Tallinna Tehnikaülikool

Automaatika teaduskond

Aarne Roosi

SIGNAALITÖÖTLUSSÜSTEEMIDE MODELLEERIMINE ARVUTIL

Magistritöö

Juhendaja: dots. Vladimir Heinrichsen

T — 5477
TALLINNA TEHNIKAÜLIKOOLI
RAAMATUKOGU

Tallinn 1992

SIGNAALITÖÖTLUSSÜSTEEMIDE MODELLEERIMINE ARVUTIL

Aarne Roosi
Tallinna Tehnikaülikool, 1992

ANNOTATSIOON

Võtmesõnad: **modelleerimine arvutil, simuleerimine, signaalitöötlus.**

Töös käsitletakse diskreetajas töötavate signaalitöötlussüsteemide modelleerimisega seotud küsimusi. Põhjalikumalt vaadeldakse TTÜ raadiotehnika kateedris loodud funktsionaalskeemide modelleerimiskeskkonda SIMULANT ja kirjeldatakse selle kasutamist signaalitöötluse laboratoorsetes töödes.

Töö koosneb seletuskirjast 65 leheküljel (25 tekstijoonist) ja sisaldab lisadena modelleerimiskeskkonna SIMULANT kasutajajuhendit ning mudelite koostamisel kasutatavate tüüpplokkide kirjeldusi.

COMPUTER MODELLING OF SIGNAL PROCESSING SYSTEMS

Aarne Roosi

Tallinn Technical University, 1992

ABSTRACT

Key words: **computer modelling, simulation, signal processing**

The thesis is devoted to the modelling of the discrete-time signal processing systems. The modelling system "SIMULANT" developed in the Department of Radio Engineering and its implementation to students laboratory works in signal processing is described.

This paper is written in Estonian. It contains of an explanatory letter on 65 pages (25 figures), appendices with user's guide to simulating environment SIMULANT and descriptions of the standard model building blocks.

SISUKORD

0. SISSEJUHATUS 5
1. ARVUTIMODELLEERIMISE ÜLEVAADE 7
1.1. Modelleerimisel kasutatavad põhimõisted 7
1.2. Ülevaade olemasolevast modelleerimistarkvarast 8
2. ÜLESANDE PÜSTITUS 12
3. MODELLEERIMISKESKKONNA KIRJELDUS 17
3.1. Mudeli ja plokitüüpide kirjeldamine 19
3.2. Mudeli koostamine 20
3.3. Simulaatori töö kirjeldamine 22
3.4. Integreeritud modelleerimiskeskkond 24
4. SIGNAALITÖÖTLUSSÜSTEEMIDE MUDELITE KOOSTAMINE 26
Näide 4.1. Harmoonilise signaali generaatori uurimine 26
Näide 4.2. Rekursiivfiltri mudel 32
Näide 4.3. Väljundsignaalide parve leidmine 35
5. LABORATOORSETE TÖÖDE NÄIDISKIRJELDUSED 38
5.1. Suhtelise faasmanipulatsiooniga moodem 48
5.2. IKM modulatsiooniviisid 53
5.3. Spektraalteisendused mittelineaarsetes ahelates 57
6. KOKKUVÕTE 63
KIRJANDUSE LOETELU 64

LISAD

Lisa 1. Akt signaalitöötluse süsteemide simulaatori programmi "SIMULANT" autorite kohta
Lisa 2. Funktsionaalplokkide kataloog (eraldi köites)
Lisa 3. RAADIOTEHNILISTE FUNKTSIONAALSKEEMIDE SIMULAATOR. Kirjeldus ja kasutusjuhend (eraldi köites)

SISSEJUHATUS

Raadiosüsteemide, automaatika, infoedastuse, mõõtmiste, signaalitöötluse jpm. valdkondades analüüsitakse mitmesuguseid algoritme. Sealjuures uuritakse, kuidas teisenduvad signaalid mingite teadaolevate funktsionaalskeemide läbimisel. Kõrvuti natuureksperimendiga kasutatakse sealjuures sageli uuritavate skeemide funktsionaalset modelleerimist arvutil.

Esimesed arvuti kasutamise kogemused raadiotehnika kateedri õppetöös pärinevad ajast, millal kateedrisse tekkisid esimesed arvutid. Kõrvuti positiivsete kogemustega (s.o. arvutit kui sellist saab kasutada modelleerimisel [2,7,10]) tekkis rida tähelepanekuid ka selle kohta, kuidas seda teha ei tohiks. Edasise töö käigus selgus, et programmide raskuskese nihkus üha enam otseselt modelleerivalt osalt kasutajaliidesele, s.o. andmete käepärasele sisestusele, väljastusele, tulemuste interpreteerimisele. Osutus ebaratsionaalseks ja sageli üle jõu käivaks teha programme iga kitsama teema või ainevaldkonna modelleerimiseks eraldi. Kasutajaliidese võimalused (sealhulgas see, millisel määral ja viisil saab mõjutada ja jälgida modelleerimise käiku) piirasid sageli modelleerimise kasutatavust õppetöös. Väheotstarbekaks osutus suund, kus programmi algoritm püüdis järgida õppetöö läbiviimise käiku, sama tulutud olid katsed kohandada laboratoorseid töid vastavalt olemasolevatele programmidele. Tekkis vajadus modelleerimiskeskkonna järele, mis jätaks töö teostajale piisavalt suure vabadusastme ja võimaldaks samas modelleerida küllalt laia ülesannete klassi. Ühe võimaliku lahendusteena pakub autor üleminekut mudeli protseduurselt kirjeldamiselt selle funktsionaalsele kirjeldamisele.

Käesoleva töö esimeses osas püütakse anda lühiülevaade signaalitöötlussüsteemide funktsionaalsel modelleerimisel kasutatavatest vahenditest. Töö teises osas vaadeldakse funktsionaalse modelleerimise keskkonnale püstitatavaid nõudeid. Kolmandas osas kirjeldatakse raadiotehnika kateedris töö autori osavõtul loodud modelleerimiskeskkonda SIMULANT.

Signaalitöötluse ülesannete funktsionaalse modelleerimise problemaatika on vaatluse all töö neljandas osas. Viiendas osas antakse ülevaade signaalitöötlussüsteemide seni väljatöötatud mudelitest ning kirjeldatakse nende kasutamist õppe-laboratoorsetes töödes. Töö sisaldab lisadena modelleerimiskeskkonna SIMULANT kasutajajuhendit ja enamkasutatavate funktsionaalplokkide kataloogi.

1. ARVUTIMODELLEERIMISE ÜLEVAADE

1.1. Modelleerimisel kasutatavad põhimõisted

Funktsionaalne modelleerimine on modelleerimisülesannete üheks erijuhuks. Modelleerimise üldprobleeme on kirjanduses viimastel aastakümnetel põhjalikult käsitletud [8,9,11,12,15].

Modelleerimine seisneb üldjuhul uuritava nähtuse, objekti või idee esitamises mingil teisel kujul selle uurimise eesmärgil [8,15].

Arvutimodelleerimine seisneb arvuti programmeerimises selleks, et saada mudelit, mis käitub uuritavas valdkonnas samade reeglite järgi kui reaalne või hüpoteetiline modelleeritav süsteem [11]. Selles kontekstis on arvutimodelleerimise metodoloogia vaadeldav tarkvara projekteerimise metodoloogia ühe erivormina [15].

Simuleerimine on arvutimudeliga katsetamine modelleeritava süsteemi käitumise mõistmise eesmärgil [11,15]. Edaspidi kasutame mõistet "simuleerimine" viimatimainitud kitsamas tähenduses. Mõiste "modelleerimine" sisaldab sel juhul nii mudeli tuletamise reaalsest või hüpoteetilisest süsteemist, mudeli koostamise kui ka selle katsetamise.

Katse keskkond konkretiseerib mudeli katsetamise tingimused. Seetõttu saab ajaliselt eristada mudeli koostamise ja selle katsetamise. Nii saab sama mudeliga teha erinevaid katseid, samuti saab sama katset sooritada erinevate mudelitega [15].

Arvutimudel koosneb reeglina seda kirjeldavast struktuurist, selle olekut kirjeldavate muutujate väärtustest ning algoritmist, mille täitmise tulemusena mudeli olek selle katsetamise ajal muutub [8].

Funktsionaalse modelleerimise täpset mõistet pole käesoleva töö autor kirjandusest leidnud. Üldjoontes mõistetakse selle all mudeli koostamist selle kirjeldamise kaudu. Sealjuures vaadeldakse mudelit koosnevana üksteisest sõltumatutest osadest. Mudeli käitumine tervikuna tuletatakse

simuleerimise käigus arvutiprogrammi poolt lähtudes mudeli koostisosade eelnevalt teadaolevatest parameetritest ja tööalgoritmidest ning kasutatavast katsekeskkonnast.

1.2. Ülevaade olemasolevast modelleerimistarkvarast

Modelleerimistarkvara aktiivsete otsingute algusaeg kateedris langes kokku IBM-tüüpi personaalarvutite laiema levikuga tehnikaülikoolis. Lisaks arvutitele levis laialdaselt ka tarkvara, sealhulgas spetsiaalselt modelleerimise jaoks koostatud programmid. Kuna oli põhjust eeldada selletüübiliste arvutite peatset kasutuselevõttu õppetöös, võis vaatluse alt välja jätta suurarvutite jaoks loodud programmipaketid.

Kättesaadava tarkvara analüüsil selgus, et erinevaid modelleerimisprogramme põhimõtteskeemi tasandil leidub piisavalt. Lihtsamate analoogskeemide modelleerimiseks sobib Micro-Cap, professionaalsel tasandil analüüsi võimaldab pakett PSPICE. Analoogskeemide analüüsiks sobib hästi ka automaatika kateedris väljatöötatud programmipakett SPADE. Loogikaskeemide modelleerimist võimaldab programm Micro-Logic. Saab kasutada ka projekteerimissüsteemidesse PCAD ja EE-DESIGNER integreeritud simulaatoreid.

Signaalitöötlusel kasutatavad skeemid on sageli küllalt keerukad, mistõttu mudeli kirjeldamine põhimõtteskeemina pole alati otstarbekas. Samuti on selliselt kirjeldatud mudel ressursimahukas. Nii kulub faashaardekontuuri töö modelleerimiseks kolmekümne perioodi (1000 signaalidiskreedi) ulatuses keskkonnas SPICE tööjaamal SUN üle veerand tunni. Uuritavad signaalitöötlusskeemid on sageli veel keerukamad. Liiga detailse mudeli korral kaob ka ülevaatlikkus. Raskusi tekib analoog- ja digitaalsõlmede kombineeritud modelleerimisel.

