limited throughput of interface limits the frequency range of virtual instrumentation too, and its application for analyzing high frequency signals is restricted.

2. Digital Alias-free signal processing technology

Digital signal processing operates with digital data. The representation of a continuous time signal by sequences of their samples is based classically on the famous Shannon sampling theorem, which determines that samples are spaced equidistantly. Such representation provides the appearance of aliasing (periodicity of spectrum) into frequency domain. The bandwidth of analysis is connected with the time period between samples. In the traditional DSP case, the only way to expand the bandwidth of analysis is increasing the sampling frequency. However this possibility is limited. In addition such an approach produces an increased number of bits requiring more complicated operation of data transfer.

An alternative way of achieving the capability of fully digital processing of wideband signals is the elimination or suppression of aliasing. In this case the non-traditional principles of analog-to-digital conversion has to be involved and so called alias-free processing approaches have to be exploited.

Digital Alias-free Signal Processing (DASP) is an advanced DSP technology for processing radio-frequency and microwave signals in a direct digital way. The highest frequency of the signals to be processed in this way is not necessarily limited by half the sampling rate. This extraordinary performance is achieved by sampling signals in a special non-uniform way leading to suppression of frequency aliasing. The digital processing of signals with frequencies up to GHz can be handled with mean sampling rate, which corresponds to the achievable data transfer rates of standard PC interfaces.

The advantage of DASP technology is basically due to the usage of special signal digitization and processing techniques. The substance of DASP Virtual instrumentation is determined by two blocks – the special sampling driver in hardware and specific DSP algorithms in software. While the essence of the idea of introducing non-uniformities into the sampling process seems to be elementary, its implementation is actually not so trivial.

3. Data acquisition hardware with non-uniform sampling driver.

The theoretical background of the possibility of suppressing frequency aliasing originated in the 60th of the 20th century. However, the first intentions of implementing the theoretical ideas of non-uniform sampling into hardware systems appeared only after a few decades. An alias-free Signal Recorder & Analyzer built in 1995 can be named as the first system, which deliberately pseudo-randomized the sampling procedure and digitally processed signals in the frequency range dc-350MHz while the mean sampling rate is about 100MSamples per second (MSps) [1]. International recognition for such an approach was obtained by the DASP Lab System [2], which was awarded the European IT Prize in 1997. DASP Lab System performs fully digital analysis of signals in Time, Frequency and Modulation Domains in the frequency range up to 1.2 GHz using the mean sampling frequency 80MSps. The wideband RF signal digitizer [3] can be mentioned as a recent development of data collection hardware devices, which are based on deliberately non-uniform sampling scheme. The main parameters of a digitizer are: the alias-free input signal bandwidth - 669MHz, spurious-free dynamic range - >55dB, mean sampling rate -53.5 MSps, resolution - 12 bit. The hardware design of this data acquisition device is illustrated in the Figure 2. As can be seen, the board consists of just a few chips: ADC, FIFO memory, sampling driver and interface controller.

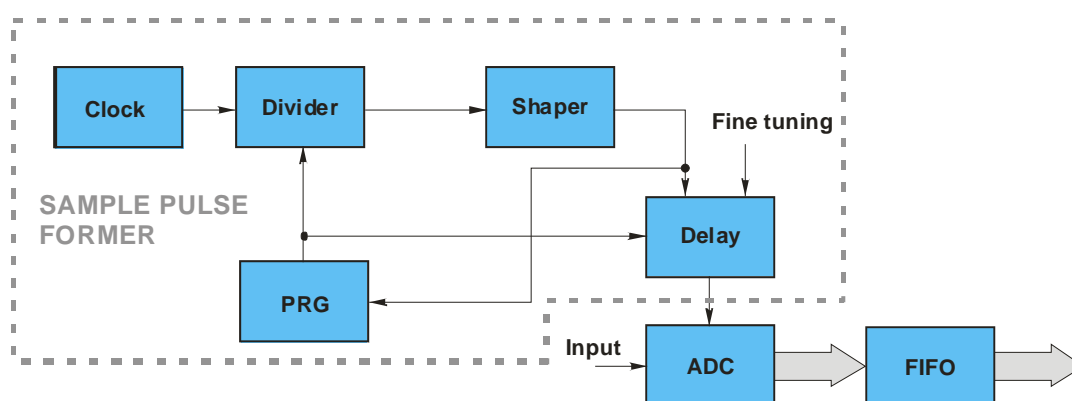


Figure 2. Schematic block diagram of DASP acquisition hardware.

Table 2.

Rate Bits	1-10 Mb/s	10-20 Mb/s	21-40 Mb/s	60-80 Mb/s	81-100 Mb/s	101-120 Mb/s	121-140 Mb/s	141-200 Mb/s	201-250 Mb/s
8				AD9057 120MHz			AD9054 380MHz	AD9411 700MHz	AD9480 750MHz
10			AD9203 390MHz	AD9214 300MHz		AD9216 500MHz		AD9411 700MHz	
12		AD9237 200MHz		AD9236 500MHz		AD9432 500MHz	AD9433 750MHz		AD9430 700MHz
14			AD9244 650MHz	AD6645 270MHz			AD9445 615MHz		
16	AD7400 100MHz			AD10677 210MHz	AD9446 540MHz		AD9461 610MHz		

As an essential fact it should be stressed that the implementation of A/D conversion of DASP digitizer can be based on conventional ADC chips, because their input bandwidth typically is much wider than the maximum sampling rate. This feature allows the carrying out of the principle that the frequency range of analysis exceeds the mean sampling rate, which determines also the data transfer rate. The Table 2 shows the examples of available ADCs, which are grouped according to the data transfer rates in different columns and according to the resolution bits in different rows. The actual input bandwidth of each ADC is displayed into the cell below the ADC model. It can be found that the possible frequency range extension varies about 4-12 times.

Unlike the traditional instruments, DASP digitizers perform pseudorandom sampling of input signal. Sample values have to be taken at time instants predetermined with high accuracy (picoseconds range). This requirement presents a serious engineering problem, because the main hardware error, causing the noise like artifacts in processing results is sampling point flow fluctuations or jitter error. To control the operation of pseudorandom sampling, special sampling driver has to be developed. The Figure 3 illustrates the structure of the DASP digitizer [3]. The sampling pulses are formed combining digital (clock frequency dividing) and analog (pulse delaying) operations, which are managed by the use of pseudo-random number generator (PRG). It should be stressed that all circuits used have to be extremely precise to provide very accurate timing. For example, the device presented in [3] uses the clock frequency 669.3266 MHz, which ensures low-jitter timing grid with 1.494 ns step.

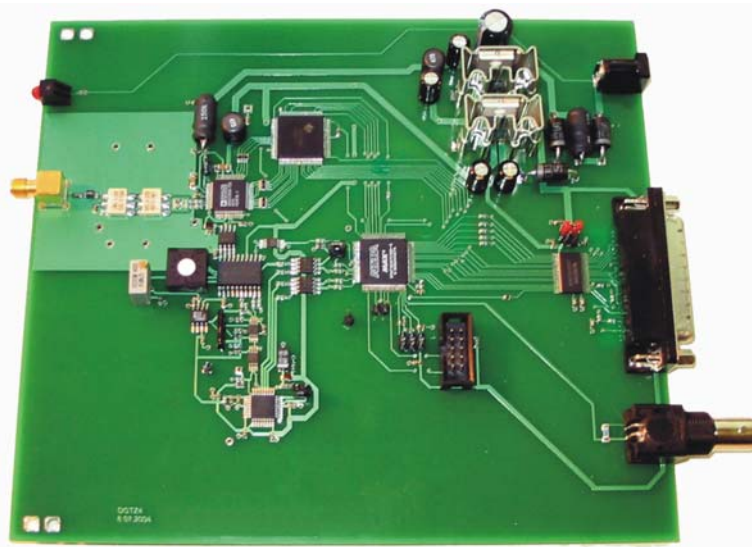


Figure 3. Hardware design of DASP digitizer.

That supports alias-free analog bandwidth only up to 335 MHz. To reach wider bandwidth for DASP applications an additional one-step controllable delay block is applied. It is designed on the basis of an adjustable delay line with 33 ps/step. The mentioned structural solution results in the actual sampling rate 53.546 MSps, where sampling instants are taken from the supposed timing grid with 1338.653 MHz frequency. In such a way the alias-free analog bandwidth 669.326 MHz is guaranteed.

In addition to the above-mentioned structure a test-signal source can be built into the hardware, to provide the capability of calibration procedure.

4. Processing of non-uniformly sampled signals

Particular attention within DASP virtual instrumentation is focused on optimal signal processing from its non-uniform samples. One way is to modify traditional algorithms (for example, Discrete Fourier Transform) to take into account the variety of sampling intervals. However, the achieved results are not satisfactory, if the sampling density is lower than twice the bandwidth of the signal. To improve processing any a priori information about signal properties should be taken into account.

4.1 Sequential component extraction (SECOEX)

An effective approach to analyzing the signals with discrete spectra is the so-called SECOEX method [4]. Quite often the goal of spectral estimation is focused on the search for discrete (sinusoidal) components. Commonly used DFT does not solve the task precisely, because it acts on the predefined equally spaced frequency grid. If the signal contains discrete component with the frequency located between the analysis frequencies, the result is not always obvious.

The SECOEX approach is cyclic and based on the extraction of one discrete spectral component in each cycle. The cycle includes frequency estimation of most powerful spectral line, estimation of components parameters, and subtraction of this component from the signal. Cycles are continued until the predefined limits (number of components or residual power) are reached.

4.2 Spectrum support function

The estimation of continuous spectra in the extended frequency range, which exceed Nyquist limit several times, is a more complicated task. Obviously, there can be many signals with different characteristics that occupy the same spectral bandwidth, for example, multitone signal, chirps, different impulses, variously "colored" or "white" noise etc. Each of these signals has its own spectral characteristics. The basic idea of enhanced processing is not supporting the whole frequency range equally, but of concentrating more processing on the regions with more powerful spectral parts of the signal. To realize it the concept of spectrum support function (SSF) can be introduced. This function determines the "weight of importance" for each frequency of analysis. It can be considered that the traditional spectral analysis approaches (for example, Fourier transform) typically operates with a constant SSF in the whole frequency range of analysis. They do not separate the regions where the actual components of the signal are to be found. If a signal to be processed consists of several subcomponents placed within various separate locations in the frequency domain, it is not reasonable to support the whole frequency band equivalently. Instead of that the processing should be concentrated on the frequency regions of signal components. In this case the spectrum support function can be constructed as a binary function, which has the value "0" for frequency subsets that are not carried by the signal and the value "1" for the frequency subsets that can be carried by the signal. If a signal spectrum is continuous in the whole

frequency range, and there are several regions of differently powerful subcomponents, the stepwise constant function can be used as spectrum support function.

4.3 Signal-dependent analysis

As a consequence of the above considerations we can assume that each frequency of spectral analysis can be supported with a different weight. The question arises - how to select the exact form of spectrum support function for each signal to be processed. The signal-dependent analysis approach theoretically assumes an equivalence between the spectral support and spectral density functions of a signal, or in the practical applications the spectrum support of analysis is adapted to the power spectral density of signal. In the method described in [5], the signal dependent spectral transformation method involves statistical characteristics of the signal into transformation kernel through values of signal autocorrelation function, which is inverse Fourier transform from power spectral density function. If there is no a priori information about spectral content before analysis, the spectrum support function can be estimated in an iterative way.

5. Software environments for virtual instruments building

To develop a complete measurement system, the data acquisition hardware and appropriate signal processing algorithms have to be integrated with virtual instrument software that provides a graphical user interface for device control and result visualization. The basic requirement for the software environment, in which the virtual instrument is developed, is: high computation performance has to be combined with a comfortable platform for building a user interface and with the possibility of managing hardware ports.

Initially the DASP virtual instrument development was orientated towards the use of MATLAB computation and data visualization software package. The module for operations with communication ports has been written in C and implemented in MATLAB MEX-file format. The user interface is implemented using MATLAB Handle Graphics system. The panels of DASP-Lab virtual instruments are shown in Figure 4.



Figure 4. Instrument panels of DASP-Lab system programmed in MATLAB.

The virtual instrument building on the bases of NI LabWindows/CVI package can be considered as the next stage. In this case the programming has to be done in language C that allows the compilation of fast and effective code including simple low level management of hardware ports. Two central instruments for signal analysis are the oscilloscope in the time domain and the spectral analyzer for processing in frequency domain. As an example, in the Figure 5 the spectrum analyzer is demonstrated. The drawback to the use of NI LabWindows/CVI is its orientation only to the MS Windows, which is not always best suited to real-time application.

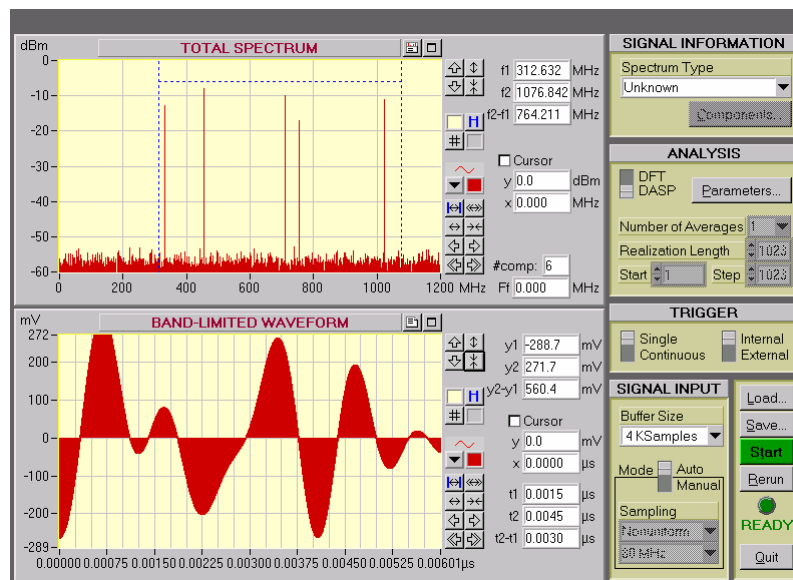


Figure 5. DASP spectrum analyzer designed in LabWindows/CVI.

NI LabVIEW software package is a convenient tool for creating virtual instruments and designing graphical user interfaces on several platforms. Although it is a graphical programming language, its versatility allows the easy incorporation of modules written in other languages. If standard PC interfaces are used for communication with DASP digitizer, there is no necessity to develop special driver for attached hardware. These are arguments why the latest developments in the framework of DASP VI are designed using NI LabVIEW. Graphical programming approach is more easily available for wider range of end-users. The block diagram of spectrum analyzer is shown in Figure 6. It demonstrates the simplicity of building virtual instruments and the possibility of adapting them to specific requirements of each end-user.