Kõige paindlikumaks ja universaalsemaks arvutimodelleerimise vahendiks on universaalsed protseduursed programmeerimiskeeled - FORTRAN, Pascal, C jt. Neid saab edukalt kasutada ka raadiotehniliste süsteemide funktsionaalsel modelleerimisel [2,9,10]. Paraku on mudeli

kirjeldamine universaalkeeltes keeruline ja aeganõudev, mudeli ja selle katsekeskkonna edaspidine modifitseerimine või korduvkasutamine on raskendatud.

Üldotstarbelised modelleerimiskeeled (GPSS, SLAM, SIMULA) [1,12] on mõnevõrra kohmakad, ressursimahukad ja keerulised kasutada. Need on orienteeritud eeskätt majandusülesannete ja tootmisülesannete lahendamisele. Ülalloetletud keeled on kasutatavad kommunikatsioonisüsteemide modelleerimisel (kanali läbilaskevõime, ooteaeg jne.).

Signaalitöötluse ülesannete modelleerimiseks sobib interpreteeriv programmeerimiskeel MatLab, millel on lisaks maatriksarvutuste kirjeldamist võimaldavale programmerimiskeelele rida signaalitöötluse ülesanneteks sobivaid funktsioone ja käepäraseid vahendeid tulemuste graafiliseks esitamiseks [13]. Ka programmeerimiskeskkonnas ASYST ja programmis ASYSTANT on samalaadsed vahendid. Sellele vaatamata jääb MatLab olemuselt protseduurseks keeleks ning ASYST on olemuselt ainult pinukalkulaator.

Lihtsamaid modelleerimisülesandeid saab lahendada ka kasutades tabelarvutuse vahendeid [5]. Tabelarvutuse programmipaketid sisaldavad küllalt mitmekülgseid vahendeid nii ülesande kirjeldamiseks kui ka tulemuste graafiliseks esitamiseks. Vaatamata publikatsioonide rohkusele selles valdkonnas pole tabelarvutuse vahendite laialdane kasutamine modelleerimisel otstarbekas.

Viimastel aastakümnetel on lisaks lõpp-produkti, s.o. mudeli efektiivsusele (töökiirus, mälu kasutamine) tähtsal kohal ka programmeerimise, s.o. mudeli koostamise efektiivsuse tõstmine. Nii tarkvara arengus üldse kui ka modelleerimistarkvara arengus on viimastel aastakümnetel toimumas üleminek protseduurselt programmeerimiselt funktsionaalsele programmeerimisele. Programmeerimiskeel omandab kirjeldava iseloomu, ülesande lahendusalgoritmi püütakse sünteesida automaatselt kirjelduse järgi. Sel juhul saab kogu kasutatavat tarkvara võrrelda jäämäega, mille veepealse osa moodustab kasutaja poolt koostatav kirjeldus [17]. Matemaatilise loogika ja kontseptuaalse programmeerimise põhiprintsiipidele tuginevate programmeerimiskeelte (PROLOG,

samuti PRIZ-keelte seeria) kasutamisel lüheneb oluliselt programmi koostamiseks ja silumiseks kuluv aeg [16]. Kontseptuaalse programmeerimise keeled (Expert-PRIZ, CPRIZ jpt.) võimaldavad uuritava mudeli kontseptuaalset kirjeldamist, kasutades kirjeldatavale ainevaldkonnale omaseid mõisteid. Paraku on signaalitöötluse ülesannetel oma eripära - arvutuste ja andmete suur maht, mis tingib vajaduse kasutada efektiivseid algoritme ja programme. See ei võimalda praegu veel rakendada heuristilise ja kontseptuaalse programmeerimise kõiki eeliseid ja eeldab modelleeriva keskkonna protseduurse külje säilitamist.

Kompromissina püütakse välja töötada modelleerimiskeskkondi konkreetsete ülesannete lahendamiseks. Lihtsamal juhul piirdutakse täiendavate võimaluste lisamisega olemasolevale protseduursele keelele [3]. Kaalumisel oli mõne universaalkeele (Small-C, Basic) kohandamine signaalitöötluse ülesannete lahendamiseks. Kuna see oleks püstitatud ülesande lahendanud ainult osaliselt, sai sellest ideest peagi loobutud. Küll osutusid nende translaatorite uurimisel omandatud kogemused kasulikuks modelleerimiskeskkonna hilisemal realiseerimisel.

Viimasel ajal kasutatakse sageli mudeli funktsionaalset (struktuurset) kirjeldamist võimaldavaid modelleerimiskeeli ja keskkondi lähtudes objekt-orienteeritud programmeerimise (OOP) põhimõtetest [4, 6, 14]. Sel juhul koostatakse mudel teatud tüüpklassidesse kuuluvatest objektidest, mis sisaldavad nii objekti simuleerimisel kasutatavaid andmeid kui ka simuleerimise algoritmi. Ka käesolevas töös kirjeldatav modelleerimiskeskkond kuulub viimatimainitute hulka.

Elektriinseneride väljaõppel kasutatava modelleerimistarkvara valikul tuleb lisaks selle üldistele omadustele arvestada veel järgmiste asjaoludega [4]:

1. Paljud üliõpilased saavad ja tahavad arvutit kasutada ka väljaspool õppetöö aega. Sealjuures ei saa neile kopeerida litsenseeritud tarkvara (modelleerimisprogrammid on paraku väga kallid ja pole üliõpilastele taskukohased), küll võivad nad piiranguteta kasutada ülikoolis väljatöötatud programme.

2. Professionaalsete tarkvarapakettide kasutamine nõuab

reeglina erialaseid eelteadmisi, seetõttu peavad vastavate ainete õpetamisel kasutatavad programmid olema oluliselt lihtsamad.

3. Otstarbekam on sobitada kasutatavat tarkvara vastavalt õppetööle kui korraldada õppetööd vastavalt olemasolevale tarkvarale.

2. ÜLESANDE PÜSTITUS

Modelleerimiskeskkonna väljatöötamisel olid aluseks järgmised nõuded ja eeldused:

1. Eeldatakse, et modelleerimine toimub funktsionaalsel tasandil. See võimaldab mudeleid kirjeldada parasjagu nii keerulisena kui seda mudeli uurimisel vajatakse ja hoiduda liigsetest detailidest.

2. Eeldatakse, et modelleeritakse diskreetajas töötavaid süsteeme, milliste töö on kirjeldatav vahevõrrandite abil. Vahevõrrandite abil saab küllaldase diskreetimissageduse korral kirjeldada ka pidevatoimelisi tagasisidestamata süsteeme.

3. Mudeli koostamine, kasutamine ja modifitseerimine peab olema ülevaatlik. Sellest tulenevalt võiks mudel olla struktuurne. Graafilise plokk-orienteeritud redaktori kasutamine hõlbustaks mudeli koostamist ja kasutamist veelgi ning võimaldaks modelleerimiskeskkonna kasutamist ka arvutit mittevaldavatel inimestel.

4. Kord koostatud mudelit peaks saama kasutada ka teiste mudelite koostamisel. Sellest nõudest tuleneb kaks järeldust:

   1) Mudel võiks koosneda plokkidest, milliste kirjeldamine ja silumine peaks olema mudeli struktuuri kirjeldamisest ajaliselt eraldatud. See võimaldaks kord koostatud ja katsetatud plokke kasutada erinevate mudelite juures ja vähendada selliselt uute mudelite koostamiseks kuluvat aega. Selle nõude täitmiseks peavad plokid olema üksteisest sõltumatud. Seetõttu tohivad plokkide väljundsignaalid sõltuda üksnes vastavate plokkide sisendsignaalidest ja parameetritest. Samuti peavad plokkide parameetrid olema lokaalsed, see võimaldab vältida erinevates plokkides samanimeliste muutujate kasutamise korral tekkivaid kõrvalefekte.

   2) Kord koostatud mudelit peaks edaspidi saama kasutada keerukama mudeli koostisosana (p.6).

5. Mudeli koostamine ja muutmine, samuti mudeli

parameetrite väärtustamine peab olema lihtne ja operatiivne, tulemuste esitus peab olema ülevaatlik. Seetõttu tuleks eelistada redaktorit, menüüsüsteemi, vigade diagnostikasüsteemi, HELP-süsteemi, väärtustajat, silujat, simulaatorit ja graafikavahendeid sisaldavat integreeritud modelleerimiskeskkonda. Lisaks sellele vajatakse vahendeid modelleerimistulemuste dokumenteerimiseks.

6. Modelleerimiskeskkond peab toetama hierarhilist mudeli kirjeldamist, s.t. kord koostatud ja silutud mudeleid või nende osi peaks saama kasutada keerukamate mudelite koostamisel makrodena. See võimaldab säilitada mudeli küllaldast ülevaatlikkust ka keerukamate mudelite korral, kuna erineva keerukusega mudelites saab kasutada erineva detailsusega struktuuriühikuid - plokke ja makroplokke. Seetõttu peab modelleerimiskeskkond sisaldama vahendeid vastavate makrokirjelduste loomiseks ning kasutamiseks.

7. Modelleeritava süsteemi uurimisel katsetatakse mudelit selle parameetrite erinevatel väärtustel. Modelleerimiskeskkond peaks võimaldama parameetrite varieerimist nii käsitsi, lähtudes eelneva simuleerimise tulemustest, kui ka automaatselt, etteantud algoritmi järgi. Erinevate simuleerimiste tulemused peaksid olema võrreldavad. See eeldab modelleerimistulemuste ja kasutatud parameetrite väärtuste vahepealset salvestamist ning esitamist parameetriliste sõltuvustena ja graafikuparvedena. Kordussimuleerimistel vajavad sealjuures mudeli teatud parameetrid (näiteks mudeli aeg) algväärtustamist. Seetõttu vajab mudel igale simuleerimisele eelnevat automaatset algseadistamist.

8. Modelleerimiskeskkond peab olema avatud. Uute mudelite kirjeldamisel vajatakse ka uusi plokitüüpe. Seetõttu peab saama lisada ja muuta plokitüüpe vastavalt vajadusele. Teiselt poolt on olemas suur hulk protseduursetes programmeerimiskeeltes (FORTRAN, C) koostatud efektiivseid signaalitöötlusprogramme, milliseid oleks samuti otstarbekas simuleerimise käigus kasutada. Seetõttu peaks modelleerimiskeskkond toetama vastavaid funktsioone sisaldavate moodulite dünaamilist laadimist/linkimist simuleerimise käigus.

9. Modelleerimiskeskkond peab võimaldama tagasisidestatud viitega struktuuride (näiteks rekursiivsed filtrid) kirjeldamist. Selleks peab tagasisidestatud mudeli osas simuleerimine toimuma signaalidiskreetide kaupa, s.o. samadel diskreetaja väärtustel kõigis plokkides üheaegselt. Samas tuleb avastada ja välistada algebraline tagasiside - s.o. tagasiside samal diskreetaja väärtusel. Viimasel juhul tuleks koostada tagasisidestatud mudeli osa kohta võrrandisüsteem ja see lahendada. Kuna sel juhul ei saa vaadelda mudeli koostisse kuuluvaid plokke enam sõltumatutena (plokkide väljundid tohivad sõltuda ainult ploki sisenditest), poleks see enam funktsionaalse modelleerimise ülesanne.