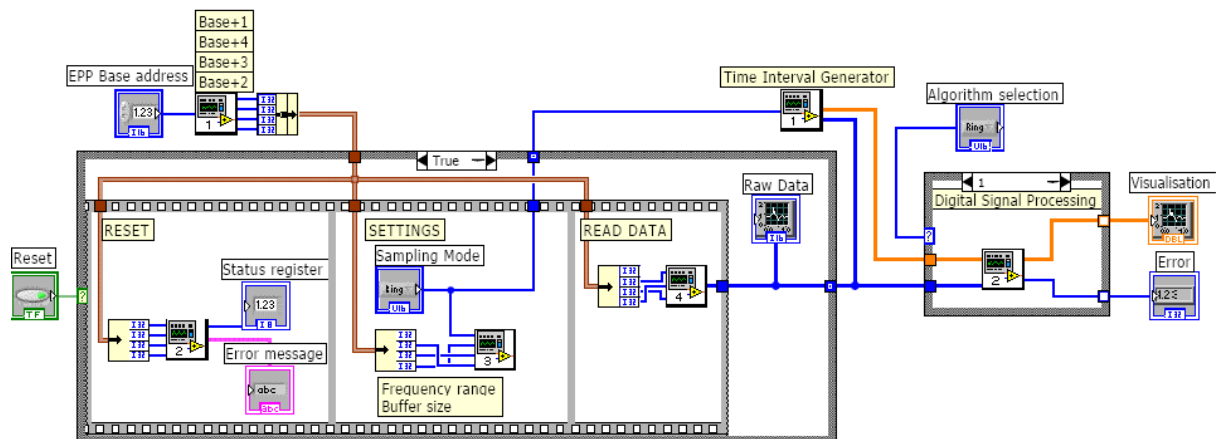


Figure 6. DASP Virtual instrumentation software in LabVIEW.

6. Conclusions

The virtual instrumentation approach is an effective way of building flexible measurement systems. The significant advantage is the possibility of utilizing the PC resources for data visualization and storage, as well as to simulate the system controls on the bases of keyboard and mouse. The drawback of such systems is the limited capacity of data transfer channel between the acquisition hardware and the PC interface. If standard uniform sampling approach is used, that limits the frequency range of measurement system in real-time applications, and leads to the restricted use of virtual instruments in the field of radio frequencies and microwaves.

The paper demonstrates an advanced approach of the analog-to-digital conversion, which is based on non-uniform sampling. Such sampling strategy allows the suppression of frequency aliasing and extends the frequency range of digital processing. From the practical point of view the problem resolves into two directions – special sampling driver in hardware and specific algorithms in software. The ADC chips typically have the feature that their input bandwidth is wider than the maximal allowable sampling rate, and that is exploited in designs of DASP digitizers.

Non-uniform sampling technique requires the appropriate digital signal processing, which takes into account the exact location of each sample point. In addition, it is essential to involve a priori information about signal properties into analysis. That leads to the concept of spectrum support function and to the construction of signal-dependent transformation. The software environment for designs of a virtual instrument has to integrate the high computation performance with the comfortable building of a user interface and with the convenient management of hardware ports. As examples MATLAB, NI LabWindows/CVI and NI LabVIEW can be mentioned. The latest developments of DASP VI are orientated towards the use of NI LabVIEW, because it is multi-platform graphical programming environment comfortable for virtual instrument building and does not require high-level skills in programming.

References:

- [1] Artyukh, Yu.; Bilinskis, I.; Rybakov, A.; Vedin, V.: "Alias-free Signal Recorder & Analyser", Proc. of the Workshop on Sampling Theory and Application, Jurmala, Latvia, Sep., 1995, pp. 220-222.
- [2] Artyukh, Y.; Bilinskis, I.; Greitans, M.; Mednieks, I.; Rybakov, A.: "DASP-Lab System – a demonstrator of the new DSP technology", Automatic Control and Computer Science, Nr.6, 2000, pp. 3-21.
- [3] Artyukh, Yu.; Bilinskis, I.; Boole, E.; Rybakov, A.; Vedin, V.: "Wideband RF signal digitising for high purity spectral analysis", Proc. of the International Workshop on Spectral Methods and Multirate Signal Processing, Riga, Latvia, Jun., 2005, pp. 123-128.
- [4] Mikelsons, A.: "Alias-free spectral estimation of signals with components of arbitrary frequencies", Proc. of Cost #229 workshop on Adaptive Methods and Emergent Techniques for Signal Processing and Communications, Ljubljana, Slovenia, 1994, pp. 105-108.
- [5] Greitans, M.: "Spectral analysis based on signal dependent transformation", Proc. of the International Workshop on Spectral Methods and Multirate Signal Processing, Riga, Latvia, Jun., 2005, pp. 179-184.

Author:

Modris, Greitans, Dr.sc.comp.
Institute of Electronics and Computer Science
14 Dzerbenes str., Riga, LV1006, LATVIA
Email: modris_greitans@edi.lv

5. PIELIKUMS. PUBLIKĀCIJA KONFERENCES RAKSTOS, INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND MULTIMEDIA APPLICATIONS, „SIGMAP2006” M. GREITANS. PROCESSING OF NON-STATIONARY SIGNAL USING LEVEL-CROSSING SAMPLING

PROCESSING OF NON-STATIONARY SIGNAL USING LEVEL-CROSSING SAMPLING

Modris Greitans

Institute of Electronics and Computer Science

14 Dzerbenes str., Riga, LV1006, LATVIA

Email: modris_greitans@edi.lv

Keywords: Clock-less design, non-stationary signal, level-crossing sampling, time-frequency representation.

Abstract: The spectral characteristics of multimedia signals typically vary with time. Preferably, the sampling density of them would comply with instantaneous bandwidth of signal. The paper discusses the level-crossing sampling principle, which provides such capability for analog-to-digital conversion. As the captured samples are spaced non-uniformly, the appropriate digital signal processing is required. The non-stationary signal is characterized by time-frequency representation. Its classical approaches are inspected for applicability to analyze the data obtained by level-crossing sampling. Several enhancements of short-time Fourier transform approach are proposed, which are based on the idea to minimize the reconstruction error not only at sampling instants, but also between them with the same accuracy. Additional benefits are gained if the instantaneous spectral range of analysis is complied with local sampling density: artifacts are removed, complexity of calculations is decreased. The performance of algorithms is demonstrated by simulations. Presented research can be attractive for clock-less designs, which receive now an increasing interest. Their promising advantages can play a significant role in future electronics' development.

1 INTRODUCTION

Conventional digital signal processing techniques often consider the stationarity of a signal within a frame of analysis. It is assumed that the statistical characteristics of signal do not change with time. The concept of stationarity provides the possibility of fixing the sampling rate (it should be at least twice as high as the maximum signal frequency), as well as of constructing effective processing methods, for example the Discrete Fourier transform (DFT). However, natural signals typically are time-varying, and they can be a mixture of events localized both in time and frequency (Akay, 1998).

Intuitively speaking, the non-stationarity of a signal should be reflected in the process of analog-to-digital (A/D) conversion. For example, let us inspect a signal with high instantaneous frequency regions and low instantaneous frequency in other regions. It is more efficient to sample the low frequency regions at a lower rate than the high frequency regions. Consequently, with appropriate non-equidistantly spaced samples one might approximate a signal with fewer samples per interval than in the uniform sampling case, where sampling frequency is defined taking into

account only the highest signal component. Two conclusions follow: non-uniform sampling is the natural choice for the discrete representation of a non-stationary signal, and the non-uniformity of sampling process has to be caused by the local properties of signal.

The work presented in this paper is based on the idea of abandoning traditional clock-driven A/D conversion and the uniform digital signal processing, which typically follows it. Instead of that, a clock-less structure of data processing system is suggested, where the A/D conversion is signal-driven. To illustrate the difference, the processing chains of both approaches are illustrated in the Fig.1. Let us emphasize the key benefits of the asynchronous electronics: lower power consumption, absence of the clock screw, reduced heat elimination, lower EMI, automatic adaptation to physical properties, etc. (Hauck, 1995). The popular types of signal-dependent sampling are based on zero-crossing, reference signal crossing, level-crossing or send-on-delta concepts. Each of them has its own advantages and limitations, however joint features are: the signal samples can be spaced non-uniformly, local sampling density depends on local properties of signal, and it is impos-

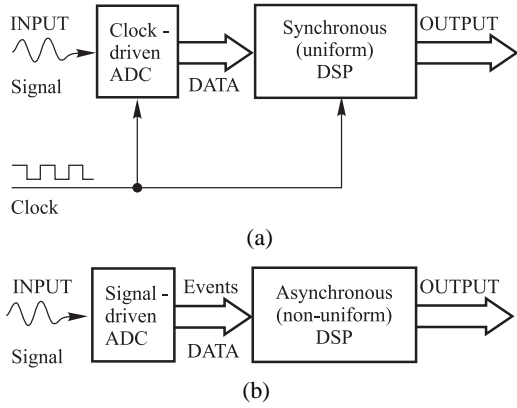


Figure 1: Structures of DSP system based on different paradigms: synchronous (a), asynchronous (b).

sible to determine the sampling time instants in advance. The paper discusses digital signal processing if the level-crossing sampling scheme is used to capture digital data from a continuous time signal.

2 LEVEL-CROSSING SAMPLING

The principle of uniform sampling is illustrated in Fig.2a: sampling is driven by an external clock with fixed period T_{sample} that gives the equidistantly spaced samples. The level-crossing sampling (LCS) scheme is based on the principle that samples are captured when the continuous time input signal crosses predefined levels. Typically, the quantization levels are uniformly disposed along the amplitude range of the signal as is shown in Fig.2b.

Such a sampling strategy is not new and has been known at least since the late 1950s (Ellis, 1959). Various terms are used to name it: event-based sampling, level-crossing sampling, magnitude-driven sampling, and sometimes, sampling in the amplitude domain. The variety of existing terminology shows that it is really a generic concept adapted to a broad spectrum of technology and applications. It has been shown that level-crossing sampling has several interesting properties and is more efficient than traditional sampling in many respects (E. Allier and Renaudin, 2003). Classical A/D conversion implements clock-driven sample-and-hold (S/H) operation, which is followed by quantization operation. Considering an ideal clock and an ideal S/H, anyway there is imprecision of conversion due to the limited number of quantization bits L . The Signal-to-(quantization)Noise Ratio (SNR) of classical ADC can be expressed as

$$SNR_{dB} = 1,76 + 6,02L, \quad (1)$$

and it depends only on the resolution of the converter.

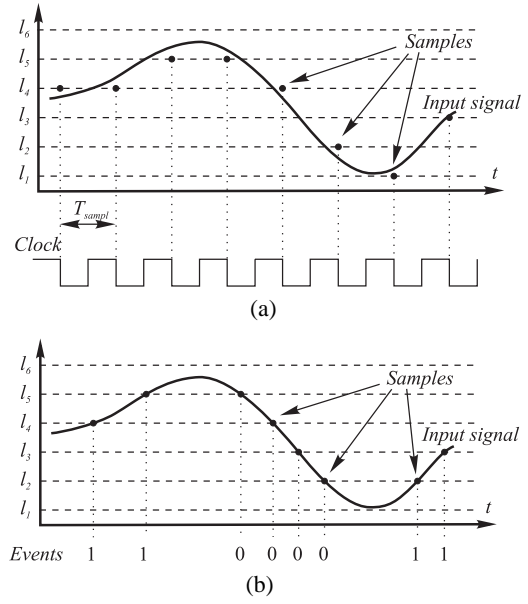


Figure 2: Analog-to-Digital conversion: clock-driven (a) vs. signal-driven level-crossing sampling case (b).

In the level-crossing based A/D converter, since a sample is taken only when a level is crossed, the amplitude value of the sample is exact. Due to the fact that samples are spaced non-equidistantly, the application of LCS often requires that the time instant of the sample also be known. In practice, the time interval is measured by a timer that quantizes the time with certain resolution T_{timer} . The SNR in this case can be estimated as (E. Allier and Renaudin, 2003):

$$SNR_{dB} = 10 \log \left(\frac{3P_x}{P_{x'}} \right) + 20 \log \left(\frac{1}{T_{\text{timer}}} \right), \quad (2)$$

where P_x is power of the random input signal, and $P_{x'}$ is power of its derivative. In this case SNR does not depend on the number of quantization levels, but depends on the properties of the input signal and on the precision of the timer. Signal-to-noise ratio can be improved simply by decreasing T_{timer} .

The goal of the proposed paper is to explore the use of the level-crossing sampling technique for analysis of a non-stationary signal. In this context, the evaluation of the local sampling density can play a significant role, because it is connected with the local statistical characteristics of a signal. If a signal is changing rapidly, the samples are spaced closer, and conversely - if a signal is varying slowly, the samples are spaced sparsely. The variability of waveform is linked with spectral content, and thereby the local sampling density can be used to estimate the instantaneous maximum frequency of signal.

If the input signal is single sinusoid

$$x(t) = A \sin(2\pi f_0 t + \varphi), \quad (3)$$

where A is the amplitude, f_0 - the frequency and φ - the initial phase, the sampling density can be expressed as

$$\sigma = 2R_{\Delta}f_0, \quad (4)$$

where R_{Δ} is the total number of different levels crossed by the signal.

Determining the sampling density of a broadband process is not as elementary as for a mono-harmonic signal. Analytically it is investigated for band-limited Gaussian process with zero mean and constant spectral density

$$P_x(f) = \begin{cases} S & |f| \leq f_{UP} \\ 0 & \text{otherwise} \end{cases}. \quad (5)$$

The expected number of level l_0 crossings per time unit can be expressed as (Mark and Todd, 1981)

$$E[\sigma_{l_0}] = \frac{2f_{UP}}{\sqrt{3}} \exp\left(\frac{-l_0^2}{4Sf_{UP}}\right). \quad (6)$$

To calculate the sampling density, it is necessary to sum up the sampling instants of all the quantization levels l_k

$$E[\sigma] = \sum_{k=1}^{2^L-1} \sigma_{l_k}. \quad (7)$$

One more of the main parameters describing the sampling process is the time interval between two adjacent samples $\Delta t_n = t_{n+1} - t_n$. The mean value of the interval is tied with sampling density as $|\Delta t_n| = \frac{1}{\sigma}$. The exact Δt_n values can be estimated analytically only for special cases, i.e., for the mono-harmonic signal (3). If the signal crosses the level l_k at the time instant t_n and the level l_{k+1} at the time instant t_{n+1} , the Δt_n can be calculated as

$$\Delta t_n = \frac{1}{2\pi f_0} \left| \left(\arcsin\left(\frac{l_k}{A}\right) - \arcsin\left(\frac{l_{k+1}}{A}\right) \right) \right|. \quad (8)$$

Around extremes the signal crosses the same level twice and the distance between crossings is

$$\Delta t_n = \frac{1}{\pi f_0} \left(\frac{\pi}{2} - \arcsin\left(\frac{|l_{min} \mid max|}{A}\right) \right). \quad (9)$$

If Δt_n cannot be estimated analytically, the upper and lower bounds of time interval can be evaluated based on the signal parameters. The minimum distance is determined as

$$T_{min} \geq \frac{\Delta l_{min}}{max(|x'(t)|)}, \quad (10)$$

where Δl_{min} is the minimal distance between two quantization levels, and $x'(t)$ is first derivative of the signal. The case, where the signal crosses the same level twice, is distinct, because $\Delta l = 0$ and T_{min} can

reach zero. The upper bound of Δt_n is infinity, because the level-crossing sampling might not be triggered if the signal waveform is located between two consecutive quantization levels. To avoid this, the distance between quantization levels has to be less than the amplitude of the signal.

In addition, the following facts should be noted - if a signal waveform has some regularities, the sample flow has the same regularities as well. This effect often leads to a problem that the methods, which are derived for deliberately non-uniform sampling, do not always work satisfactorily for a particular case - level-crossing sampling, which provides signal-dependent non-uniform data. The level-crossing based analog-to-digital conversion is asynchronous in the sense that it does not have the clock that determines the positions of samples. That leads to a drastic change in the standard signal and data processing and initiates a new research area - asynchronous signal processing.

3 CLASSICAL TFRs AND NON-UNIFORM SAMPLING

The non-stationary signal is characterized by time-frequency representation (TFR). As the signal samples captured according to the level-crossing principle are spaced non-uniformly, the appropriate digital signal processing is required. In this section, the applicability of classical TFR approaches to analyze LCS data is inspected. The time-frequency representation is characterized by points on a time-frequency gram. For practical applications it is assumed that a finite duration Θ of bandlimited to Ω signal is observed. The traditional approaches for TFR calculations are based on Short-time Fourier transform (STFT) (Gabor, 1946), joint time-frequency distribution (Cohen, 1995) and wavelet transform (WT) (Chui, 1992).

3.1 Short-time Fourier transform

The classical method for analyzing non-stationary signals is short-time Fourier transform. It was proposed by Gabor (Gabor, 1946). STFT is based on the well known Fourier transform. The basic idea of STFT is to introduce a time window, which is moved along the signal, and in such a way the time indexed spectrogram of $x(t)$ is defined as

$$STFT(t, f) = \int_{-\infty}^{\infty} x(\tau)w^*(t-\tau) \exp(-j2\pi f\tau)d\tau, \quad (11)$$

where $w(t)$ is a time window and $*$ denotes the complex conjugates.

In the case of finite number of discrete samples $x_n = x(t_n)$, $n = \overline{1, N}$ (N is a number of samples

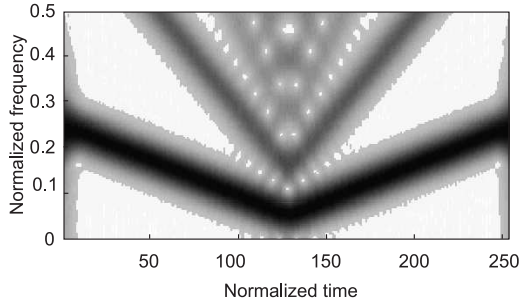


Figure 3: STFT based time-frequency representation of test-signal sampled by crossing 7 levels.

within time interval Θ), the STFT based TFR on the uniformly spaced time-frequency grid with frequency step $\frac{1}{\Theta}$ and time step $\frac{1}{2\Omega}$ can be calculated as

$$TFR_{STFT}(k, m) = \left| \sum_{n=1}^N x_n w^*(k/2\Omega - t_n) \exp(-j2\pi t_n m / \Theta) \right|. \quad (12)$$

The expression (12) uses the general form of DFT, in which the restriction, that requires the uniform spacing of samples $x_n = nT$, can be ignored. To examine what happens if this expression is used for analysis of level-crossing samples, the single chirp (parameters of it will be described in the Section 6) is chosen as a test-signal. The Fig.3 illustrates the fact that in addition to the true component, spurious components appear at the higher odd harmonics. These artifacts are due to the use of LCS approach for signal with regularities in the waveform. The additional source of artifacts can be the absence of the orthogonality of transformation functions $\exp(-j2\pi t_n m / \Theta)$ if t_n are not placed uniformly.

A well-known problem inherent in STFT is the inverse relationship between time and frequency resolutions. Extension of the window's $w(t)$ length improves the frequency resolution, but at the same time degrades the temporal selectivity. To overcome this difficulty of short time Fourier transform, alternative methods of time-frequency analysis have been developed. The two most popular of them are a wavelet transform and a Wigner-Ville distribution.

3.2 Wavelet transform

The wavelet transform of a continuous-time signal $x(t)$ is defined as

$$WT(t, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(\tau) h^* \left(\frac{t-\tau}{a} \right) d\tau, \quad (13)$$

where a is the scaling factor and $h(t)$ is the so-called analyzing wavelet. The time-frequency version is obtained by making the substitution $a = f_0/f$. The

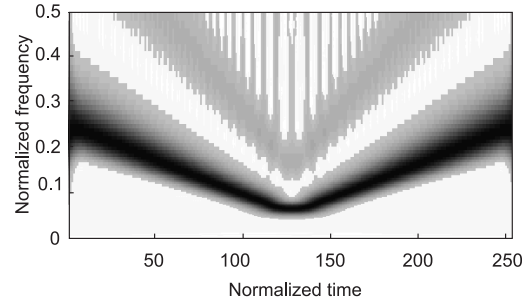


Figure 4: WT based time-frequency representation of test-signal sampled by crossing 7 levels.

analysis can be viewed as a filter bank comprising bandpass filters with bandwidths proportional to frequency. The multi-resolution nature of wavelet analysis leads to some limitations. Wavelet transform uses a scaling profile such that frequency resolution decreases at high frequencies, and temporal resolution decreases at low frequencies. While this choice of scaling leads to nice mathematical structures and algorithms, there is no physical reason to assume that it corresponds to natural structure behavior. For discrete WT, in order to get the best performance of analysis, the time- and scale-sampling grid often should be considerably over-sampled, that introduces the redundancy in the TFR.

The general form of time-frequency representation based on discrete wavelet transform can be expressed as

$$TFR_{WT}(k, m) = \left| \frac{1}{\sqrt{f_0\Theta/m}} \sum_{n=1}^N x(t_n) h^* \left(\frac{k/2\Omega - t_n}{f_0\Theta/m} \right) \right|. \quad (14)$$

Such a notation enables the processing of both uniformly and non-uniformly sampled data. The nice mathematical feature of WT for equidistantly spaced samples states: for any k and $a = 2^m$ ($k, m \in \mathbb{Z}$) the $\{h(t_h)\}_{(m,k)}$ is a subset of one discrete wavelet, which is uniformly sampled at the sampling frequency of the signal. In the case of non-equidistantly spaced samples this property is lost, because the values of wavelet $h(t)$ have to be calculated at different points set $\{t_h\}_{(m,k)}$ for each scaling factor a (or frequency of analysis $f = f_0/a$). Due to this fact, the computation complexity of WT in the non-uniform sampling case considerably exceeds the complexity of the uniform sampling case.

The time-frequency representation obtained by (14), if level-crossing sampling is used, is demonstrated in Fig.4. It shows the reduction of the temporal resolution in the low frequency region and diminished spectral resolution in the high frequency region. The additional artifacts appear as well.

3.3 Wigner-Ville distribution

Time-frequency analysis, based on the use of Wigner-Ville function, is defined as

$$WVD(t, f) = \int_{-\infty}^{\infty} x\left(t + \frac{\tau}{2}\right) x^*\left(t - \frac{\tau}{2}\right) \exp(-j2\pi f\tau) d\tau. \quad (15)$$

It provides high-resolution representation in time and in frequency for mono-component signals. However, if the signal consists of several subcomponents, additional interference or cross-terms appear due to the quadratic nature of kernel and non-linear properties of it. In order to mitigate this deleterious effect, a variety of modified kernels have been introduced. One way to remove the interference is by smoothing the time-frequency plane, but this will be at the expense of decreased resolution in both time, and frequency. A promising approach of how to suppress cross-terms and improve resolution is the use of signal-dependent kernels (Baraniuk and Jones, 1993).

A discrete form of the Wigner-Ville distribution (WVD) can be expressed as

$$TFR_{WVD}(k, m) = 2 \left| \sum_{n=1}^N x(k/2\Omega + t_n) x^*(k/2\Omega - t_n) \cdot \exp(-j4\pi t_n m / \Theta) \right|. \quad (16)$$

The necessity of knowing signal values at time instants $\tau + t_n$ and $\tau - t_n$ for all $n = \overline{1, N}$ leads to the fact that the expression (16) can be used only for uniform and specifically regular sampling series. Therefore it is impossible to use the WVD approach for processing data captured by the level-crossings.

4 ENHANCEMENTS OF DFT

It can be concluded from the discussion above, that the most useful approach for practical applications using the level-crossing sampling is based on STFT. However it has to be enhanced and adjusted to the LCS to suppress the presence of spurious components.

The key operation of discrete STFT is the DFT algorithm, which is applied to the windowed signal samples. Thus the STFT enhancement can be reduced to the development of DFT-like methods, which take into account LCS features. The level-crossing sampling principle provides not only samples at certain events, but also the rule that the signal between two sampling instants does not cross any quantization

level. This information can be exploited in the processing. The proposed idea is to minimize the error between the original signal and that reconstructed by the Fourier series, not only at sampling time instants, but also between them with the same accuracy. The problem lies in the fact that the reconstruction error can be obtained only at the time moments in which the signal samples are known.

Basing on the Fourier series the signal waveform can be reconstructed from its spectral estimates by the following formula

$$\hat{x}(t) = \sum_m X_m \exp(j2\pi t f_m), t \in [0, \Theta], \quad (17)$$

where X_m are Fourier coefficients at frequencies $f_m = m/\Theta$. If the original continuous-time signal is $x(t)$, the reconstruction error is

$$\varepsilon(t) = x(t) - \hat{x}(t), \quad (18)$$

and the following minimization task

$$\int_0^\Theta |\varepsilon(t)|^2 dt \rightarrow \min \quad (19)$$

can be established on the understandings, that the signal values are known only at sampling points, and the reconstructed signal is defined by (17). The problem (19) has to be resolved with respect to the coefficients $\{X_m\}$. Two approaches can be considered: the first one is based on setting up the continuous time signal by interpolation of known samples, while the second approach, which minimizes the continuous time reconstruction error, is based on the interpolation of error samples.

4.1 Signal interpolation

If signal samples $\{x_n\}$ are interpolated within the time interval $[0, \Theta]$, the problem (19) can be rewritten as

$$\int_0^\Theta \left| \tilde{x}(t) - \sum_m X_m \exp(j2\pi t f_m) \right|^2 dt \rightarrow \min, \quad (20)$$

where $\tilde{x}(t)$ is the interpolated signal. To find the minimum, all the individual derivatives of X_m have to be considered as being equal to zero. Taking into account that $\{\exp(j2\pi f_m t)\}$ is a set of orthogonal functions into interval $[0, \Theta]$ if frequencies $f_m = m/\Theta$, after some algebra the following formula for $X_m^{(x)}$ ($^{(x)}$ denotes that signal samples are interpolated) estimation can be obtained:

$$X_m^{(x)} = \frac{1}{\Theta} \int_0^\Theta \tilde{x}(t) \exp(-j2\pi t f_m) dt. \quad (21)$$

The expression (21) is similar to the formula for the calculation of the Fourier series coefficients for signal $\tilde{x}(t)$.

Signal interpolation can easily be done by connecting the samples with polynomials $p_n^k(t)$ of order k as $\tilde{x}(t) = \sum_n p_n^k(t)$, or a band-limited interpolation can be performed using a sum of time-shifted sinc functions.

If signal samples are interpolated with zero-order polynomials (piece-wise constant line changing value at midpoints between samples):

$$\begin{aligned} X_m^{(x_0)} &= \sum_{n=1}^N x_n \int_{\frac{t_n+t_{n-1}}{2}}^{\frac{t_n+t_{n+1}}{2}} \exp(j2\pi f_m t) dt \\ &= \frac{j}{2\pi f_m} \sum_{n=1}^N x_n \exp(j2\pi f_m t_n) \\ &\quad \cdot (1 - \exp(-j2\pi f_m \Delta t'_n)), \end{aligned} \quad (22)$$

where $\Delta t'_n = (t_{n+1} - t_{n-1})/2$, $t_0 = 0$, $t_{N+1} = \Theta$.

For piece-wise linear interpolation the polynomial $p_n^1(t) = \alpha_n(t - t_n) + x_n$ can be used, where $\alpha_n = \Delta x_n / \Delta t_n$, $\Delta x_n = x_n - x_{n-1}$, $\Delta t_n = t_n - t_{n-1}$, which gives:

$$\begin{aligned} X_m^{(x_1)} &= X_m^{(x_0)} + \frac{1}{(2\pi f_m)^2} \sum_{n=1}^N \alpha_n \exp(j2\pi f_m t_n) \\ &\quad \cdot (1 - \exp(-j2\pi f_m \Delta t_n)) + \frac{j}{2\pi f_m} \sum_{n=1}^N \alpha_n \Delta t_n \\ &\quad \cdot \exp(j2\pi f_m t_n) \exp(-j2\pi f_m \Delta t_n). \end{aligned} \quad (23)$$

Band-limited interpolation of samples can be described as:

$$\tilde{x}^{(sinc)}(t) = \sum_{k=0}^K c_k \text{sinc}(2\Omega t - k). \quad (24)$$

In this case DFT transform gives:

$$X_m^{(sinc)} = \sum_{k=0}^K c_k \exp(-j\pi f_m k / \Omega), \quad (25)$$

where c_k are coefficients that can be found from a linear equation system

$$x_n = \sum_{k=0}^K c_k \text{sinc}(2\Omega t_n - k). \quad (26)$$

Such an approach, besides the complexity of DFT, also requires the solution of linear system with N equations and with $K + 1$ unknowns. Interpolation by sinc functions can be effectively done for the stationary signal and if the gaps between samples do not exceed $1/2\Omega$. In this case the appropriate width of function can be fixed. However, for the non-stationary signal, the sinc functions should be stretched and time-shifted in accordance with instantaneous signal bandwidth and local sampling density.