10. Modelleeritavate süsteemide erinevates osades võivad protsessid kulgeda erinevas tempos. Kiiremate protsesside modelleerimisel peab kasutama väiksemat diskreetaja sammu, aeglasemaid protsesse võib teatud juhtudel modelleerida suurema diskreetaja sammuga. Sel juhul töötavad mudeli erinevad osad erinevatel diskreetimissagedustel. Harilikult kasutatakse simuleerimisel globaalse diskreetaja mõistet (kõiki modelleeritavaid protsesse sünkroniseeritakse väljaspool mudelit asuva nn. globaalse mudeliaja suhtes [12]). Erinevate takteerimissageduste kasutamisel pole globaalaja kasutamine otstarbekas [6]. Teiselt poolt, reaalse maailma eelnevalt kettale salvestatud signaalide kasutamisel sõltuvad diskreetaja väärtused otseselt kasutatavast signaalist. Seetõttu on otstarbekas kasutada simuleerimisel globaalaja asemel nn. kohalikku aega, mis on seotud kasutatavate signaalidega. Signaalidega teisenduste sooritamisel (näiteks filtreerimise käigus nende diskreetimissageduse muutmisel) muutuvad vastavalt ka signaale kirjeldava diskreetaja parameetrid.

11. Modelleerimise käigus toimub simuleerimine reeglina signaalidiskreetide kaupa. Teatud töötlusalgoritmid (näiteks FFT, tulemuste graafiline esitus jm.) nõuavad sealjuures töötlusele eelnevalt reast järjestikustest signaalidiskreetidest koosneva signaalipaketi formeerimist. Näiteks FFT baasil realiseeritud filtri mudel sisaldab osa, kus signaalidiskreetidest formeeritakse signaalipakett, FFT abil leitav spekter korrutatakse läbi realiseeritava filtri

sageduskarakteristikuga ja teisendatakse Fourier pöördteisenduse kaudu uuesti ajasignaaliks. Sellise väljundsignaali võib vajadusel uuesti teisendada diskreetide jadaks. Modelleerimiskeskkond peab toetama pakett-signaalide formeerimist ja kasutamist kõrvuti järjestikuste signaalidiskreetide kasutamisega.

12. Modelleerimiskeskkonna potentsiaalseteks kasutajateks võivad olla nii esimeste kursuste üliõpilased kui ka uusi signaalitöötlusalgoritme välja töötavad teadurid. Seetõttu peab modelleerimiskeskkond rahuldama selle kasutajate erinevaid vajadusi.

Eelnevalt väljatöötatud ja testitud mudelite kasutamisel õppetöös vajatakse dialoogvahendeid katsekeskkonna kujundamiseks ja simuleerimise juhtimiseks, samuti mudeli kasutamist hõlbustavat HELP-süsteemi. Seetõttu vajab simuleerimiskeskkond vahendeid mudeli kasutajamenüüde kirjeldamiseks.

Erinevate signaalitöötlusalgoritmide võrdlemisel on vaja lisaks mudeli parameetritele teha ka mõningaid muudatusi mudeli struktuuris, lisada ja asendada mudelisse kuuluvaid plokke ning väärtustada nende parameetreid. Kuna vaadeldaval juhul ei saa enam piirduda eelnevalt koostatavate kasutajamenüüdega, peab HELP-süsteem toetama ka plokitüüpide valikut ja nende kasutamist, sealhulgas nende parameetrite väärtustamist.

Uute mudelite koostamisel vajatakse ka uusi plokitüüpe. Et kirjeldatavaid plokitüüpe saaks edaspidi edukalt kasutada, peab simuleerimiskeskkond sisaldama vahendid ka nende dokumenteerimiseks ning HELP-teabega varustamiseks, samuti loodavate plokitüüpide parameetrite algväärtustamiseks ja väärtustamisel kasutatavate menüüde kirjeldamiseks.

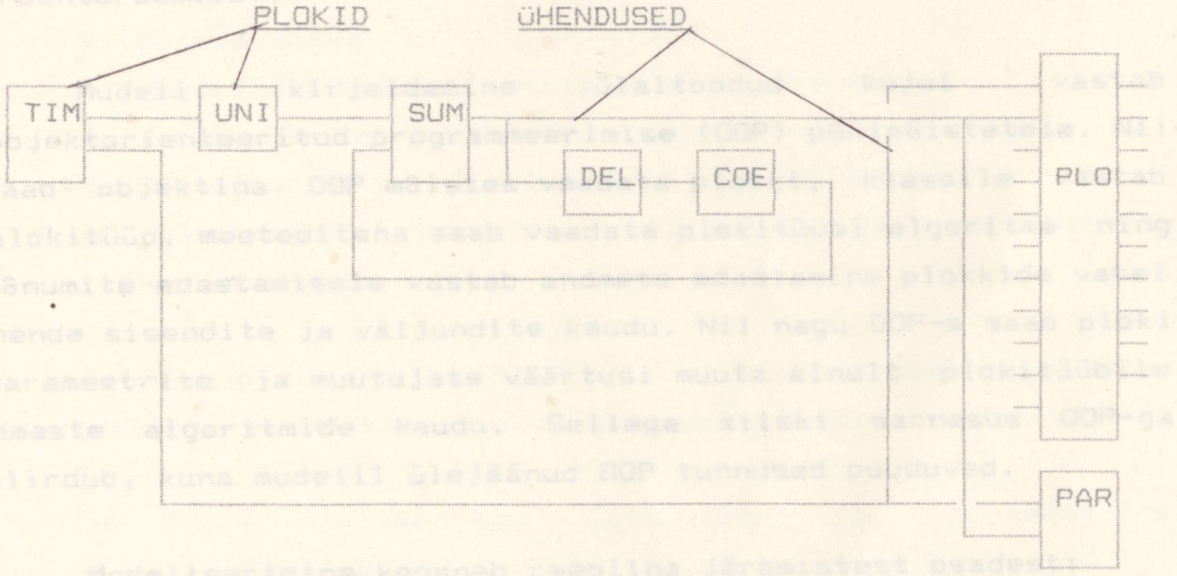
Teistes programmeerimiskeeltes koostatud välisfunktsioonide kasutamisel vajavad need prototüüpimist, s.o. funktsioonide parameetrite ja tagastatavate väärtuste eelnevat kirjeldamist. Prototüüpimise hõlbustamiseks ja erinevates mudelites ning plokkides erinevast kirjeldamisest tulenevate vigade vältimiseks peaks modelleerimiskeskkond toetama funktsioonikirjeldusi sisaldavate päisfailide kasutamist.

Lähtudes ülaltoodud nõuetest (p.1...5) tuleks eelistada sündmus-orienteeritud diskreetmodelleerimist protsess-orienteeritud modelleerimisele ning interpreteerivat tüüpi translaatorit kompilaatorile [8,12].

3. MODELLEERIMISKESKKONNA KIRJELDUS

Modelleerimiskeskkonna väljatöötamisel on töö eelnevates osades kirjeldatud nõudeid püütud võimaluste piires arvestada.

Kuivõrd mudel peab olema struktuurne, kirjeldatakse seda objektidest koosnevana. Vaadeldava modelleerimiskeskkonna objektideks on plokid (joon.3.1). Plokid on mudeli koostisosad, millel on andmestruktuurid neid kirjeldavate parameetrite ja nende sisend- ning väljundväärtuste säilitamiseks ning mis on suutelised sooritama teatud tegevusi. Tegevuste täitmise järjekord simuleerimise käigus sõltub plokkide vahelistest suhetest - s.o. ühendustest.



Joon.3.1. Plokkidest koosnev mudel

Kuna signaalitöötlusülesanded on sageli jagatavad regulaarse struktuuriga protseduurideks, siis plokkide poolt täidetavaid tegevusi kirjeldatakse algoritmiliselt, protseduurse keele abil. Ploki protseduur on reeglite kogum, mille alusel leitakse ploki väljundmuutujate väärtused lähtudes ploki parameetrite, senise oleku ja sisendmuutujate väärtustest. Sisendmuutujatena kasutatakse teiste plokkide väljundmuutujaid vastavalt plokkide vahelistele ühendustele. Kuna plokkide algoritmid globaalseid muutujaid ei kasuta, siis

toimub simuleerimise käigus kõigi arvutustulemuste ja signaalide ülekanne plokkide vahel sisendmuutujate ja väljundmuutujate kaudu. Simuleerimist saab vaadata kvaasiparalleelse diskreetajas toimuva tegevusena, kus plokkide olek mudeli ajas muutub hetkeliselt ja samaaegselt, plokkide uus olek leitakse aga plokkide algoritmide järjestikusel täitmisel.

Kõigi samatüübiliste plokkide andmestruktuur ja tööalgoritmid määratakse kasutatava ploki tüübiga. Plokitüüpide kirjeldamine eelneb mudeli koostamisele, kord kirjeldatud plokitüüpe saab kasutada erinevate mudelite juures. Iga ploki andmestruktuur on unikaalne. Kõigi samatüübiliste plokkide simuleerimisalgoritm esineb arvuti mälus ühekordselt. Seetõttu peavad plokkide simuleerimisalgoritmid olema korduvkäivitatavad (reenteraablid).

Mudeli kirjeldamine ülaltoodud kujul vastab objektorienteeritud programmeerimise (OOP) põhimõistetele. Nii saab objektina OOP mõistes vaadata plokki, klassile vastab plokitüüp, meetoditena saab vaadata plokitüübi algoritme ning sõnumite edastamisele vastab andmete edastamine plokkide vahel nende sisendite ja väljundite kaudu. Nii nagu OOP-s saab ploki parameetrite ja muutujate väärtusi muuta ainult plokitüübile omaste algoritmide kaudu. Sellega siiski sarnasus OOP-ga piirdub, kuna mudelil ülejäänud OOP tunnused puuduvad.

Modelleerimine koosneb reeglina järgmistest osadest:

1. Mudeli kirjeldamine.
2. Mudeli parameetrite väärtustamine
3. Mudeli initsialiseerimine ja simuleerimine (vajadusel korduv)
4. Simuleerimistulemuste töötlus (vajadusel).

3.1. Mudeli ja plokitüüpide kirjeldamine

Mudeli kirjeldamisel võib kasutada protseduurset keelt SIMULANT, mis sisaldab lauseid järgmiste tegevuste sooritamiseks:

1. Plokkide ja makroplokkide tekitamine.
2. Plokkide ühendamine.
3. Simuleerimisel kasutatavate muutujate kirjeldamine.
4. Muutujate ja parameetrite väärtustamine.
5. Simuleerimise käivitamine.
6. Sisestus/väljastuslaused.
7. Kasutajamenüüde formeerimine.
8. Tingimuslaused
9. Korduslaused.
10. Moodulite dünaamilise laadimise juhtimine
11. Funktsioonide prototüüpimine.
12. Päisfailide ja asendusnimede kasutamine.
13. Makromudeli sisendite, väljundite ja parameetrite kirjeldamine.

Plokitüübi kirjeldamisel kasutatakse lisaks ülalloetletule veel lauseid järgmtegevuste sooritamiseks:

14. Ploki sisendite, väljundite ja parameetrite kirjeldamine
15. Ploki algoritmide kirjeldamine
16. Ploki sisendväärtuste küsimine

Iga ploki kirjeldus võib sisaldada järgmisi algoritme:

CREATE - ploki moodustamisel selle andmestruktuuri algväärtustamiseks kasutatav algoritm

EVAL - ploki parameetrite väärtustamisel kasutatav algoritm (väärtustamisel kasutatava menüü formeerimine, sisestatavate parameetrite kontroll)

INIT - iga simuleerimise alustamisel ploki muutujate

algseadistamiseks kasutatav algoritm

CALC - simuleerimise igal taktil täidetav ploki simuleerimisalgoritm

NEXT - tagasisidestatud mudeli simuleerimisel vajalik abialgoritm

AFTER - iga simuleerimise lõpul täidetav algoritm

Mudeli ja plokitüüpide kirjeldamisel kasutatavat protseduurset keelt vaadeldakse põhjalikumalt simulaatori kasutajajuhendis (lisa 3).

3.2. Mudeli koostamine

Mudeli koostamine ja simuleerimine võib toimuda dialoogis kasutajaga või vastavalt protseduursele kirjeldusele.

1. Mudeli koostamisel dialoogreziimis kasutatakse plokkide moodustamisel, nende ühendamisel, parameetrite väärtustamisel ja simuleerimise juhtimisel modelleerimiskeskkonda integreeritud graafilist redaktorit. Kord koostatud mudelil võib simuleerimise käivitada korduvalt, muutes vajadusel enne korduskäivitust mudeli parameetrite väärtusi või mudeli struktuuri. Vajadusel saab mudeli hetkeseisu salvestada faili. Samuti saab mudeli suvalist osa kirjeldada makrona. Kord kirjeldatud makrosid saab kasutada keerukamate mudelite koostamisel samaväärselt plokkidega.

2. Mudeli koostamisel vastavalt kirjeldusele eelneb simuleerimisele mudeli kirjeldamine. Selleks kasutatakse modelleerimiskeskkonda integreeritud tekstiredaktorit. Kirjeldus sisaldab korraldusi mudeli plokkide moodustamiseks, nende ühendamiseks ning parameetrite väärtustamiseks, samuti simuleerimise (vajadusel korduva) juhtimiseks ning tulemuste töötluseks. Lisaks mudelile saab kirjeldada ka selle parameetrite väärtustamisel ning simuleerimise juhtimisel kasutatavaid dialoogmenüüsid ning sisestus-väljastuslauseid. Vajadusel võib mudeli ja simuleerimise käigu kirjeldus sisaldada ka tingimuslikke ja korduslauseid.

Plokkide moodustamisel tekitatakse uus unikaalne ploki andmestruktuur vastavalt plokitüübi kirjeldusele. Selle käigus kontrollitakse eelnevalt, kas plokitüübi kirjeldus on arvuti mälus, selle puudumisel laaditakse plokitüübi kirjeldus arvuti kettalt. Nii plokitüübi kirjeldus ja algoritmid kui ka loodava ploki andmestruktuur paigutatakse paisktabelitesse.

Plokkide ühendamisel tekitatakse kahesuunaline seos ühe ploki sisendi (nimetame siin seda plokki sisendplokiks) ja teise ploki väljundi (nimetame siin väljundplokiks) vahel. Selleks kantakse sisendploki sisendit kirjeldavasse andmestruktuuri aadressid, mis viitavad väljundploki andmestruktuurile ja selle ühendatavale väljundile. Need on vajalikud, et sisendploki sisendväärtusena saaks kasutada väljundploki väljundväärtust, selle puudumisel aga saaks käivitada väljundploki simuleerimisalgoritmi. Teiselt poolt, sisendploki sisendi aadress lisatakse väljundploki ühenduste listi. See võimaldab peale väljundploki simuleerimisalgoritmi töö lõpetamist kanda andmete olemasolu tunnuse kõigisse selle väljunditega ühendatud sisendite andmestruktuuridesse.

Parameetrite väärtustamisel muudetakse plokkide andmestruktuuridesse kantud parameetrite väärtusi. Väärtustamisel saab kasutada funktsioone ja teiste parameetrite väärtusi. Parameetrite väärtustamine toimub kolmel juhul:

1. Uue ploki tekitamisel toimub selle parameetrite vaikimisi algväärtustamine, kasutades selleks vastava plokitüübi algoritmi CREATE.
2. Enne simuleerimise käivitamist, simuleerimise katkestuspunktides või järjestikuste simuleerimiste vahel võib väärtustada plokkide parameetreid dialoogis kasutajaga, sealjuures saab kasutada ka teiste plokkide parameetrite väärtusi. Algoritmi EVAL abil võib formeerida väärtustamisel kasutatavaid menüüsid, kontrollida ning teisendada sisestatavaid väärtusi.
3. Simuleerimisel vastavalt kirjeldusele toimub ka väärtustamine vastavalt kirjeldusele. Sealjuures saab kasutada ka teiste plokkide parameetrite väärtusi.

Lisaks sellele võivad plokkide parameetrid muutuda veel

kolmel viisil:

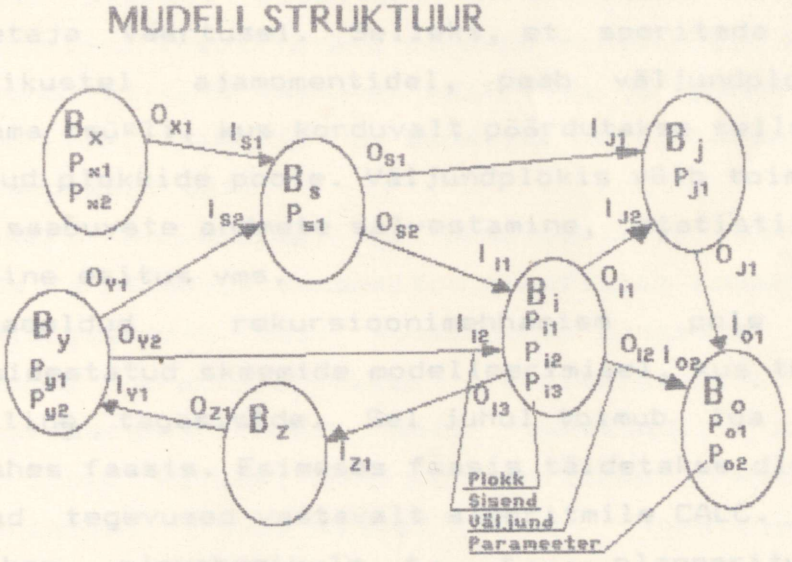
- igale simuleerimisele eelneva algseadistamise käigus vastavalt algoritmile INIT;
- igal simuleerimise taktil vastavalt algoritmidele CALC, NEXT;
- iga simuleerimise järeltöötlusel vastavalt algoritmile AFTER.

3.3. Simulaatori töö kirjeldamine

Simuleerimise käiku võib üldjoontes kirjeldada järgmiselt:

Simuleeritav mudel koosneb plokkidest. Igal plokil B_i võib olla k sisendit $I_i=\langle I_1 \ldots I_k \rangle$, l väljundit $O_i=\langle O_1 \ldots O_l \rangle$ ja m parameetrit $P_i=\langle P_1 \ldots P_m \rangle$ (joon.3.2). Plokki simuleeriva algoritmi F_i täitmise käigus leitakse diskreetsetel ajamomentidel t_j ploki väljundväärtused sõltuvana selle sisendväärtustest ja parameetritest:

$$O_{ij}=F_i(t_j,P_{ij},I_{ij}).$$



Joon.3.2. Mudeli struktuur

Sisendmuutujatena kasutatakse vastavate sisenditega ühendatud plokkide väljundmuutujaid. Neid saab kasutada üksnes peale nende väärtuste leidmist vastavalt nende plokkide simuleerimisalgoritmidele. Olgu ploki B_i sisend I_{i1} ühendatud ploki B_s väljundiga O_{s2}. Sel juhul

$$O_{ij}=F_i(t_j,P_{ij},(F_s(t_j,P_{sj},I_{sj}),I_{i2j})).$$

Kui ploki B_s sisend I_{s1} on omakorda ühendatud mõne järgmise ploki B_x väljundile O_{x1}, käivitub vajadusel omakorda selle ploki simuleerimisalgoritm. Nii toimub mudeli simuleerimisel diskreetajal t_j rekursiivne plokkide lahendusalgoritmide käivitumine kuni plokini, mille väljundväärtus vaadeldaval diskreetaja väärtusel on arvutatav üksnes ploki parameetrite kaudu ja ei sõltu ploki sisendite väärtustest. Selliseks plokiks on reeglina signaalallikas või viitelüli. Seejärel jätkuvad arvutused ülejäänud plokkides kuni kõik vajalikud väljundväärtused diskreetajal t_j on leitud.

Ülalkirjeldatud rekursiivne mehhanism tagab mudeli kvaasiparalleelse töö, s.o. mudeliajas samaaegselt toimuvate sündmuste õiges järjekorras arvutamise. Sealjuures algab simuleerimine plokist, mille väljundväärtusi soovitakse leida. Edaspidi nimetame sellist plokki väljundplokiks. Rekursioonimehhanism iseenesest tagab väärtuste leidmise ühel diskreetaja väärtusel. Selleks, et sooritada simuleerimist järjestikustel ajamomentidel, peab väljundploki algoritm sisaldama tsükli, kus korduvalt pöördutakse selle sisenditega ühendatud plokkide poole. Väljundplokis võib toimuda sisendite kaudu saabuvate andmete salvestamine, statistiline töötlus, graafiline esitus vms.

Vaadeldud rekursioonimehhanism pole küllaldane tagasisidestatud skeemide modelleerimisel, kus tuleb välistada algebraline tagasiside. Sel juhul toimub iga simuleerimise takt kahes faasis. Esimeses faasis täidetakse diskreetajal t_i toimuvad tegevused vastavalt algoritmile CALC. Teises faasis täidetakse ajavahemikule $t_i...t_{i+1}$ planeeritud tegevused vastavalt algoritmile NEXT. Simuleerimise käigus toimivad ülalvaadeldud plokid pseudogeneraatoritena. Ajamomendil t_i omistatakse algoritmi CALC abil ploki väljundmuutujale

väärtus, mis on leitud eelmisel taktil vastavalt algoritmile NEXT ja sisendite väärtustele ajavahemikus $t_{i-1}...t_i$. Seetõttu vajab selliste plokkide algolek simuleerimise käivitusmomendil t_0 täiendavat algseadistamist. Tüüpiliseks ülaltoodud kirjeldusele vastavaks plokiks on viitelüli DELAY (lisa 2).