4.2 Error interpolation

Like the interpolation of signal samples, the continuous-time reconstruction error function $\tilde{\varepsilon}(t)$ can be constructed from its values $\varepsilon_n = x_n - \hat{x}_n$, and the problem (19) can be interpreted as minimization of area under the function $|\tilde{\varepsilon}(t)|^2$.

Using zero-order polynomial interpolation the minimization task becomes:

$$\sum_{n=1}^N \left| x_n - \sum_m X_m^{(\varepsilon_0)} \exp(j2\pi f_m t_n) \right|^2 \Delta t'_n \rightarrow \min. \quad (27)$$

After the derivation and some algebra the solution can be expressed in the matrix form:

$$\mathbf{X}^{(\varepsilon_0)} = \mathbf{\Psi} \mathbf{x} (\mathbf{\Phi} \mathbf{\Psi}^T)^{-1}, \quad (28)$$

where $\phi_{mn} = \exp(j2\pi f_m t_n)$, $\psi_{mn} = \phi_{mn} \Delta t'_n$, and \cdot^T, \cdot^{-1} denotes the transpose and inverse operation of matrix respectively.

The first-order polynomial interpolation of error samples provides the problem, which looks like a sum of two zero-order interpolation tasks:

$$\frac{1}{2} \left(\sum_{n=1}^{N-1} |\varepsilon_n|^2 \Delta t_n + \sum_{n=2}^N |\varepsilon_n|^2 \Delta t_{n-1} \right) \rightarrow \min. \quad (29)$$

The solution is similar to the expression (28):

$$\mathbf{X}^{(\varepsilon_1)} = (\mathbf{\Psi}' \mathbf{x}' + \mathbf{\Psi}'' \mathbf{x}'') (\mathbf{\Phi}' \mathbf{\Psi}'^T + \mathbf{\Phi}'' \mathbf{\Psi}''^T)^{-1}, \quad (30)$$

where $\mathbf{\Phi}'$, $\mathbf{\Psi}'$, \mathbf{x}' and $\mathbf{\Phi}''$, $\mathbf{\Psi}''$, \mathbf{x}'' matrices are formed from $\mathbf{\Phi}$, $\mathbf{\Psi}$, \mathbf{x} by using indexes $n' = 1, N-1$ and $n'' = 2, N$ respectively.

5 PROPOSED APPROACH

The proposed approach is based on the same time windowing principle as in the STFT case. However, instead of general DFT more sophisticated methods are used, which have been described in the Section 4. Enhanced algorithms have increased mathematical complexity, particularly the error interpolation case, because the solving of linear system with N equations and M unknowns is required. M represents a number of frequencies in the Fourier series. The equation system can be solved correctly, if the number of samples is equal or greater than the number of frequencies. The greater the N/M ratio, the higher the stability of the solution. It has been shown, that, using the level-crossing sampling approach, the number of samples depends on the signal properties. Relationships between the local sampling density and the instantaneous upper spectral frequency of signal have

been derived. Performing the time-frequency analysis, these interdependencies can be exploited from an other point of view. The bandwidth of analysis can be limited using information about the local sampling density. The number of frequencies, as well as the dimensions of matrices vary with the time. For simulations, which will follow in the next section, the analysis bandwidth is selected as a minimum value of two frequencies: total bandwidth Ω or highest signal frequency estimated from the sampling density:

$$\Omega_a(t) = \min \left(\frac{N_w(t)}{2R_\Delta T_w} + \Omega_\Delta, \Omega \right), \quad (31)$$

where $N_w(t)$ is the number of signal samples in the time interval with length T_w , and Ω_Δ is necessary to ensure the coverage of actual signal bandwidth. The frequencies of analysis are $f_m = m/\Theta : |f_m| \leq \Omega_a$.

6 SIMULATION RESULTS

The computer simulation has been carried out to demonstrate the performance of approaches, which have been developed for time-frequency analysis of data captured by level-crossings. As a test-signal a chirp has been selected, which in the first half of observation diminishes from middle frequency to low frequency region (down to the normalized frequency 0.05), while in the second half rises back to the normalized frequency 0.25. Seven quantization levels have been placed equidistantly to cover the input range of the test-signal. The observation time is $\Theta = 256$, and 536 samples in total are obtained.

Time-frequency representations calculated by STFT and WT approaches have been already illustrated in the Section 3 (see Fig.3 and Fig.4). Let us inspect the enhanced algorithms, which are based on interpolation (expressions (22) and (23)). Fig.5 shows TFR obtained in the case, where time windowed test-signal samples are interpolated by zero-order polynomials. The spurious components are attenuated, however the presence of them is still observable. If the first-order polynomials are used for interpolation, the result is slightly improved, but the complexity of calculations is higher. Fig.6 shows time-frequency representation of test-signal if error samples are interpolated by zero-order polynomials (28). The spurious harmonics are completely suppressed, however other artifacts are observable in the time region, where the frequency of chirp is low. Reduction of the instantaneous frequency results in the decreasing of local sampling density. As the grid of temporal analysis and length of time window $w(t)$ are fixed, the number of the significant samples can fall below the number of frequencies of analysis. Such a situation causes the problem of

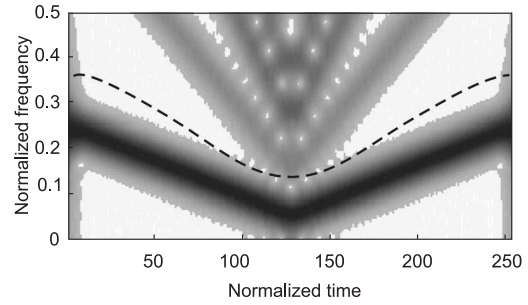


Figure 5: STFT approach in combination with zero-order interpolation of signal samples (dashed line shows instantaneous bandwidth estimate from local sampling density).

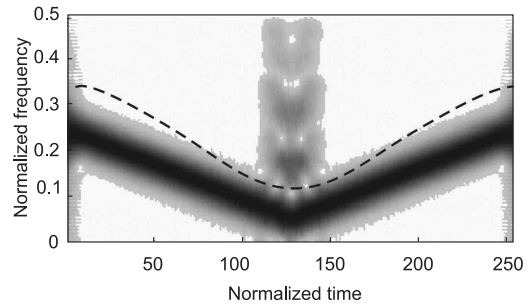


Figure 6: STFT approach in combination with zero-order interpolation of error samples (dashed line shows instantaneous bandwidth estimate from local sampling density).

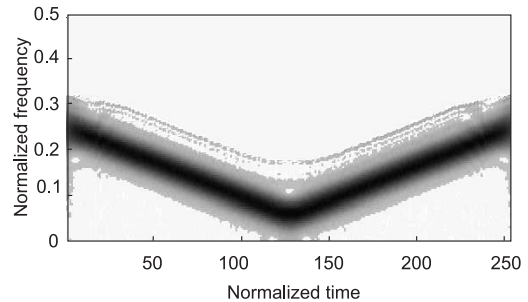


Figure 7: TFR of test-signal if approach of varying the range of analysis is used.

matrix inversion quality and leads to the appearance of artifacts. The use of interpolation by first-order polynomials does not have an impact on this effect.

To improve the quality of TFR in the region, where sampling density is low, the bandwidth of analysis has been cut down according to the expression (31). The estimated bandwidth of signal is illustrated in Fig.6 and Fig.5 by dashed line ($\Omega_\Delta = 0.1$). The coherence between the sampling density of a signal and the frequency range of an analysis gives several benefits - the stability of the algorithm is increased, the complexity of calculations is decreased and the presence of artifacts is eliminated. Fig.7 demonstrates the

time-frequency representation obtained by the algorithm based on the expression (28) in the case, where the number of analysis frequencies are varied according to the sampling density. The chirp can be tracked without any presence of artifacts.

7 CONCLUSION

The processing of non-stationary signal using level-crossing sampling approach has been investigated. On the one hand, such a sampling strategy provides several interesting properties - signal to quantization noise ratio does not depend on the number of quantization bits, local sampling density reflects the instantaneous bandwidth of signal, etc. On the other hand, the captured samples are placed non-uniformly and that requires rethinking of the processing methodology. The classical approaches of time-frequency analysis have been discussed. Time-frequency representations have been obtained using general forms of them, which are suitable also for processing of non-uniformly sampled signals. The simulation shows that the main drawback of STFT is the appearance of spurious components, while wavelet transform gives low spectral resolution at high frequencies and low temporal resolution at low frequencies.

Several enhancements have been proposed, which are based on the idea of minimizing the error between the original signal and that reconstructed by the Fourier series, not only at sampling time instants, but also between them with the same accuracy. The problem lies in the fact that the original signal values are known only at sampling instants. One solution is based on the consideration, that the continuous time signal is constructed by interpolation of known signal samples. The expressions for zero-order and first order polynomial interpolation as well as for band-limited interpolation with sinc functions have been established. The other approach is to interpolate the error samples in the same manner.

Simulation results show the improvement of TFRs if enhanced algorithms are used instead of classical ones. Additional benefits can be gained if the bandwidth of analysis is varied along the time axis according to changes in local sampling density: the artifacts are removed, the complexity of calculations can be decreased. The common drawback of STFT based methods is restrictions on the resolution. Extension of the windows $w(t)$ length improves the frequency resolution but at the same time degrades the temporal selectivity. To overcome this rule, the signal-dependent transformation described in (Greitans, 2005) can be used. Due to the limited size of the paper, this method is not discussed above, however the TFR obtained by signal-dependent algorithm is shown in the Fig.8 for

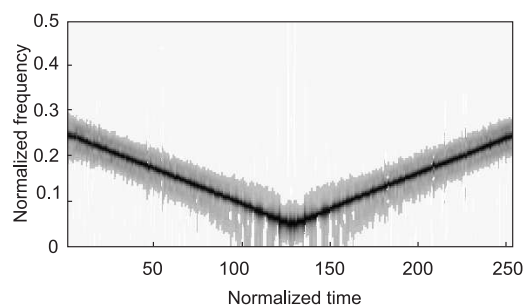


Figure 8: TFR of test-signal if signal-dependent transformation is used.

the illustration. The increased resolution is achieved by adapting the transformation functions to the local spectral characteristics of the signal. As it is being done in an iterative way, the mathematical complexity is higher than for STFT based algorithms.

The proposed approach of processing non-stationary signals using level-crossing sampling is attractive for clock-less designs, which are now receiving increasing interest. Their advantages can play a significant role in future electronics' development.

REFERENCES

- Akay, M., editor (1998). *Time frequency and wavelets in biomedical signal processing*. IEEE Press.
- Baraniuk, R. G. and Jones, D. L. (1993). A signal-dependent time-frequency representation: Optimal kernel design. *IEEE Trans. Signal Proc.*, 41(4):1589–1602.
- Chui, C. K. (1992). *Wavelet Analysis and its Applications*. Academic Press, Boston, MA.
- Cohen, L. (1995). *Time-frequency analysis*. Prentice-Hall.
- E. Allier, G. Sicard, L. F. and Renaudin, M. (2003). A new class of asynchronous a/d converters based on time quantization. In *Proc. of International Symposium on Asynchronous Circuits and Systems ASYNC'03*, pages 196–205, Vancouver, Canada.
- Ellis, P. H. (1959). Extension of phase plane analysis to quantized systems. *IRE Transactions on Automatic Control*, AC(4):43–59.
- Gabor, D. (1946). Theory of communication. *Journal of the IEE*, 93(3):429–457.
- Greitans, M. (2005). Spectral analysis based on signal dependent transformation. In *Proc. of the International Workshop on Spectral methods and multirate signal processing*, pages 179–184, Riga, Latvia.
- Hauck, S. (1995). Asynchronous design methodologies: An overview. *Proc. of the IEEE*, 83(1):69–93.
- Mark, J. W. and Todd, T. D. (1981). A nonuniform sampling approach to data compression. *IEEE Trans. on Comm.*, 29(1):24–32.

6. PIELIKUMS. MAĢISTRA DARBA ANOTĀCIJA. R. ŠĀVELIS „SIGNĀLA LAIKFORMAS ATJAUNOŠANA NO LĪMEŅU ŠĶĒRSOJUMA NOTIKUMU NOLASĒM”.

• ANOTĀCIJA

Signālu atjaunošana no nolasēm, kas iegūtas signālatkarīgu analogs-ciparu pārveidojumu rezultātā, ir aktuāla problēma asinhronās jeb beztaktēšanas sistēmās. Pie šādiem signālatkarīgiem pārveidojumiem pieder darbā apskatītais līmeņu-šķērsojuma analogs-ciparu pārveidojums, kura rezultāts ir laikā nevienmērīgi izvietotas signāla nolases. Darba mērķis ir apgūt dažādas signāla atjaunošanas metodes, to priekšrocības un trūkumus. Mērķa sasniegšanai tiek:

- Īsumā apskatīti analogs-ciparu pārveidojumi,
- veikts līdz šim jau publicēto signāla atjaunošanas metožu principu apskats
- piedāvāts jauns signāla atjaunošanas paņēmiens uzlabotās *sinc* funkciju metodes veidā,
- veikta signāla atjaunošanas simulācija *MATLAB* vidē ar visām iepriekš apskatītajām metodēm.

Darba nobeigumā sniegts iegūto rezultātu novērtējums un secinājumi. Darba rezultātus var pielietot signālatkarīgā datu apstrādē, turklāt apskatītā uzlabotā *sinc* funkciju metode ir labs aizsākums turpmākajiem pētījumiem.

Darbā: 70 lpp. teksts, 56 attēli, 2 tabulas, 5 pielikumi un 16 nosaukumu informācijas avoti.

7. PIELIKUMS. A

```
%*****  
% Funkcijas, kas paredzeeta dazhaadu signaalu genereeshanai  
%*****  
function Uno=noise(t, amp, f, ph)  
  
U=[];  
for vv=1:length(t)  
    usig=sum(amp.*sin(2*pi*f*t(vv)+ph)); % katram mazajam gabalinjam tiek  
apreekjinaatas signaala vertiibas  
    U=[U usig+0.5];  
end;  
%Uno=U/max(abs(hi), abs(lo));  
Uno=U/sqrt(length(f));  
%hold on;  
%plot(Uno);
```

```

clc;
close all;
clear all;

amp=1; %amplituda, V
T=30; % periodu skaits peetaama signaala garumaa.
TH=128; % signaala garums
a=T/TH;% trokshveida signaala joslas patums, Hz

%b=0.001;% frekvences pieaugums, Hz/s
qbit=4; % kvantešanas bitu skaits, gab
fsk=TH*4;
rand('state', 1);
ph=rand(1,fsk)*2*pi;
dt=1024;
f=linspace(0, a, fsk);
signal='noise(t, amp, f, ph)';

run_level_crossing

%*****
% LEVEL VALUE HISTOGRAMM
%*****
lvd=[];
cnt=[];
for ss=1:length(lv)
    lvd=[lvd lv(ss)];
    cnt=[cnt length(lv3(lv3==lv(ss)))];
end;
%set(gca, 'xlim', [0 TH]);
figure;
stem(lvd, cnt, '.', 'k');
title('Sample density over levels');
xlabel('level value');
ylabel('pts');

%*****
% THEORETICAL LEVEL VALUE HISTOGRAMM
%*****
N=[];
for ii=1:length(lv)
    E=((2*max(f)/sqrt(3))*TH)*exp((( -lv(ii)^2)/4*amp*max(f))*sqrt(TH));
    N=[N E];
end;
hold on;
plot(lv, N);

%*****
% trokshnja jauda
%*****
S=0;
lve=lv3.^2;

Preg=1/TH*sum(lve)

for xx=1:length(t3)-1
    dt=(t3(xx+1)-t3(xx))/2;
    s=lve(xx)*dt+lve(xx+1)*dt;

```

```

        S=S+abs(s);
end;
1/TH*S

%*****
% delta t histogrammas apreekjins
%*****
dtt=[];
for xx=1:length(t3)-1
    a1=t3(xx+1)-t3(xx);
    dtt=[dtt a1];
end;

figure;
[x,y]=hist(dtt, linspace(min(dtt), max(dtt), 100));
stem(y,x, 'k', '.');
title('delta t hist');
xlabel('time, s');
ylabel('pts');

%*****
% QUANTIZATION ERROR
%*****

figure;
stem(t3, lvdu, '.', 'k');
set(gca, 'xlim', [0 TH]);
title('Quantization error');
xlabel('time, s');
ylabel('quantization level');

figure;
x2=histc(lvdu, linspace(min(lvdu), max(lvdu), 100));
stem(x2, 'k', '.');
title('Quantization error hist');
xlabel('quantization level');
ylabel('pts');
set(gca, 'xtick', [0 length(x2)/2 length(x2)]);
set(gca, 'xTickLabelMode', 'manual');
set(gca, 'xticklabel', ['-T/2'; ' 0 '; ' T/2']);

```

```

%*****
% Limenju-shkjersojshais ACP modelis
%*****
U=[];
hi=0;
lo=0;
%for xx=1:TH
    t=linspace(0,TH-1,fsk);

    U=eval(signal);
    %usig=sum(amp.*sin(2*pi*f*t(xx)+ph)); % katram mazajam gabalinjam tiek
apreekjinaatas signaala vertiibas
    %U=[U usig];
    %end;

Ulidz=1/TH*sum(U);
if hi<max(U)
    hi=max(U);
end;
if lo>min(U)
    lo=min(U);
end;

R=max(abs(U));
deltaR=(R*2)/(2^qbit-1);
lv=[];
lvx=(-R)-deltaR/2;
lv=lvx;
for xx=0:2^qbit-1
    lvx=lvx+deltaR;
    lv=[lv lvx];
end;
%lv=[lv(1)-deltaR lv lv(length(lv))+deltaR];
%lv=lv+deltaR/2
% lv=linspace(lo, hi, 2.^qbit);
% %lv=[-1 -0.75 -0.25 0.25 0.75 1];
% % tiek panjemas ari galejas vertibas.
% qlev=(max(lv)-min(lv))/(length(lv)-1);
% lv=[lv(1)-qlev lv lv(length(lv))+qlev];
% lv=lv+qlev/2;
%*****
% ACP modelis
%*****

lv3=[];
t3=[];
lvdu=[];
for xx=0:TH-1; % Signaals tiek sadaliits 128 mazos gabalinjos
    t=linspace(xx, xx+1, dt);
    %f=b*t+a;
    %*****
    % Generee signaalu
    %*****
    U=eval(signal);
    %U=amp*sin(2*pi*f.*t+ph); % katram mazajam gabalinjam tiek apreekjinaatas
signaala vertiibas
    hold on;
    plot(t, U);
    %*****
    for jj=1:dt;
        for ii=1:length(lv)-1;

```

```

if ((U(jj)>=lv(ii)) && (U(jj)<lv(ii+1)))
    if jj==1
        lv2=lv(ii);
        lv21=lv(ii+1);
    else
        end;
    if ((U(jj)>=lv2) && (U(jj)<lv21))

else
    if (U(jj)<lv2) % krusto apaksheejo liiniju
        if U(jj)>lv2
            elseif lv2==U(jj)
                t3=[t3 t(jj)]; %(kk*M)
                lv3=[lv3 lv2];
                lvdu=[lvdu lv2-U(jj)];
                lv2=lv(ii);
                lv21=lv(ii+1);
                break;
            else
                hiv=(U(jj-1)/lv2)-1;
                lov=(lv2/U(jj))-1;
                if hiv>lov
                    t3=[t3 t(jj-1)];
                    lv3=[lv3 lv2];
                    lvdu=[lvdu lv2-U(jj)];
                    lv2=lv(ii);
                    lv21=lv(ii+1);
                    break;
                else
                    t3=[t3 t(jj)];
                    lv3=[lv3 lv2];
                    lvdu=[lvdu lv2-U(jj)];
                    lv2=lv(ii);
                    lv21=lv(ii+1);
                    break;
                end;
            end;
        else % krusto augsheejo liiniju. Signaala virziens augosh
            if U(jj)<lv21
                elseif U(jj)==lv21
                    t3=[t3 t(jj)];
                    lv3=[lv3 lv21];
                    lvdu=[lvdu lv21-U(jj)];
                    lv2=lv(ii);
                    lv21=lv(ii+1);
                    break;
                else
                    hiv=(U(jj)/lv21)-1;
                    lov=(abs(lv21)/abs(U(jj)))-1;
                    if hiv>lov
                        t3=[t3 t(jj-1)];
                        lv3=[lv3 lv21];
                        lvdu=[lvdu lv21-U(jj)];
                        lv2=lv(ii);
                        lv21=lv(ii+1);
                        break;
                    else
                        t3=[t3 t(jj)];
                        lv3=[lv3 lv21];
                        lvdu=[lvdu lv21-U(jj)];
                        lv2=lv(ii);
                    end;
                end;
            end;
        end;
    end;
end;

```

```
lv21=lv(ii+1);
break;
end;
end;
end;
end;
U2=U(jj);
lv2=lv(ii);
lv21=lv(ii+1);
end;
end;
end;
end;
hold on;
plot (t3, lv3, '-r');
set(gca, 'xlim', [0 TH]);
title('Signal');
xlabel('time, s');
ylabel('amplitude, V');
```

7. PIELIKUMS. B

DATORPROGRAMMA SIGNĀLA LĪMEŅU-ŠĶĒRSOJUMA NOLAŠU IEGŪŠANAI *MATLAB* VIDĒ

```
function
[uu,tt]=lvcrossing (TH,dt,Sigt,N,R)

%TH - uzdotā signāla beigu laiks
u(1)=-R/2; u(2^N)=R/2;
%dt - signāla nolašu sekošanas solis
for k=2:2^N-1
%Sigt - uzdotā signāla izteiksme
    u(k)=u(k-1)+R/(2^N-1);
%N - bitu skaits līmeņu kodēšanai
end;
%R - AACP dinamiskais diapazons

%uu - līmeņu-šķērsojuma nolases
tt=[]; uu=[]; t=0;
%tt - nolašu laika momenti
s1=eval(Sigt);
for k=1:length(u)-1
    if s1==u(k)
        p=k; uu=[uu, s1]; tt=[tt,
t]; break;
    elseif s1<u(k+1)
        p=k; break;
    end;
end;

if s1==u(length(u))
    uu=[uu, s1]; tt=[tt, t];
p=length(u);
end;

for ti=0:0.1:TH-0.1
    t=ti+dt:dt:ti+0.1+dt;
    S=[s1, eval(Sigt)];
    for k=1:length(t)-1
        if p==length(u)
            b=p;
        else
            b=p+1;
        end;
        if S(k+1)>=u(b)
            tt=[tt, t(k)]; uu=[uu,
u(b)];
            p=b;
        elseif
S(k+1)>S(k) && S(k+2)<S(k+1) && u(b) -
S(k+1)<0.1*dt;
            tt=[tt, t(k)]; uu=[uu,
u(b)];
        elseif
S(k+1)<S(k) && S(k+2)>S(k+1) && S(k+1) -
u(p)<0.1*dt;
            tt=[tt, t(k)]; uu=[uu,
u(p)];
        elseif S(k+1)<=u(p)
            if S(k+1)==p
                tt=[tt, t(k)];
uu=[uu, u(p)];
            elseif S(k)>u(p)
```

```
                tt=[tt, t(k)];
uu=[uu, u(p)];
            end;
            if p>1 && S(k+1)~u(p);
                p=p-1;
            end;
        end;
    end;
    s1=S(length(S)-1);
end;
%Funkcijas izsaukuma piemērs FM
signāla diskretizācijai:

[Sn,tn]=lvcrossing(256,1e-
4,'cos(2*pi*(0.05+0.2/256*t).*t)',2,
2);
plot(tn,Sn,'.')
```

DATORPROGRAMMA SIGNĀLA ATJAUNOŠANAI AR *SINC* FUNKCIJU METODI *MATLAB* VIDĒ

```
function Satj=SincRec(Sn,tn,t,fmax)

%Sn - signāla nolašu vektors
%tn - nolašu laika momentu vektors
%t - atjaunotā signāla laika vektors
%fmax - loga maksimālā frekvence
%Satj - atjaunotā signāla vērtību
vektors

Satj=zeros(1,length(t));
dt=1;

for k=1:length(tn);

Satj=Satj+Sn(k)*2*fmax*dt.*sinc(2*pi
*fmax*(t-tn(k)));
end;

%Funkcijas izsaukuma piemērs laikā
vienmērīgi diskretizēta FM signāla
%atjaunošanai:

t=0:1:256;
y=cos(2*pi*(0.05+0.2/256*t).*t);
tt=0:0.1:256;
Satj=SincRec(y,t,tt,0.5);
plot(tt,Satj)

%Funkcijas izsaukuma piemērs
amplitūdā vienmērīgi diskretizēta FM
signāla %atjaunošanai:

[Sn,tn]=lvcrossing(256,1e-
5,'cos(2*pi*(0.05+0.2/256*t).*t)',2,
2);
Satj=SincRec(Sn,tn,tt,0.5);
plot(tt,Satj)
```

DATORPROGRAMMAS SIGNĀLA ATJAUNOŠANAI AR SPLAINU METODI MATLAB VIDĒ

```
function Satj=SplRec(Sn,tn,t)

%Sn - signāla nolases
%tn - nolašu laika momenti
%Satj - atjaunotais signāls kā laika
t funkcija
```

```
Satj=zeros(size(t));
```

```
for k=1:length(tn)
    Satj=Satj+Sn(k)*CardSpl(t-
tn(k),[-10:1:10]);
end;
```

```
%Funkcija CardSpl paredzēta kubiskā
pamat-splaina aprēķinam
```

```
function n=CardSpl(t,p)
```

```
%t - laika vērtību vektors
%p - pakāpju vektors
%n - pamat-splaina vērtības
```

```
p1=p(length(p));
tt=t(t>=-p1-2&t<=p1+2);
n1=zeros(size(tt));
z1=sqrt(3)-2;
```

```
for k=1:length(p)
    n1=n1+6*z1/(z1^2-
1)*z1^abs(p(k))*CubSpl(tt-p(k));
end;
```

```
n=[zeros(size(t(t<-p1-2))), n1,
zeros(size(t(t>p1+2)))];
```

```
%Funkcija CubSpl paredzēta kubiskā
B-splaina aprēķinam
```

```
function b=CubSpl(t)
```

```
%t - laika vērtību vektors
%b - splaina vērtību vektors
```

```
b1=0*t(t<=-2);
b2=((2-abs(t(t<=-1&t>-2))).^3)/6;
b3=(4-6*abs(t(t<=0&t>-
1)).^2+3*abs(t(t<=0&t>-1)).^3)/6;
b4=(4-
6*abs(t(t>0&t<1)).^2+3*abs(t(t>0&t<1
)).^3)/6;
b5=((2-abs(t(t>=1&t<2))).^3)/6;
b6=0*t(t>=2);
b=[b1 b2 b3 b4 b5 b6];
```

```
%Funkcijas izsaukuma piemērs
amplitūdā vienmērīgi diskretizēta FM
signāla %atjaunošanai:
```

```
[Sn,tn]=lvcrossing(256,1e-
5,'cos(2*pi*(0.05+0.2/256*t).*t)',2,
2);
t=0:0.1:256;
Satj=SplRec(Sn,tn,t);
plot(t,Satj)
```