3.4. Integreeritud modelleerimiskeskkond

Integreeritud modelleerimiskeskkond sisaldab vahendeid mudeli koostamiseks, silumiseks ja simuleerimiseks.

Simulaatori tööd juhib **menüüsüsteem**. Lisaks standardsetele menüüdele võimaldab keel SIMULANT kirjeldada ka plokkide ja mudelite parameetrite väärtustamisel ning simuleerimisel kasutatavaid täiendavaid menüüsid.

Kontekstuaalne **HELP-süsteem** võimaldab lisaks menüüvalikute ja veateadete selgitamisele saada teavet ka keele SIMULANT ja kasutatavate plokitüüpide ning makrode kohta. HELP-süsteem on täiendatav: iga plokitüübi HELP lisatakse otseselt plokitüübi kirjeldusse. Soovi korral saab simuleerimisel kasutatavate menüüde koostamisel kirjeldada ka igale menüüvalikule vastavat HELP-rida ja kaadrit.

Plokkide, makrode ja mudelite kirjeldused koostatakse **tekstiredaktori** abil.

Mudeli koostamisel ja simuleerimise juhtimisel saab kasutada ka plokk-orienteeritud **graafilist redaktorit**.

Tstiredaktori abil koostatavad plokkide ja mudelite kirjeldused transleeritakse leksikaanalüsaatori LEX ja grammatikaparseri YACC abil vahekoodi, mida hiljem, simuleerimise käigus, interpreteeritakse. Transleerimisel ja simuleerimisel avastatavad vead lokaliseeritakse ja võimaldatakse nende kohest parandamist.

Dünaamiline laadur/linkur võimaldab lisaks standardsetele keele SIMULANT funktsioonidele kasutada ka C ja ASSEMBLER-keeles kirjutatud funktsioone sisaldavaid laademooduleid.

Mudeli parameetrite väärtustamisel saab kasutada **väärtustajat** ja **kalkulaatorit.**

Mudeli silumisel saab kasutada plokk-orienteeritud **silujat.**

Keskkond sisaldab ka rea funktsioone simuleerimistulemuste graafiliseks esituseks.

4. SIGNAALITÖÖTLUSSÜSTEEMIDE MUDELITE KOOSTAMINE

Signaalitöötlussüsteemide uurimisel kasutatavad mudelid koosnevad reeglina kolmest osast: stiimulsignaalide allikast, töötlevast skeemist ja registreerivast seadmest. Teiselt poolt, katse läbiviimisel saab eristada katsetatavat mudeli osa ja katsekeskkonda. Selline liigitus on mõnevõrra tinglik, kuna vaadeldavas modelleerimiskeskkonnas kuuluvad nii katsetatav mudeli osa kui ka katsekeskkond mudeli koosseisu. Samuti sõltub mudeli osade eristamine modelleerimise eesmärgist - näiteks saab signaaligeneraatorit kasutada teatud töötlust sooritava mudeli osa katsetamisel stiimulsignaali allikana, mõnel teisel juhul võib see ise olla katsetuste objektiks.

Eraldi plokina sisaldab mudel reeglina mudeliaja allika - taimeri - millega sünkroniseeritakse teiste plokkide tööd ning formeeritakse signaaliallikate ja töödeldavate signaalide ajabaas. Kuna mudelis globaalaja mõistet ja globaalseid muutujaid ei kasutata, peavad kõik sünkroniseerimist vajavad plokid omama taimeriga ühendatava ajasisendi.

Simuleerimist juhitakse samuti mudeli plokkide kaudu. Simuleerimise käivitamiseks tuleb näidata väljundplokk, mille parameetrid simuleerimise tulemusena soovitakse leida. Väljundplokk koos kõigi selle sisenditega ühendatud plokkidega, nende plokkide sisenditega ühendatud plokkidega jne. moodustab signaalitee. Simuleerimises osalevad ainult signaaliteesse kuuluvad plokid. Samas mudelis võib valida erinevate katsete käigus erinevaid väljundplokke, nendele vastavad signaaliteed on sel juhul samuti erinevad.

Näide 4.1. Harmoonilise signaali generaatori uurimine

Lihtsa näitena võib vaadata taimerist, harmoonilise signaali allikast ja tabuleerivast seadmest koosnevat mudelit harmoonilise signaali genereerimiseks ning selle uurimiseks. Kirjeldame esmalt mudelis kasutatavaid plokitüüpe.

Taimeri TIMER parameetriteks on diskreetimisperiood DT ja

diskreetaja jooksev väärtus T (joon.4.1). Reeglina algab diskreetaeg igal simuleerimisel nullist, kuid vajadusel võib kirjeldada ka nullist erineva diskreetaja algväärtuse.

```
/*+
     TIMER

Sisend:
     ---
Väljund:
     T - diskreetaja jooksev väärtus
Parameetrid:
     DT - diskreetimissamm
     TINIT - diskreetaja algväärtus simuleerimise alustamisel
Kirjeldus:
     T(0)=TINIT
     T(I+1)=T(I)+DT
-*/

out float t; /* Taimeri väljund on reaalarvu formaadis */
float tinit,dt; /* simuleerimise algaeg ja aja samm */

create {        /* Ploki loomisel täidetav algoritm */
     tinit=0;  /* Simuleerimise algusaja vaikimisi väärtus */
     dt=1;      /* Diskreetimisperioodi vaikimisi väärtus */
}

init  {         /* Iga simuleerimise alguses täidetav algoritm */
     t = tinit;
}

calc· {         /* Taimeri simuleerimisel täidetav algoritm */
     t = t + dt;
}
```

Joon.4.1. Taimer TIMER.BLO

Ploki kirjelduse esimestes ridades märkide /*+...-*/ on ploki HELP-kaader. Sellele järgneb tüübideklaratsioonide osa, kus kirjeldatakse ploki sisendeid, väljundeid ja parameetreid. Edasi järgnevad simuleerimise erinevatel etappidel täidetavate algoritmide kirjeldused. CREATE-algoritm algväärtustab ploki loomisel simuleerimise algusaja TINIT ja diskreetimisperioodi DT. Neid väärtusi saab soovi korral enne simuleerimise käivitamist muuta. INIT-algoritm algseadistab taimeri jooksva aja T iga simuleerimise alguses. CALC-algoritm suurendab igal järgneval simuleerimistaktil taimeri jooksvat aega diskreetimisperioodi DT võrra.

Väljundplokina kasutatakse plokki TABLE, mis leiab signaali sisendil Y sõltuvana ajast sisendil X kokku POINTS diskreetaja väärtusel ja väljastab selle tabelina (joon.4.2).

```
/*+
     TABLE
Sisendid:
     x - aja sisend
     y - signaali sisend
Parameeter:
     points - punktide arv
Kirjeldus:
     Tabuleeritakse sõltuvus Y(X) kokku POINTS
     diskreetaja väärtusel
-*/
in float x,y;  /* sisendite x,y kirjeldamine */
float points;  /* simuleerimise punktide arv */
string xtext,ytext; /* sisendite nimetused */
int i;         /* diskreedi loendur */

create  {
     points=100;
     xtext="Aeg";
     ytext="Signaal";
}

calc  {
     print xtext,tab(20),ytext,"\n\n";
     for(i=0;i<points;i++)  {
          get x;
          get y;
          print x,tab(20),y,"\n";
     }
}
```

Joon.4.2. Tabulaator TABLE.BLO

Sellel plokil on kaks sisendit: ajasisend X ja signaalisisend Y. Kuivõrd vaadeldavat plokki kasutatakse reeglina simuleerimise väljundplokina ja tulemused väljastatakse ekraanile, siis plokil väljundid puuduvad. Algoritmi täitmise käigus algseadistatakse tsükliloendur ja trükitakse tabeli pealkiri. Seejärel saadakse kokku POINTS arvupaari sisenditelt X,Y ja kuvatakse nende väärtused ekraanil tabelina.

Viimasena kirjeldame uuritava signaali allika. Sellena

kasutame harmoonilise signaali generaatorit GENCOS (joon.4.3).

```
/**
   Koosinussignaali generaator

Sisend:
    float inp - aja sisend
Väljund:
    float outp - signaali väljund
Parameetrid:
    float freq - signaali sagedus
    float ampl - signaali amplituud
    float phase - signaali algfaas
Kirjeldus:
    outp = ampl * cos(2*pi*(inp*freq + phase/360));
-*/
in float inp;
out float outp;
float phase;
float freq;
float ampl;
float twopi;

create  {
     phase = 0;
     freq = 0;
     ampl=1;
     twopi=2*4*atan(1);
 }

calc  {
     get inp;
     outp = ampl * cos(twopi*(inp*freq + phase/360.0));
 }
```

Joon.4.3. Signaaligeneraator GENCOS.BLO

Vaadeldaval plokil on ajasisend INP ja signaaliväljund OUTP. Signaalidiskreedid väljundil OUTP arvutatakse arvestades mudeliaja väärtust sisendil INP.

Nüüd, peale plokkide kirjeldamist, võime koostada taimerist TIMER, signaaliallikast GENCOS ja tabulaatorist TABLE koosneva mudeli. Koostame esmalt tekstiredaktori abil mudeli kirjelduse (joon.4.4).

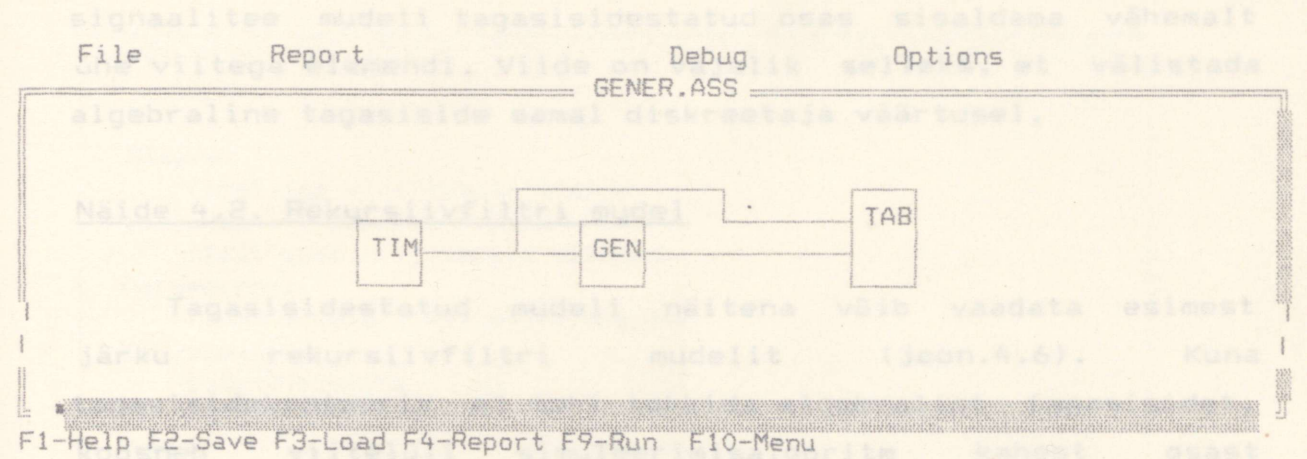
```
/* Harmoonilise signaali generaatori uurimine */

/* Plokkide moodustamine */

block timer tim;
block gencos gen;
block table tabl;

/* Plokkide vaheliste ühenduste kirjeldamine */

connect gen.inp to tim.t;
connect tabl.x to tim.t;
connect tabl.y to gen.outp;

/* Plokkide parameetrite väärtustamine */

tabl.points=64;
tim.dt=1./tabl.points;
gen.ampl=1;
gen.freq=2;

/* Simuleerimise käivitamine */

?tabl;
```

Joon.4.4. Mudeli kirjeldus GENER.SIM

Kasutatavatel parameetrite väärtustel genereeritakse ühe sekundi pikkuse simuleerimisaja jooksul kaks harmoonilise signaali täisperioodi ja tabuleeritakse signaali hetkväärtused. Nii taimeri diskreetimisperioodi kui ka generaatori sagedust saab väärtustada vastavalt modelleeritavale protsessile. Nii näiteks võib moodemi simuleerimisel, kus kandevsagedus on 1.2 kHz ja soovitakse leida kaheksa diskreeti kandevsageduse perioodi kohta, väärtustada parameetrid järgmiselt:

```
gen.freq=1.2;
tim.dt=1./gen.freq/8
```

Mudeli võib koostada ka dialoogreziimis, graafilist plokk-skeemi redaktorit kasutades (joon.4.5).



Joon.4.5. Generaatori katsetamine dialoogreziimis

Vaadeldud näite abil saab selgitada simuleerimise ajastamist hargneva funktsionaalskeemi korral. Taimeri väljundsignaal on nii generaatori kui tabulaatori sisendsignaaliks. Kuigi mõlemast plokist käiakse taimeri väljundsignaali küsimas, käivitub taimeri simuleerimisalgoritm samal diskreetaja väärtusel ainult ühel korral. Ploki kord leitud väljundväärtus on samal diskreetajal kasutatav kõigi selle väljundiga ühendatud plokkide sisendväärtusena. Vastupidisel juhul ei toimuks generaatori ja tabulaatori simuleerimine samadel taimeri diskreetaja väärtustel ja taimeri aeg suureneks kirjeldatuga võrreldes kahekordse kiirusega.

Eelpool kirjeldatud lihtsas näites juhitakse simuleerimise käiku tabulaatori parameetri POINTS (määrab simuleerimise punktide arvu) ja taimeri parameetrite DT, TINIT (määravad mudeliaja sammu ja selle algväärtuse) kaudu.

Sageli on otstarbekas simuleerimise punktide arv seostada signaali allikaga - näiteks juhul, kui signaal loetakse failist või mõõtesüsteemi liidesest. Paljud plokid (näiteks integraator) vajavad diskreetimisperioodi väärtust. Seetõttu on otstarbekas lisada taimerile lisaks jooksva diskreetaja väljundile ka diskreetaja algväärtust, diskreetimisperioodi ja kestust kirjeldava andmemassiivi väljund OUTSIG[] (fail TIMER.BLO lisas 2).

Tagasisidestatud süsteemide modelleerimisel peab

signaalitee mudeli tagasisidestatud osas sisaldama vähemalt ühe viitega elemendi. Viide on vajalik selleks, et välistada algebraline tagasiside samal diskreetaja väärtusel.

Näide 4.2. Rekursiivfiltri mudel

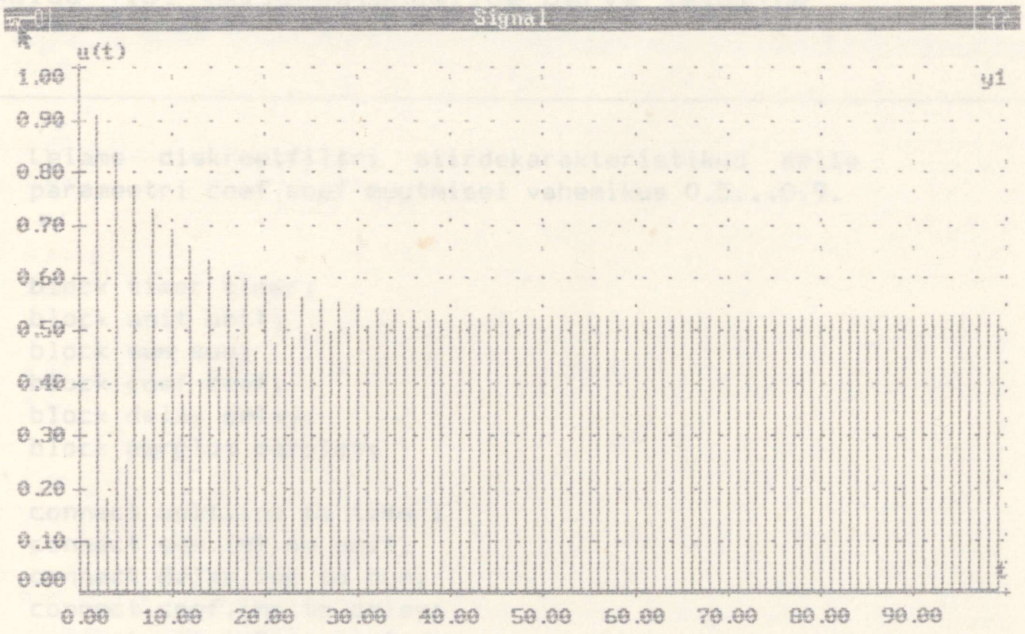
Tagasisidestatud mudeli näitena võib vaadata esimest järku rekursiivfiltri mudelit (joon.4.6). Kuna tagasisidekontuuris ei tohi tekkida algebralist tagasisidet, koosneb viitelüli simuleerimisalgoritm kahest osast (joon.4.7): algoritm CALC täidetakse igal diskreetaja väärtusel, algoritm NEXT täidetakse iga kahe järjestikuse diskreetaja väärtuse vahel. Simuleerimise tulemused esitatakse graafiliselt plotteri PLOT abil (joon.4.8). Mudelis kasutatavate plokkide kirjeldused on toodud lisas 2.

```
/* Esimest järku rekursiivfiltri modelleerimine */

block timer timer;
block unit unit;
block sum sum;
block coef coef;
block delay delay;
block plot plot;

connect unit.inp to timer.outp;
connect sum.in1 to unit;
connect delay.inp to sum;
connect coef.inp to delay;
connect sum.in2 to coef;

connect plot.insig to timer.outsig;
connect plot.y1 to sum;

timer.points=100;
coef.coef=-0.9;            /* filtri koefitsient */

?plot;
```

Joon.4.6. Esimest järku rekursiivfiltri mudel

```
/*+
   DELAY

Sisend:
    float inp - viitelüli sisend
Väljund:
    float outp - viitelüli väljund
Parameeter:
    float istate - viitelüli algolek
Kirjeldus:
    outp(0)=istate
    outp(i+1)=inp(i)
-*/

in float inp;
out float outp;
float istate,state;

create {
     istate=0;
}

init  {
     state=istate;
}

calc  {
     outp=state;
}

next  {
     get inp;
     state=inp;
}
```

Joon.4.7. Viitelüli DELAY.BLO



Joon.4.8. Diskreetfiltri väljundsignaal

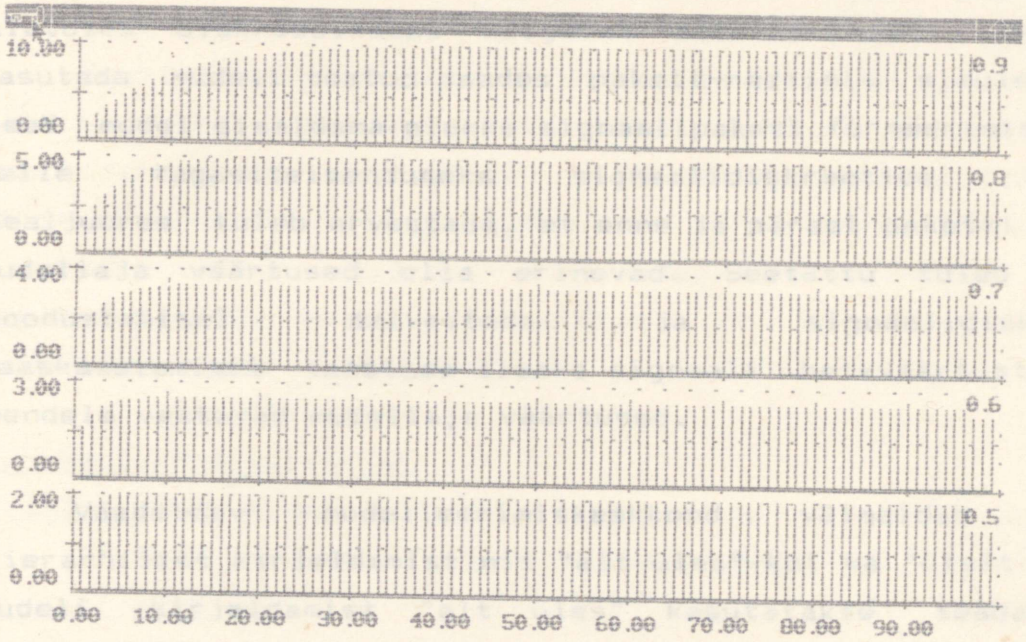
Ülalvaadeldud näidetes leiti mudeli väljundsignaal selle parameetrite püsiväärtustel. Vajadusel võib uurida mudeli käitumist selle teatud parameetri varieerimisel, sooritades selleks rea kordussimuleerimisi. Tulemusena võib saada erinevate parameetri väärtustele vastavate väljundsignaalide parve või parameetrilise sõltuvuse. Mõlemal juhul varieerib mudeli väljundplokk aadressi kaudu viidatavat mudeli parameetrit etteantud vahemikus lineaarse sammuga. Näitena võib uurida eelpool kasutatud diskreetfiltri siirdekarakteristiku sõltuvust selle koefitsiendist (joon.4.9,4.10).

Soovi korral võib koostada filtri uurimisel kasutatava mudeli dialoogreziimis (joon.4.11). Mudelil võib olla mitu väljundplokki, sel juhul väljundploki valik määrab simuleerimise signaalitee ja järelikult ka simuleerimise käigu (joon.4.8 ploki PLOT valikul, joon.4.10 ploki PARPLOT valikul).