DATORPROGRAMMAS SIGNĀLA ITERATĪVAI ATJAUNOŠANAI MATLAB VIDĒ

```
function
[t,Siz_i]=ItRec(tn,Sn,it_skaits,metode)
```

```
%Sn - signāla nolases
%tn - nolasēm atbilstošie laika
momenti
%it_skaits - uzdotais iterāciju
skaits
%metode - uzdotā metode
%Siz_i - atjaunotais signāls kā
laika t funkcija
```

```
dt=0.1; TH=256; fmin=0; fmax=0.5;
t=0:dt:TH;
Siz=zeros(size(t));
Siz_i=zeros(it_skaits,length(t));
```

```
for k=1:it_skaits;
    if strcmp(metode,'Vor')==1;
        Sa=Vor_apr(Sn,tn,TH,dt);
        tt=t;
    elseif strcmp(metode,'Lin')==1;
        Sa=Lin_apr(Sn,tn,TH,dt);
        tt=t;
    elseif strcmp(metode,'Marv')==1;
        Sa=Marv_apr(Sn,0.3);
        dt=1;
        tt=tn;
    elseif
    strcmp(metode,'Adkoef')==1;
        Sa=Ad_koef_apr(Sn,tn,0.5);
        dt=1;
        tt=tn;
    end;
```

```
[f,FT]=furjtrans(Sa,tt,dt,fmin,fmax);
```

```
;
    S0t=invfurjtrans(FT,f,t);
    S0n=invfurjtrans(FT,f,tn);
    Siz=Siz+real(S0t);
    Siz_i(k,:)=Siz;
    Sn=Sn-real(S0n);
end;
```

```
%Funkcija Vor_apr paredzēta signāla
aproximācijai ar taisnstūra
funkcijām.
```

```
function u=Vor_apr(U,T,TH,dt)
```

```
%U - signāla nolases
%T - nolašu laika momenti
%TH - signāla beigu laiks
```

```
%dt - aproksimācijas signāla laika
diskretizācijas solis
%u - aproksimācijas signāls
```

```
tvid=zeros(length(T)-1);
for k=1:length(T)-1;
    tvid(k)=(T(k)+T(k+1))/2;
end;
```

```
t=0:dt:TH;
u=zeros(size(t));
a=1;s=1;
for k=1:length(tvid);
    for n=a:length(t);
        if t(n)<=tvid(k);
            u(n)=U(k);
            s=s+1;
        elseif
        t(n)>tvid(length(tvid));
            u(n)=U(length(U));
        elseif t(n)>tvid(k)
            break;
        end;
    end;
    a=s;
end;
```

```
%Funkcija Lin_apr paredzēta signāla
aproximācijai ar lineārām
funkcijām.
```

```
function u=Lin_apr(U,T,TH,dt)
```

```
%U - signāla nolases
%T - nolašu laika momenti
%TH - signāla beigu laiks
%dt - aproksimācijas signāla laika
diskretizācijas solis
%u - aproksimācijas signāls
```

```
t=0:dt:TH;
u=zeros(size(t));
b=length(T);
```

```
for k=1:b-1;
    tga(k)=(U(k+1)-U(k))/(T(k+1)-
T(k));
end;
if t(length(t))-T(b)>0
    tga=[tga, U(b)/2/(t(length(t))-
T(b))];
end;
```

```
a=1;s=1;
for k=1:length(T)-1;
    for n=a:length(t);
        if t(n)<=T(k+1);
            u(n)=U(k)+(t(n)-
T(k))*tga(k);
            s=s+1;
        elseif t(n)>T(k+1)
            break;
        end;
```

```

elseif t(n)>T(b);
    u(n)=U(b)+(t(n)-
T(b))*tga(b);
end;
end;
a=s;
end;

```

%Funkcija **Marv_apr** paredzēta signāla aproksimācijai ar vienības nolasēm.

```
function u=Marv_apr(U,lamda)
```

```

%U - signāla nolases
%lamda - globālais relaksācijas
koeficients
u% - aproksimācijas signāls

```

```
u=lamda*U;
```

%Funkcija **Ad_koef_apr** paredzēta signāla aproksimācijai ar vienības nolasēm.

```
function u=Ad_koef_apr(U,T,gamma)
```

```

%U - signāla nolases
%T - nolašu laika momenti
%gamma - koeficients
u% - aproksimācijas signāls

```

```
dt=zeros(size(T));
dt(1)=T(2)-T(1);
```

```

for k=2:length(T)-1;
    dt(k)=(T(k+1)-T(k-1))/2;
end;

```

```
dt(length(T))=T(length(T))-T(k);
u=gamma*U.*dt;
```

%Funkcija **furjtrans** paredzēta signāla vienpusīgā spektra iegūšanai

```
function
[f,FT]=furjtrans(Sn,tn,dt,fmin,fmax)
```

```

%Sn - signāla nolases
%tn - nolašu laika momenti
%dt - diskretizācijas solis
vienmērīgas diskretizācijas
gadījumam, ja nolases %ir
nevienmērīgi izvietotas, tad dt=1
%fmin - aprēķināmā spektra minimālā
frekvence
%fmax - spektra maksimālā frekvence
%FT - spektra kompleksās vērtības
atkarībā no frekvences f

```

```
f=fmin:1/300:fmax;
```

```
FT=zeros(size(f));
```

```

for k=1:length(f);
    for n=1:length(Sn)
        FT(k)=FT(k)+Sn(n)*exp(-
j*2*pi*f(k)*tn(n));
    end;
end;
FT=2*dt*FT;

```

%Funkcija **invfurjtrans** paredzēta signāla aprēķinam no tā spektra, kas iegūts %ar funkciju **furjtrans**.

```
function s=invfurjtrans(FT,f,t)
```

```

%FT - spektra kompleksās vērtības
atkarībā no frekvences f
%s - aprēķinātais signāls atkarībā
no laika t

```

```
s=zeros(size(t));
```

```

for k=1:length(t);
    for n=1:length(f)

```

```
s(k)=s(k)+FT(n)*exp(j*2*pi*f(n)*t(k));
```

```

end;
end;
s=1/300*s;

```

%FM Signāla atjaunošanas piemērs no tā līmeņu-šķērsojuma nolasēm ar visām %četrām metodēm:

```
[Sn,tn]=lvcrossing(256,1e-
5,'cos(2*pi*(0.05+0.2/256*t)).*t',2,
2);
```

```

[t,SatjV]=ItRec(tn,Sn,5,'Vor');
[t,SatjL]=ItRec(tn,Sn,5,'Lin');
[t,SatjM]=ItRec(tn,Sn,5,'Marv');
[t,SatjA]=ItRec(tn,Sn,5,'Adkoef');

```

```

plot(t,SatjV(1,:));
plot(t,SatjV(5,:));
plot(t,SatjL(5,:));
plot(t,SatjM(5,:));
plot(t,SatjA(5,:));

```

DATORPROGRAMMAS SIGNĀLA ATJAUNOŠANAI AR UZLABOTO SINC FUNKCIJU METODI MATLAB VIDĒ

%Funkcija **ImpSsincRec** paredzēta signāla atjaunošanai ar simetrisku *sinc* %funkciju metodi.

```
function Satj=ImpSsincRec(Sn,tn,t,M)
```

```
%Sn - signāla nolases
%tn - nolasēm atbilstošie laika
momenti
%M - nolašu skaits lokālā nolašu
blīvuma aprēķinā
%Satj- atjaunotais signāls kā laika
t funkcija
```

```
omega=zeros(1,length(tn));
Avdt=Averdt1(tn,2*M);
```

```
for k=1:length(tn);
    omega(k)=pi/Avdt(k);
end;
```

```
OMEGA=zeros(length(tn),length(tn));
```

```
for k=1:length(tn)
    for n=1:length(tn)
        OMEGA(n,k)=sinc(omega(k)*(tn(n)-
tn(k)));
    end;
end;
```

```
C=OMEGA\Sn';
Satj=zeros(size(t));
```

```
for k=1:length(C)
    Satj=Satj+C(k)*sinc(omega(k)*(t-
tn(k)));
end;
```

%Funkcija **Averdt1** paredzēta lokālā nolašu blīvuma aprēķinam.

```
function Avdt=Averdt1(tn,numb)
```

```
%tn - nolašu laika momenti
%numb - nolašu skaits 2*M
%Avdt- sinc funkciju galvenās vāles
pus platumi
```

```
Avdt=zeros(size(tn));
```

```
for k=1:numb
```

```
    Avdt(k)=(tn(k+numb)-
tn(1))/(numb+k-1);
end;
```

```
d=2*numb-1;
for k=length(tn)-numb+1:length(tn)
    Avdt(k)=(tn(length(tn))-tn(k-
numb))/d;
    d=d-1;
end;
```

```
for k=numb+1:length(tn)-numb
    Avdt(k)=(tn(k+numb)-tn(k-
numb))/(2*numb);
end;
```

%Funkcija **ImpAsincRec** paredzēta signāla atjaunošanai ar nesimetrisku *sinc* %funkciju metodi.

```
function Satj=ImpAsincRec(Sn,tn,t,M)
```

```
%Sn - signāla nolases
%tn - nolasēm atbilstošie laika
momenti
%M - nolašu skaits lokālā nolašu
blīvuma aprēķinā
%Satj- atjaunotais signāls kā laika
t funkcija
```

```
for k=1:length(tn)
    [dtl, dtr]=Averdt2(tn(k),tn,M);
    for n=1:length(tn)
        if tn(n)<=tn(k)
```

```
            sc(n,k)=sinc(pi/dtl*(tn(n)-tn(k)));
        else
```

```
            sc(n,k)=sinc(pi/dtr*(tn(n)-tn(k)));
        end;
    end;
```

```
end;
```

```
C=sc\Sn';
```

```
Satj=zeros(size(t));
```

```
for k=1:length(tn);
```

```
    Satj=Satj+C(k)*AsSinc2(tn(k),tn,M,t)
;
end;
```

%Funkcija **Averdt2** paredzēta lokālā nolašu blīvuma aprēķinam nesimetrisku *sinc* %funkciju gadījumā

```
function [dtl,
dtr]=Averdt2(tk,tn,numb)
```

```
%tk - nolases laika moments,
attiecībā pret kuru rēķina nolašu
blīvumu
%tn - visu nolašu laika momenti
```

```

%numb - nolašu skaits lokālā nolašu
blīvuma aprēķinā
%dtl, dtr- kreisās un labās puses
sinc funkciju galvenās vāles pus
platumi

for k=1:length(tn)
    if tn(k)==tk
        ind=k;break;
    end;
end;

if ind>numb
    dtl=(tk-tn(ind-numb))/numb;
else
    if ind==1
        dtl=tn(2)-tn(1);
    else
        dtl=(tk-tn(1))/(ind-1);
    end;
end;
if ind<=length(tn)-numb
    dtr=(tn(ind+numb)-tk)/numb;
else
    if ind==length(tn)
        dtr=tn(length(tn))-
tn(length(tn)-1);
    else
        dtr=(tn(length(tn))-
tk)/(length(tn)-ind);
    end;
end;
%FM Signāla atjaunošanas piemērs no
tā līmeņu-šķērsojuma nolasēm ar abām
sinc %funkciju metodēm:

[Sn,tn]=lvcrossing(256,1e-
5,'cos(2*pi*(0.05+0.2/256*t).*t)',2,
2);
t=0:0.1:256;
SatjS=ImpSsincRec(Sn,tn,t,3);
plot(t,SatjS)

SatjA=ImpAsincRec(Sn,tn,t,3);
plot(t,SatjA)

%FM Signāla atjaunošanas piemērs no
tā laikā vienmērīgām nolasēm ar abām
sinc %funkciju metodēm:

t=0:1:256;
y=cos(2*pi*(0.05+0.2/256*t).*t);
tt=0:0.1:256;
SatjS=ImpSsincRec(y,t,tt,3);
plot(tt,SatjS)

SatjA=ImpAsincRec(y,t,tt,3);
plot(t,SatjA)

```

7. PIELIKUMS. C

```

% Nolases seciibas aproksimacija ar taisnstuuriem
% lai peec tam iegauto aproksimeeto funkciju apstradaatu ar Wignera
% sadalijuma integraali
clear all
close all
%***
delta=0.01; % aproksimācijas funkcijas konstatātais solis
time=20; % laika beigu moments
i=0; % matricu koeficients aproksimācijas laika un veertiību vektoriem
j=1; % matricu koeficients diskretai seciībai laika un veertiību vektoriem
tt=0:delta:time; % laika vektors
y=cos(2*pi*(0.05+0.2/256*tt).*tt); % saakuma funkcija
%Liimenju shkersojuma diskretizācija tiek viekta ar R.Shaavelja izveidotu
%funkciju lvcrossing. tn ir nolases laika moments, sn ir nolases
%amplituudas veertiība. Diskretizācijas precizitāte 10.e-4; ACP bitu
%skaits ir 4 un ACP amplituudu diapazons R=Umax-Umin ir 4
[sn,tn]=lvcrossing(20,1e-4,'cos(2*pi*(0.05+0.2/256*t).*t)',2,2);
% grafiskais atteelojums
plot(tt,y),hold on
stem(tn,sn,'r')
xlabel('laiks,t')
ylabel('x(t),sn[tn]')
xx=[]; % aproksimācijas funkcijas matrica
tn(:,length(tn)+1)=time+1; % laika momenta length(tn)+1 veertiība
sn(:,length(sn)+1)=1+1; % nolases length(sn)+1 veertiība
% ciklaa ir aprakstiits aproksimācijas process
for i=1:time/delta+1
    if tn(:,j+1)<=tt(:,i),
        j=j+1;
    end
    xx(:,i)=sn(:,j);
end
plot(tt,xx,'-.'), hold off