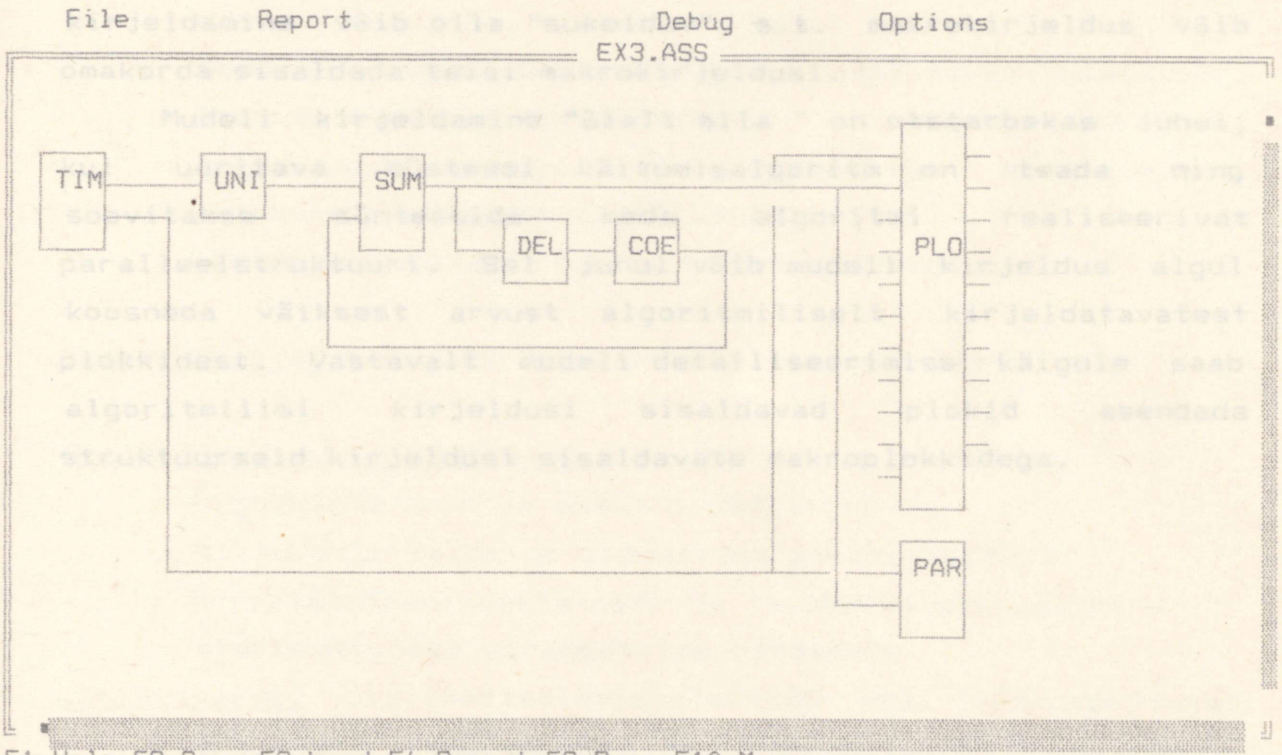
Näide 4.3. Väljundsignaalide parve leidmine

```
/*
Leiame diskreetfiltri siirdekarakteristikud selle
parameetri coef.coef muutmisel vahemikus 0.5...0.9.
*/

block timer timer;
block unit unit;
block sum sum;
block coef coef;
block delay delay;
block parplot parplot;

connect unit.inp to timer;
connect sum.in1 to unit;
connect delay.inp to sum;
connect coef.inp to delay;
connect sum.in2 to coef;

connect parplot.insig to timer.outsig;
connect parplot.y to sum;

parplot.par=&coef.coef;
parplot.parmin=0.5;
parplot.parmax=0.9;
parplot.curves=5;
timer.points=100;
coef.coef=0.9;

?parplot;
```

Joon.4.9. Rekursiivfiltri siirdekarakteristiku sõltuvuse leidmine filtri koefitsiendist



Joon.4.10. Filtri siirdekarakteristiku sõltuvus filtri koefitsiendist



Joon.4.11. Rekursiivfiltri katsetamine dialoogreziimis

Ülalvaadeldud näited illustreerivad kogu signaalitee ulatuses diskreedikaupa kulgevat simuleerimist. Vajadusel kasutada mudeli teatud osades pakett-reziimis simuleerimist peab mudel sisaldama plokke signaalipaketi formeerimiseks ja selle tagasiteisenduseks signaalidiskreetide kujule. Sealjuures tuleb arvestada, et enne ja pärast paketti võivad mudeliaja väärtused olla erinevad. Seetõttu tuleb paketi moodustamisel salvestada ja signaalidiskreetide taaskasutamisel taastada lisaks signaali hetkväärtustele ka nendele vastavad mudeliaja väärtused.

Vaadeldav modelleerimiskeskkond võimaldab mudeli hierarhilist kirjeldamist nii "alt üles" kui ka "ülalt alla ". Mudeli kirjeldamist "alt üles" kasutatakse teadaolevate struktuuride modelleerimisel. Kord koostatud ja testitud mudelit või selle osa saab kirjeldada teatud sisendeid, väljundeid ning parameetreid omava makroplokina ning kasutada seda edaspidi keerukamate mudelite koostisosana. Makrode kirjeldamine võib olla "sukelduv", s.t. makrokirjeldus võib omakorda sisaldada teisi makrokirjeldusi.

Mudeli kirjeldamine "ülalt alla " on otstarbekas juhul, kui uuritava süsteemi käitumisalgoritm on teada ning soovitakse sünteesida seda algoritmi realiseerivat paralleelstruktuuri. Sel juhul võib mudeli kirjeldus algul koosneda väiksest arvust algoritmiliselt kirjeldatavatest plokkidest. Vastavalt mudeli detailiseerimise käigule saab algoritmilisi kirjeldusi sisaldavad plokid asendada struktuurseid kirjeldusi sisaldavate makroplokkidega.

5. LABORATOORSETE TÖÖDE NÄIDISKIRJELDUSED

Funktsionaalskeemide simulaatorit on kasutatud raadiotehnika kateedri õppetöös kahe aasta vältel raadiotehniliste signaalide ja ahelate, raadioautomaatika, telekommunikatsioonisüsteemide ja raadiosüsteemide laboratoorsetes töödes, kursuseprojekteerimises ning diplomprojekteerimises. Koostatud on mudelid järgmiste ainevaldkondade uurimiseks:

1. Signaalide uurimine.

Eesmärk: tutvumine erinevate signaalide kirjeldusviisidega, nende parameetritega ning omadustega

Signaaliallikad:

- regulaarsete signaalide allikad (alaliskomponent, harmooniline signaal, impulss-signaal, murdlineaarne signaal, eksponentsignaal)
- juhuslike signaalide allikad (ühtlase jaotusega, normaaljaotusega, binaarjaotusega) .
- moduleeritud signaalide allikad (amplituudmodulatsioon, balanss-amplituudmodulatsioon, faasmodulatsioon, sagedusmodulatsioon, ühe külgribaga modulatsioon, erinevad impulss-modulatsioonid)
- mõõtesüsteemist ja kettalt loetavad signaalid

Muundusplokid: summaatorid, korrutid, integraatorid jm.

Registreerivad ja töötlevad plokid:

- signaalide kuju ja spektri jälgimine
- sõltuvuste leidmine signaalide parameetritest
- korrelatsioonifunktsiooni ja jaotusseaduse uurimine
- energeetiliste parameetrite hindamine

Lühiülevaade. Ülalloetletletud plokkide abil genereeritakse erinevaid regulaarseid ja mitteregulaarseid signaale, uuritakse nende kuju ning spektreid, samuti signaali erinevate parameetrite toimet. Uuritakse ka signaalide autokorrelatsioonifunktsiooni ja jaotusseadust. Vajadusel võib olemasolevatest signaalidest formeerida keerukamaid, kasutades selleks summaatoreid, korruteid,

integraatoreid ja teisi muundusplokke. Vaadeldakse signaalide koosmõju: erinevate signaalide summat ja korrutist ning nende omadusi.

2. Signaalide parameetrite struktuurne hindamine. Signaali spektri ja korrelatsioonifunktsiooni mõõtmine.

Eesmärk: tutvumine signaalide parameetrite struktuurse hindamisega

Signaaliallikad:

- p.1 kirjeldatud signaalid
- baasfunktsioonide generaatorid (lihtbaasfunktsioonid, kompleks-eksponentbaasfunktsioonid, Walshi baasfunktsioonid)

Muundusplokid: summaatorid, korrutid, keskmistavad integraatorid, viitelülid

Registreerivad ja töötlevad plokid: signaali kuju ning spektri jälgimine, sõltuvuste leidmine

Lühiülevaade. Korruti ja keskmistava integraatori baasil signaalide võimsuse ja kahe signaali vastastikuse võimsuse hindamine. Kahe signaali ortogonaalsuse ning ortogonaalsusvahemiku hindamine. Spektrikomponentide hindamine. Spektrikomponendi hinnangu vea uurimine sõltuvalt hindamise kestusest. Baasfunktsioonide ortonormeerimine. Mitteharmoonilise perioodilise signaali spektri hindamine. Korruti, viitelüli ja keskmistava integraatori baasil signaalide autokorrelatsiooni ja vastastikuse korrelatsiooni väärtuste hindamine. Korrelatsioonifunktsiooni leidmine sõltuvusena viiteajast.

3. Lineaarsete ahelate mõju uurimine

Eesmärk: tutvumine lineaarsete sagedusselektiivsete ahelate mõjuga neid läbivatele regulaarsetele ja mitte-

regulaarsetele signaalidele.

Signaaliallikad: p.1 kirjeldatud signaaliallikad

Muundusplokid: digitaalfiltrid, diskreetsed integraatorid, summaatorid.

Registreerivad ja töötlevad plokid: signaalide kuju ning spektri jälgimine, energeetiliste parameetrite hindamine, korrelatsiooni ja jaotusseaduse mõõtmine

Lühiülevaade. Erinevate signaalide kuju, spektri, jaotusseaduse ja korrelatsionifunktsiooni muutumine nende läbiminekul erineva läbilaskeribaga filtritest ning diskreetsest integraatorist. Juhusliku signaali normaliseerumine lineaarsetes ahelates. Juhusliku signaali autokorrelatsioonifunktsioon selektiivse ahela väljundis. Perioodilise ja juhusliku signaali summa korrelatsioonifunktsiooni ning spektri uurimine.

4. Mittelineaarse elemendi uurimine

Eesmärk: tutvumine mittelineaarsete elementide mõjuga neid läbivatele signaalidele ning mittelineaarsete elementide kasutamisega signaalide muundamisel.