```



```
% Nolases seciibas aproksimacija ar taisnstuuriem, kad nolases vertiiba ir
% apmeeram pa vidu starp divam blakus esosham nolaseem
```

```
clear all
close all
```

```
***
```

```
delta=0.01; % aproksimācijas funkcijas konstatātais solis
time=20; % laika beigu moments
i=0; % matricu koeficients aproksimācijas laika un veertiību vektoriem
j=1; % matricu koeficients diskretai seciībai laika un veertiību vektoriem
tt=0:delta:time; % laika vektors
y=cos(2*pi*(0.05+0.2/256*tt).*tt); % saakuma funkcija
%Liimenju shkersojuma diskretizācija tiek viekta ar R.Shaavelja izveidotu
%funkciju lvcrossing. tn ir nolases laika moments, sn ir nolases
%amplituudas veertiiba. Diskretizācijas precizitāte 10.e-4; ACP bitu
%skaits ir 4 un ACP amplituudu diapazons R=Umax-Umin ir 4
[sn,tn]=lvcrossing(20,1e-4,'cos(2*pi*(0.05+0.2/256*t).*t)',2,2);
% grafiskais atteelojums
plot(tt,y),hold on
stem(tn,sn,'r')
xlabel('laiks,t')
ylabel('x(t),sn[tn]')
xx=[]; % aproksimācijas funkcijas matrica
tn(:,length(tn)+1)=time+1; % laika momenta length(tn)+1 veertiiba
sn(:,length(sn)+1)=1+1; % nolases length(sn)+1 veertiiba
% ciklaa ir aprakstiits aproksimācijas process
for i=1:time/delta+1
    if (tn(:,j+1)+tn(:,j))/2<=tt(:,i),
        j=j+1;
    end
    xx(:,i)=sn(:,j);
end
plot(tt,xx,'-.'), hold off
```

```
% Nolases seciibas aproksimacija ar liinijam, kurie savieno katru nolasi
% ar naakosho.
```

```
clear all
close all
```

```
***
```

```
delta=0.01; % aproksimācijas funkcijas konstatātais solis
time=20; % laika beigu moments
i=0; % matricu koeficients aproksimācijas laika un veertiību vektoriem
j=1; % matricu koeficients diskretai seciībai laika un veertiību vektoriem
tt=0:delta:time; % laika vektors
y=cos(2*pi*(0.05+0.2/256*tt).*tt); % saakuma funkcija
%Liimenju shkersojuma diskretizācija tiek veikta ar R.Shaavelja izveidotu
%funkciju lvcrossing. tn ir nolases laika moments, sn ir nolases
%amplituudas veertiība. Diskretizācijas precizitāte 10.e-4; ACP bitu
%skaits ir 4 un ACP amplituudu diapazons R=Umax-Umin ir 4
[sn,tn]=lvcrossing(20,1e-4,'cos(2*pi*(0.05+0.2/256*t).*t)',2,2);
% grafiskais attēlojums
plot(tt,y),hold on
stem(tn,sn,'r')
xlabel('laiks,t')
ylabel('x(t),sn[tn]')
xx=[]; % aproksimācijas funkcijas matrica
tn(:,length(tn)+1)=time+1; % laika momenta length(tn)+1 veertiība
sn(:,length(sn)+1)=1+1; % nolases length(sn)+1 veertiība
for i=1:time/delta+1
    if tn(:,j+1)<=tt(:,i),
        j=j+1;
    end
    xx(:,i)=(sn(:,j+1)-sn(:,j))/(tn(:,j+1)-tn(:,j))*(tt(:,i)-tn(:,j))+sn(:,j); %
shii formula ir iegūta no taisnās līnijas viennadojuma caur 2 punktiem
end
plot(tt,xx,'-.'), hold off
```

```
%Ar liimenja-skeersojuma principu diskretizeeta signaala vilnishanaliize  
%(Waveler analiize)
```

```
close all  
clear all
```

```
%Izveidojam laika momentu vektoru  
N=256; %nolasu skaits  
%signaals ir defineets laika intervalaa  
t=0:.01:256;  
%Piemeera signaals ir FM kosinusoida.  
%Signala veertiibu aprekins  
x=cos(2*pi*(0.05+0.2/256*t).*t);  
%Liimenju shkersojuma diskretizaacija tiek viekta ar R.Shaavelja izveidotu  
%funkciju lvcrossing. tn ir nolases laika moments, sn ir nolases  
%amplituudas veertiiba. Diskretizaacijas precizitaate 10.e-4; ACP bitu  
%skaits ir 4 un ACP amplituudu diapazons R=Umax-Umin ir 4  
[sn,tn]=lvcrossing(256,1e-4,'cos(2*pi*(0.05+0.2/256*t).*t)',4,4);  
%signaala, kaa laika funkcijas grafiska reprezentacija  
plot(t,x,tn,sn,'.')  
xlabel('laiks')  
ylabel('s(t)')  
%***  
%Wavelet'a formeeshana  
%***  
wave=sqrt(N)/4;  
fmin=1/N;  
fmax=0.5;  
f = logspace(log10(fmin),log10(fmax),N/2); % logaritmiskaa merogoshana  
frekvenceem  
f=[fliplr(-f) f];  
a = logspace(log10(fmax/fmin),log10(1),N/2); % logaritmiskaa merogoshana  
Wavelet'a a paremetram  
a=[fliplr(a) a];  
ta=0:1:N-1; %analizes laika momenti  
% cikls:  
for r=1:length(f); %mainam wavelet'a platumu  
    nha = wave*a(r); % => wave*[a(r(1)), a(r(2)), ..., a(r(length(f)))]  
    for tnob = 1:length(ta) %biidam wavelet'u  
        tt=tn-ta(tnob);  
        % tiek pielietots Morlet'a Wavelet  
        hal(tnob,:) = sqrt(nha).*exp(-  
(2*log(10)/nha^2)*(tt.^2)).*exp(j*2*pi*f(r)*tt);  
    end  
    plot(real(hal(50,:)));  
        ha2(r,:) =sn*hal'./sqrt(a(r)); %integrejam signaala un waveleta  
reizinajumu  
end  
  
ha3=(ha2).*conj(ha2);  
tfr=ha3;  
%Veidojam krasu gammu 3D bildem  
L=16; %limenu tonu skaits  
ac=[1-1/L 1-1/L 1-1/L];  
for ii=3*L/4:-1:1/L,  
    ac=[ac; [ii./L ii./L ii./L]];  
end  
% zimeejam rezultaatu  
figure  
mesh(ta, f, tfr);
```

```
colormap(ac);
set(gca, 'xlim', [0 length(ta)], 'fontsize', 14)
set(gca, 'ylim', [-0.5 0.5], 'fontsize', 14)
set(gca, 'zlim', [0 max(max(abs(tfr)))], 'fontsize', 14)
xlabel('Laiks', 'fontsize', 16)
ylabel('Frekvence', 'fontsize', 16)
view(0, 90);
N=length(f);
%set(gca, 'ytick', [0 .2*N 0.4*N 0.5*N 0.6*N .8*N N]);
%set(gca, 'yticklabel', [-0.5 -0.3 -0.1 0 0.1 0.3 0.5]);
set(gca, 'xgrid', 'off', 'ygrid', 'off', 'zgrid', 'off');
set(gca, 'ztick', [], 'zcolor', [1 1 1]);
set(gcf, 'color', [1 1 1]);
```

```

% MATLAB programma Iisintervala Furje transformaācijas aprekinashanai,
% gadiijumaa, kad signaals ir diskretizēts ar liimenja-shkersojuma
% principu

close all
clear all

%signaals ir defineets laika intervalaa
t=0:.01:256;
%Piemeera signaals ir FM kosinusoida.
%Signāla veertiibu aprekins
x=cos(2*pi*(0.05+0.2/256*t).*t);
%Liimenju shkersojuma diskretizācija tiek veikta ar R.Shaavelja izveidotu
%funkciju lvcrossing. tn ir nolases laika moments, sn ir nolases
%amplituudas veertiiba. Diskretizācijas precizitāte 10.e-4; ACP bitu
%skaits ir 4 un ACP amplituodu diapazons R=Umax-Umin ir 4
[sn,tn]=lvcrossing(256,1e-4,'cos(2*pi*(0.05+0.2/256*t).*t)',4,4);
%signāla, kā laika funkcijas grafiska reprezentācija
plot(t,x,tn,sn, '.')
xlabel('laiks')
ylabel('s(t)')
%***
%Laika un frekvences analīzes rezģis
%***
N=256;
ta=0:N-1; %analīzes laika momenti
fa=[-.5+1/N:1/N:0.5]; %analīzes frekvences
N=length(fa);
%***
%DFT analīze bez laika ass (spektrs)
%***
figure
dftp=(1/length(sn)).*abs(sn*exp(i*2*pi*tn'*fa));
plot(fa, dftp, 'k');
set(gca, 'xlim', [-0.5 0.5], 'fontsize', 14)
set(gca, 'ylim', [0 max(dftp)], 'fontsize', 14)
xlabel('frekvence', 'fontsize', 16)
ylabel('spektrs', 'fontsize', 16)
NN=length(fa);
set(gca, 'xtick', [-0.5 -0.3 -0.1 0 0.1 0.3 0.5]);
set(gca, 'ytick', []);
set(gca, 'xgrid', 'off', 'ygrid', 'off');
set(gcf, 'color', [1 1 1]);
%***
%Laika analīzes loga funkcijas aprekins.
%lietojam n_window(N,'rect'|'hamming'(default)|'hanning',t) funkciju
%***
T=32; %loga garums
%Izveidojam signals.*logs matricu
tfr=zeros(length(tn),length(ta));
%ciklaa ir aprakstiita loga bidishana pa visu signāla laika garumu
for icol=1:length(ta),
    tau=ta(icol);
    ha=n_window(T,'hamming',tn+T/2-tau); % tn+L/2-tau ir laika momenti logam
    tfr(:,icol)=sn'.*conj(ha); % reizinaajums signāls un kompleksa saistita
    Hamminga loga f-ja
end;
%***
%Veidojam krasu gammu 3D bildem
%***
L=16; %limenu tonu skaits

```

```

ac=[1-1/L 1-1/L 1-1/L];
for ii=3*L/4:-1:1/L,
    ac=[ac; [ii./L ii./L ii./L]];
end
%***
%Lai iegutu precizaako rezultatu, mees pielietosim DFT ne-ortogonālas
transformāciju
%***
non_exp=exp(-i*2*pi*tn'*fa);
non_tmp=non_exp*inv(non_exp'*non_exp+.2*eye(size(non_exp'*non_exp))); %notiek
inversija ar satbilizāciju
non_tfr=tfr'*non_tmp/length(tn);
figure
mesh(abs(non_tfr'));
colormap(ac);
set(gca, 'xlim', [0 length(ta)], 'fontsize', 14)
set(gca, 'ylim', [0 length(fa)], 'fontsize', 14)
set(gca, 'zlim', [0 max(max(abs(non_tfr)))], 'fontsize', 14)
xlabel('Laiks', 'fontsize', 16)
ylabel('Frekvence', 'fontsize', 16)
view(0, 90);
NN=length(fa);
set(gca, 'ytick', [0 .2*NN 0.4*NN 0.5*NN 0.6*NN .8*NN NN]);
set(gca, 'yticklabel', [-0.5 -0.3 -0.1 0 0.1 0.3 0.5]);
set(gca, 'xgrid', 'off', 'ygrid', 'off', 'zgrid', 'off');
set(gca, 'ztick', [], 'zcolor', [1 1 1]);
set(gcf, 'color', [1 1 1]);

```

7. PIELIKUMS. D.

```

%Programma DFT-veidigu parveidojumu modelesanai
%no analiitiski noteiktam level-crossing sampling sinusoidalam
stacionaram signalam
%
%VPP "IT zinātniska baze" 3.projekts
%Autors M.Greitans
%2006.gada 26.junijs
%
%Signals aprakstams formaa  $x(t)=amp*\cos(2*pi*fr*t+ph)$ 

clear all
close all

%%Tiek analizets signala intervaals [0...TH[ (sekundes)
TH=128;

amp=1;      %amplituda =const
fr=12.22/128; %frekvence normalizeeta attiecibaa pret Fsamp=1
%fr=0.09;
rand('seed', 931316785);

ph=rand*2*pi;      %saakuma faaze
%ph=2*pi*11/1000;

%%Qlev[] satur kvanteshanas limenus
%Qlev=[-1 -1./3 1./3 1];
%alternatiiva aprekināam limenus vienmerigi sadalot diapozonu
qbit=3;      %bitu skaits liimenu kodeshanai
Qlev=linspace(-amp+(amp./2.^qbit), amp-(amp./2.^qbit), 2.^qbit-1);

K=ceil(TH*fr)+1; %max signala periodu skaits laika intervaalaa TH

tlev=[];      %tlev ieguto laika momentu vektors
xlev=[];      %xlev ieguto limenu vertibu vektors

for k=0:K,
    tlev=[tlev (k*2*pi-acos(Qlev./amp)-ph)./(2*pi*fr)];
    xlev=[xlev Qlev];
    tlev=[tlev (k*2*pi+acos(Qlev./amp)-ph)./(2*pi*fr)];
    xlev=[xlev Qlev];
end

%atstajam tikai taas nolases kas ietilpst [0..TH]
tlc=tlev(tlev>=0&tlev<=TH);
xlc=xlev(tlev>=0&tlev<=TH);

%sakartojam pieaugosha laika seciibaa
[tlc, I]=sort(tlc);
xlc=xlc(I);

%ziimejam iegutaas veertibas
figure(1)
plot(tlc, xlc, '*');

%salidzinajums ar "istajaam" vertibaam laikos tlc
xc=amp.*cos(2*pi*fr*tlc+ph);
hold on
plot(tlc,xc, 'g');

%zimejam "originalo" signalu

```



```

tt=linspace(0,TH,1000);
xx=amp*cos(2*pi*fr*tt+ph);
hold on
plot(tt,xx,'r')

dtlc=0.5*( [tlc(2:length(tlc)) TH]-[0 tlc(1:length(tlc)-1)]);

%ieguustam General DFT no signaala
fra=-0.5+1/TH:1/TH:0.5; %analiizes frekvencu vektors
n_exp=exp(-i*2*pi*tlc'*fra);
ndft=1./diag(n_exp'*n_exp)'.*(xlc*n_exp);
figure(2)
stem(fra(64:128),10*log10(abs(ndft(64:128))), 'color', 'k','marker',
'no', 'ShowBaseLine', 'off', 'BaseValue', -
30);%floor(min(10*log10(abs(ndft)))));
xlabel('Frequency [Hz]')
ylabel('|GDFT| [dB]')
set(gca, 'ylim', [-30 0]);

hold on
%Modifikaacija ar svertaam nolaseem
ndft_m=1./diag(n_exp'*n_exp)'.*(xlc.*dtlc)*n_exp);
stem(fra,10*log10(abs(ndft_m)), 'color', 'r','marker', '+',
'ShowBaseLine', 'off', 'BaseValue', min(10*log10(abs(ndft)))));

%Modifikacija ar 0-kartas signala interpolaciju
nidft=i./(2.*pi.*fra).*(xlc*(n_exp.*(1.-exp(i.*2.*pi.*dtlc'*fra))));
figure(3)
stem(fra(64:128),10*log10(abs(nidft(64:128))), 'marker', 'no',
'ShowBaseLine', 'off', 'BaseValue', min(10*log10(abs(nidft)))));
hold on
figure

%Modifikacija ar 1-kartas signala interpolaciju
tlcm=tlc-tlc(1);
dtlcm=[tlcm(2:length(tlcm)) TH]-tlcm;
alc=( [xlc(2:length(xlc)) xlc(1)]-xlc)./dtlcm;
n_expm=exp(-i*2*pi*tlcm'*fra);
dn_expm=exp(i*2*pi*dtlcm'*fra);
nidft_m=i./(2.*pi.*fra).*(xlc*(n_expm.*(1.-
dn_expm))+1./(2.*pi.*fra).^2.*(alc*(n_expm.*(1.-
dn_expm))+i./(2.*pi.*fra).*(xlc.*dtlcm)*(n_expm.*dn_expm));

stem(fra(64:128),10*log10(abs(1/length(tlc)*nidft_m(64:128))),
'color', 'k','marker', 'no', 'ShowBaseLine', 'off', 'BaseValue', -
30);
xlabel('Frequency [Hz]')
ylabel('|X| [dB]')
set(gca, 'ylim', [-30 0]);

figure
%Modifikacija ar SINC interpolaciju signalam
t_si=.5:1:TH;
n_sinc=sinc(ones(size(t_si))*tlc-t_si*ones(size(tlc)));
nsincft=1./TH.*(xlc/n_sinc)*exp(-i*2*pi*t_si'*fra);
figure(4)
stem(fra(64:128),10*log10(abs(nsincft(64:128))), 'color',
'k','marker', 'no', 'ShowBaseLine', 'off', 'BaseValue', -
30);%min(10*log10(abs(nsincft)))));
xlabel('Frequency [Hz]')
ylabel('|X| [dB]')

```

```

set(gca, 'ylim', [-30 0]);

hold on

%Modifikacija ar neortogonalu parveidojumu
fran=-0.5+1/TH:1/TH:0.5;
non_exp=exp(-i*2*pi*tlc'*fran);
%nondft=xlc*pinv(non_exp');
nondft=xlc/non_exp';

%Modifikacija ar 0-kartas kludas interpolaciju
for kk=1:length(fran),
    non_exp_dt(:,kk)=dtlc'.*non_exp(:,kk);
end
nondft_m=(non_exp'*(xlc.*dtlc)')'*inv(non_exp'*non_exp_dt);

%Modifikacija ar 1-kartas kludas interpolaciju
for kk=1:length(fran),
    non_exp_dt2(:,kk)=(dtlc.^2)'.*non_exp(:,kk);
end
nondft_m2=(non_exp'*(xlc.*(dtlc.^2)')'*inv(non_exp'*non_exp_dt2);
dtlcm1=tlc(2:length(tlc))-tlc(1:length(tlc)-1);
nondft_m1=(non_exp(1:length(xlc)-1,:)'*(xlc(1:length(xlc)-1).*dtlcm1)+non_exp(2:length(xlc),:)'*(xlc(2:length(xlc)).*dtlcm1)')'*inv(non_exp(1:length(xlc)-1,:)'*non_exp_dt(1:length(xlc)-1,:)+non_exp(2:length(xlc),:)'*non_exp_dt(2:length(xlc),:)));
%plot(fran,10*log10(abs(nondft)));
%stem(fra,10*log10(abs(nondft)), 'color', 'b', 'marker', 's', 'ShowBaseLine', 'off', 'BaseValue', min(10*log10(abs(nondft))));
figure
stem(fra(64:128),10*log10(abs(nondft_m(64:128))), 'color', 'k', 'marker', 'no', 'ShowBaseLine', 'off', 'BaseValue', -30);%min(10*log10(abs(nondft))));
%stem(fra,10*log10(abs(nondft_m2)), 'color', 'c', 'marker', 'x', 'ShowBaseLine', 'off', 'BaseValue', min(10*log10(abs(nondft))));
%stem(fra,10*log10(abs(nondft_m1)), 'color', 'r', 'marker', '*', 'ShowBaseLine', 'off', 'BaseValue', min(10*log10(abs(nondft))));
xlabel('Frequency [Hz]')
ylabel('|NO-DFT| [dB]')

%Salīdzinājums ar signalatkariigu DFT-veidīgu analīzi
frasd=-0.5+1/TH/4:1/TH/4:0.5;
adft=nft_m(xlc, tlc, frasd);
figure(5)
stem(frasd(64*4:128*4),10*log10(abs(adft(64*4:128*4))), 'color', 'k', 'marker', 'no', 'ShowBaseLine', 'off', 'BaseValue', -100);%min(10*log10(abs(nondft))));
xlabel('Frequency [Hz]')
ylabel('|SD-DFT| [dB]')

%Signāla rekonstrukcijas kļūdu apmekināšana
tt=linspace(4,TH-4,5000);
xx=amp*cos(2*pi*fr*tt+ph);
figure(6)
plot(tt, xx, 'r')
hold on
x_ndft=real(ndft*exp(i*2*pi*(tt'*fra)'));
plot(tt, x_ndft)
x_ndft_m=real(ndft_m*exp(i*2*pi*(tt'*fra)'));
plot(tt, x_ndft_m, 'k')
x_nondft=real(nondft*exp(i*2*pi*(tt'*fran)'));

```

```

plot(tt, x_nondft, 'm')
x_nondftm=real(nondft_m*exp(i*2*pi*(tt'*fran')));
plot(tt, x_nondftm, 'g')
x_nondftm2=real(nondft_m2*exp(i*2*pi*(tt'*fran')));
plot(tt, x_nondftm2, 'c')
x_nondftml=real(nondft_ml*exp(i*2*pi*(tt'*fran')));
plot(tt, x_nondftml, 'y')
x_nsincft=real(nsincft*exp(i*2*pi*(tt'*fra')));
plot(tt, x_nsincft, 'r')
x_sdft=real(adft'*exp(-i*2*pi*(tt'*frasd')));
plot(tt, x_sdft)

e_ndft=sum((x_ndft-xx).^2)
e1_ndftm=sum((x_ndft_m-xx).^2)
e_nondft=sum((x_nondft-xx).^2)
e3_nondftm=sum((x_nondftm-xx).^2)
e4_nondftm2=sum((x_nondftm2-xx).^2)
e5_nondftml=sum((x_nondftml-xx).^2)
e_sinc=sum((x_nsincft-xx).^2)
e_sd=sum((x_sdft-xx).^2)

%atjaunosanas kludas zimesana
tt=linspace(0,TH,1000);
xx=amp*cos(2*pi*fr*tt+ph);
x_ndft=real(ndft*exp(i*2*pi*(tt'*fra')));
x_nondftm2=real(nondft_m2*exp(i*2*pi*(tt'*fra')));
figure(7)
plot(tt, (xx-x_ndft), 'k')
hold on
stem(tlc, (xlc-real(ndft*exp(i*2*pi*(tlc'*fra')))), 'color', 'k',
'marker', '.');
%set(gca,'ylim',[0 1]);
set(gca, 'xlim', [0 5]);
xlabel('Time [s]')
ylabel('Reconstruction error \epsilon(t)')
plot(tt, xx-x_nondftm2, 'm')
stem(tlc, xlc-real(nondft_m2*exp(i*2*pi*(tlc'*fra'))), 'marker', '*',
'color', 'm');

```

7. PIELIKUMS. E

```

%Apreekina faazi GSM signaalam (signals moduleets ar GMSK) peec
%uztvershanas un diskretizaacijas ar level crossing metodi. Ieejas
lielumi
%ir laika un amplitudas vektori.
%
*****
%
clear all;
close all;
run level_crossing;
clear all;
load prz1;
load prz2;
t=[]; tt3=[]; tt33=[]; lvv3=[]; phase=[]; t_phase=[];
C=[]; A=[]; B=[]; D=[]; guga=[]; phr=[]; PH=[];
X=(1/fr)*9; %Loga garums
SOLIS=(1/fr)*1; %Loga paarvietoshanaas solis
%solim jaabuut vienaadam ar vesalu skaitu signaala periodu
SOL=0; %Uzkraajamaa sola veertiiba
solis=X/4; %Ziimes mainas sola veertiiba
%*****
%
% Formee laika logu no vektora t3
%*****
for zb=0:SOLIS:t3(end)-X;%X-t3(end)/d-SOLIS;
zz=[];
for zz=1:length(t3);
ver=X+SOL; %Loga saakuma veertiba
if (t3(zz)>=ver-X)&&(t3(zz)<=ver);
t=[t,t3(zz)];
guga=[guga,zz];%Formee atbilstosaas lv3 adreses
veertiibas
end;
if t3(zz)==t3(end);
continue;
end;
if t3(zz+1)>ver;
break;
end;
end;
t_phase=[t_phase,t(round(length(t)/2))];%iestada fazes laika
veertiibu
%*****
%
% Formee t3 laika loga veertiibu ziimi
%*****
%*****
%Katram laika t3 logam X formee katras nolases ziimi,atkariibaa no

```

```

%taisnstuurveida funkcijas ziimes.Vienaa logaa ietilps viens
taisnstura
%f-cas periods.Otra taisnstura fca ir nobiidiita faazee par
pi/2.Kaa
%taisnstuura fcu izmanto fcu +1; -1.Attieciigi, ja nolase iekriit
%+ apgabala, amplituudas veertiibai tiek pieskirta "+" veertiiba,
%bet,ja iekriit "-" apgabala - miinuss veertiiba.
for gar=1:length(t);
bin=[];
if (t(gar)<=(ver-(solis*3)));
bin=1;
elseif ((t(gar)>(ver-(solis*3)))&&(t(gar)<=(ver-
(solis*2)))));
bin=1;
elseif ((t(gar)>(ver-(solis*2)))&&(t(gar)<=(ver-solis)));
bin=-1;
elseif ((t(gar)>(ver-solis))&&(t(gar)<=ver));
bin=-1;
end;
tt3=[tt3,bin];
bin=[];
if (t(gar)<=(ver-(solis*3)));
bin=1;
elseif ((t(gar)>(ver-(solis*3)))&&(t(gar)<=(ver-
(solis*2)))));
bin=-1;
elseif ((t(gar)>(ver-(solis*2)))&&(t(gar)<=(ver-solis)));
bin=-1;
elseif ((t(gar)>(ver-solis))&&(t(gar)<=ver));
bin=1;
end;
tt33=[tt33,bin];
bin=[];
end;
lvv3=lvv3(guga(1):guga(end)); %Atbilstoso amplitudas adreshu
veritbas
%*****
%
% Aprekina fazes vertibu katram logam
%*****
A=tt3*lvv3'; %Reizina amplituudu ar + vai - ziimi
B=tt33*lvv3'; %Reizina amplituudu ar + vai - ziimi
C=[C,A];
D=[D,B];
SOL=SOL+SOLIS; %Naakosaa loga saakuma veertiiba
t=[];
tt3=[];
tt33=[];
lvv3=[];
zz=[];

```

```

guga=[];
p=(B/A);
A=[];B=[];
pp=atan(p);
phr=[phr,pp];
a=ones(1,length(phr));
phase=phr(1).*a; %Signals ar nemainigu fazi
end;
%*****
%
%           Korigee faazes izmainu
%*****
gmph=phr-phase; %Faazu starpiiba
epsi=pi/3; %Pielaujamaa kluudas robeza
for ee=1:length(gmph);
    if ee==1;
        if (gmph(ee)<0-epsi);
            gm=pi-gmph(ee);
            PH=[PH,gm];
            k=gm;
        elseif (gmph(ee)>0+epsi);
            gm=-pi+gmph(ee);
            PH=[PH,gm];
            k=gm;
        else
            gm=gmph(ee);
            PH=[PH,gm];
            k=gm;
        end;
        continue;
    end;
    if ee>1;
        if k>0;
            if (gmph(ee)<k-epsi);
                gm=pi+gmph(ee);
                PH=[PH,gm];
                k=gm;
            elseif (gmph(ee)>k+epsi);
                gm=pi-gmph(ee);
                PH=[PH,gm];
                k=gm;
            else
                gm=gmph(ee);
                PH=[PH,gm];
                k=gm;
            end;
            continue;
        end;
        if k<0;
            if (gmph(ee)>k+epsi);
                gm=gmph(ee)-pi;

```

```

                PH=[PH,gm];
                k=gm;
            elseif (gmph(ee)<k-epsi);
                gm=-(pi+gmph(ee));
                PH=[PH,gm];
                k=gm;
            else
                gm=gmph(ee);
                PH=[PH,gm];
                k=gm;
            end;
            continue;
        end;
        if k==0;
            if (gmph(ee)>k+epsi);
                gm=gmph(ee)-pi;
                PH=[PH,gm];
                k=gm;
            elseif (gmph(ee)<k-epsi);
                gm=-(pi+gmph(ee));
                PH=[PH,gm];
                k=gm;
            else
                gm=gmph(ee);
                PH=[PH,gm];
                k=gm;
            end;
            continue;
        end;
    end;
end;
%*****
%
%           Atkodee bitus
%*****
i=length(data)*symb_per; %datu suutiisanas ilgums
epsi=SOLIS*4 % Pielaujamaa kluda
m=[];adr=[];bit=[];t_adr=[];
sy=symb_per;
for ii=1:symb_per:i-symb_per;
    for gg=1:length(t_phase);
        if t_phase(gg)>ii+epsi;
            break;
        end;
        if (t_phase(gg)>=ii-epsi)&&(t_phase(gg)<=ii+epsi);
            m=[m,t_phase(gg)];
            adr=[adr,gg];
            continue;
        else;
            continue;
        end;
    end;
end;

```

```

        end;
    end;
    for mm=1:length(m);
        if (abs(m(mm)-sy))>(abs(m(mm+1)-sy));
            continue;
        else
            bit=[bit,m(mm)]; %Bitu laika vertibas
            t_adr=[t_adr,adr(mm)]; %Atilstosaas adreses
            m=[];
            adr=[];
        end;
        break;
    end;
    sy=sy+symb_per;
    if ii==length(data)-2;
        break;
    end;
end;
pii=[];%Faazes veertiibas simbola mainas punktus
for mm=1:length(t_adr);
    for nn=1:length(PH);
        if t_adr(mm)==nn;
            pii=[pii,PH(nn)];
            break;
        end;
    end;
end;
bits=[];%Atkodeetie biti
for mm=1:length(pii);
    if mm==1;
        bi=0;
        bits=[bits,bi];
        ko=pii(mm);
        continue;
    end;
    if pii(mm)>ko;
        if bi==0;
            a=0;
            bits=[bits,a];
            bi=a;
            ko=pii(mm);
            continue;
        else
            a=1;
            bits=[bits,a];
            bi=a;
            ko=pii(mm);
            continue;
        end;
    end;
end;
if pii(mm)<=ko;
    if bi==0;

```

```

        a=1;
        bits=[bits,a];
        bi=a;
        ko=pii(mm);
        continue;
    else
        a=0;
        bits=[bits,a];
        bi=a;
        ko=pii(mm);
        continue;
    end;
end;
end;
figure;plot(t_phase,PH); %Zimee uztvertaa signaala faazes izmainu
xlabel('laiks, s');
ylabel('Faze, rad');grid;
hold on;
plot(laiks,mod_ph,'r'); %Ziimee raidiitaa signaala faazes izmainu
%*****
%
%                Apreekina bitu paarraides kluudu
%*****
ok=0;error=0;
for bb=1:length(bits);
    n=bb;
    if bits(bb)==data(n);
        ok=ok+1;
    else
        error=error+1;
    end;
end;
if ok==length(bits);
    BER=0;
else
    BER=error/length(bits);
end;
end;

```