Signaalid: harmoonilised signaalid, juhuslikud signaalid, alaliskomponent

Muundusseadmed:

- korrutid, summaatorid
- polünomiaalne mittelineaarsus
- murdsirgetega realiseeritud mittelineaarsus
- Tshebõshevi funktsiooni, logaritmi, ruutjuurt ja teisi funktsionaalseid teisendusi realiseerivad plokid
- filtrid

Registreerivad ja töötlevad plokid: signaalide kuju ja spektri jälgimine, sõltuvuste leidmine, jaotusseaduse hindamine

Lühiülevaade. Uuritakse erinevate mittelineaarsete elementide mõju neid läbivate signaalide kujule, spektrile, jaotusseaduse funktsioonile. Tutvutakse mittelineaarsete elementide kasutamisega jaotusseaduse teisendamisel. Tutvutakse mittelineaarsete elementide aproksimeerimisviisidega, vaadeldakse üleminekut murdlineaarse mittelineaarsuse kasutamiselt polünomiaalse mittelineaarsuse kasutamisele. Uuritakse signaalide kordistamist, sageduse muundamist, moduleerimist sõltuvana sisendsignaalide ning mittelineaarsuse parameetritest ning tööpunktist, kontrollitakse teoreetiliselt leitud optimaalseid muundusreziime. Tutvutakse vajalike spektrikomponentide eristamisega filtrite abil. Tutvutakse mittelineaarsete moonutuste hindamisega spektrikomponentide kaudu ja tõkkefiltri kasutamisega.

5. Signaalide digitaalse genereerimise uurimine

Eesmärk: tutvumine digitaalsignaali genereerimisvõimalustega

Muundusplokid: summaatorid, korrutid, registrid, koefitsiendi plokid, integraatorid, Schmidti trigerid, erinevad funktsionaalsed muundurid, filtrid

Registreerivad ja töötlevad plokid: signaalide kuju ja spektri jälgimine

Lühiülevaade. Tutvutakse signaali genereerimisega vastavalt teist järku diferentsvõrrandi lahendile. Tutvutakse faasi summeerimise ning tabelteisenduse abil siinussignaali genereerimisega. Tutvutakse integraatoril, Schmidti trigeril ning siinuse tabelil realiseeritava funktsionaalgeneraatoriga ning hinnatakse selle amplituudi ebatäpsusest põhjustatavaid mittelineaarmoonutusi. Uuritakse erinevate genereerimisviiside mittelineaarmoonutusi sõltuvana diskreetide arvust genereeritava signaali perioodis.

Uuritakse madalpääsfiltri kasutamist diskreetsignaali teisendamisel analoogsignaaliks ning madalpääsfiltri parameetritele esitatavaid nõudeid.

6. Signaalide moduleerimine

Eesmärk: tutvumine signaalide modulatsioonitehnikaga

Signaaliallikad: p.1 kirjeldatud signaaliallikad

Muundusplokid: korrutid, summaatorid, filtrid, mittelineaarsed elemendid, funktsionaalsed muundurid, faasinihkelülid

Registreerivad ja töötlevad plokid: signaali kuju ja spektri jälgimine, sõltuvuste leidmine

Lühiülevaade.

Balanssamplituudmodulaatorite realiseerimine:
- korruti baasil
- mittelineaarse elemendi ja filtri baasil
- mittelineaarse elemendiga kompensatsioonlülituse baasil

Kesksagedusega AM-modulaatori realiseerimine:
- korruti ja summaatori baasil
- mittelineaarse elemendi baasil
- kompensatsioonlülituse uurimine
- ringmodulaatori uurimine

Ühepoolse spektriga AM-modulaatorite realiseerimine:
- korrutite ja filtrite baasil
- kvadratuurkanalite baasil
- faas-filtermeetodil

Polaar-AM-signaalide modulaatorite realiseerimine.

Faasmodulaatorite ja sagedusmodulaatorite realiseerimine:
- AM-modulaatorite baasil
- digitaalse generaatori baasil

Sagedusmanipulaatorite realiseerimine.

Faasmanipulaatorite realiseerimine.

Impulss-modulaatorite realiseerimine.

Uuritakse erinevate modulaatorite väljundsignaale ja nende spektreid, võrreldakse erinevate modulaatorite realiseeritavust.

7. Sünkroondetektorite uurimine

Eesmärk: tutvumine sünkroondetektorite realiseerimisvõimalustega ja nende kasutamisega

Signaaliallikad: plokid ja makroplokid erinevate modulatsiooniviisidega signaalide genereerimiseks

Muundusplokid: korrutid, summaatorid, filtrid, faasinihkelülid, viitelülid

Registreerivad ja töötlevad plokid: signaali kuju ja spektri jälgimine, sõltuvuste leidmine

Lühiülevaade.

Faasitundliku sünkroondetektori realiseerimine:

- korruti ja filtri baasil
- mittelineaarse elemendi ja filtri baasil
- piirikute, korruti ja filtri baasil
- faasinihkeahelaga ja korrutiga kvadratuurkanalite baasil
- viitelüliga ja korrutiga kvadratuurkanalite baasil.

Faasitundetu sünkroondetektori realiseerimine:

- korrutite, filtrite ja aritmeetikaplokiga kvadratuurkanalite baasil
- korrutite, filtrite ja amplituuddetektoriga kvadratuurkanalite baasil
- korrutite, Hilberti muunduri ja aritmeetikaplokiga kvadratuurkanalite baasil
- korrutite ja viiteliinidega kvadratuurkanalite baasil.

Uuritakse sünkroondetektori omaduste sõltuvust selle struktuurist. Hinnatakse filtritele, faasinihke-

ahelatele, viitelülidele esitatavaid nõudeid, uuritakse nende mitteideaalsuse mõju detektori tööle. Uuritakse sünkroondetektori häirekindlust ja detekteerimiskarakteristikuid. Tutvutakse sünkroondetektorite kasutamisega AM, ÜAM, FM, faasmanipuleeritud ning sagedusmanipuleeritud signaalide detekteerimisel. Tutvutakse sünkroondetektori tugisignaali formeerimisega.

8. Signaalide optimaalne vastuvõtt ja optimaalne eristamine

Eesmärk: tutvumine signaalide optimaalse avastamisega ja optimaalse eristamisega

Signaaliallikad: p.1 kasutatavad signaaliallikad

Muundusplokid: summaatorid, korrutid, filtrid, funktsionaalsed muundurid, integraatorid, konvolutsiooni plokk

Registreerivad ja töötlevad plokid: signaali kuju ja spektri jälgimine, korrelatsiooni ja konvolutsiooni leidmine

Lühiülevaade. Optimaalsel signaalide avastamisel ja eristamisel kasutatavate signaalide ja nende omadustega (spekter, AKF) tutvumine. Laia baasiga signaalide ja perioodiliste moduleeritud impulssjadade formeerimine ja nende omadused. Signaalide eristamine signaali ja müra segust nende spektri ja korrelatsioonifunktsiooni kaudu. Korrelaatori kasutamine optimaalse avastajana ja eristajana. Eristamisel kasutatavate signaalide valik. Avastamise ja eristamise signaalinivoode määramine. Häirekindluse uurimine. Tugisignaali parameetrite vigade (sageduse nihe, viide) mõju uurimine.

9. Analoogsignaali diskreetimine. Analoogsignaali taastamine. Diskreetimise ja kvanteerimise mõju uurimine.

Eesmärk: tutvumine analoogsignaali diskreetimisega ja kvanteerimisega ning diskreetsignaalist pidevsignaali taastamisega

Signaaliallikad: p.1 kasutatavad signaaliallikad

Muundusplokid: diskreetivad ja kvanteerivad plokid, filtrid

Registreerivad ja töötlevad plokid: signaali kuju jälgimine, sõltuvuste leidmine

Lühiülevaade. Harmoonilise signaali diskreetimine ja taastamine. Diskreetimissageduse ja filtri läbilaskeriba valik, nende ebaõigest valikust tekkivate vigadega tutvumine. Signaali lineaarne ja logaritmiline kvanteerimine. Kvanteerimismüra uurimine. Lõpmatu spektriga signaalide diskreetimine.

10. Diskreetsüsteemide uurimine

Eesmärk: tutvumine signaalitöötlusel kasutatavate diskreetsüsteemide omadustega ja nende kasutamisega

Signaaliallikad: p.1 kasutatavad signaaliallikad.

Muundusplokid: viitelülid, summaatorid, koefitsiendi plokid

Registreerivad ja töötlevad plokid: signaali kuju ja spektri jälgimine, sõltuvuste leidmine

Lühiülevaade. Mitterekursiivse ja rekursiivse filtri mudeli koostamine ja nende omaduste uurimine. Filtri parameetrite normeerimine. Kvanteerimisvigade uurimine. Kanoonilise filtri struktuuri realiseerimine, kirjeldamine makrona. Filtrite paralleelühenduse ja järjestikühenduse uurimine. Tõkkefiltri ja ribafiltri sünteesi teist järku kanoonilise struktuuri baasil, selle kasutamine sisendsignaali filtreerimisel. Hilberti muunduri realiseerimine ja uurimine. Hilberti muunduri sagedussõltuvuse uurimine. Viitelüli kasutamine kitsaribalise Hilberti muundurina.

11. Diskreetse infokanali modelleerimine

Eesmärk: tutvumine diskreetse järjestikinfokanali modelleerimisega

Signaaliallikad: tekstisümbolite allikas, juhusliku signaali allikas

Muundusplokid: sümboli teisendaja järjestikkoodi, järjestikkoodi teisendaja sümboliks, kanali mudel, summaatorid

Registreerivad ja töötlevad plokid: sümbolite vastuvõtja

Lühiülevaade. Informatsiooni ülekande uurimine etteantud bitivigade tekkimise tõenäosusel. Paarsuse kontrolli kasutamine.

12. Impulss-koodmodulatsioon

Eesmärk: tutvumine digitaalsignaali edastusel kasutatavate modulatsiooniviisidega

Signaaliallikad: tekstisümbolite allikas, juhusliku signaali allikas

Muundusplokid: sümboli teisendaja järjestikkoodi, järjestikkoodi teisendaja suhtelisse järjestikkoodi, modulaatorid NRZP, NRZU, RZP, RZU, Manchesteri, duobinaarsesse ja modifitseeritud duobinaarsesse koodi; summaatorid, filtrid, komparaatorid, viitelülid, plokid erinevat tüüpi ja erineva takteerimissagedusega andmevoogude sobitamiseks

Registreerivad ja töötlevad plokid: signaali kuju ja spektri kuju jälgimine, sõltuvuste leidmine

Lühiülevaade. Erinevatel modulatsiooniviisidel kasutatavate signaalide, nende spektrite, ribalaiuse ja häirekindluse võrdlemine. Kanali mõju uurimine selle

väljundsignaalile. Sünkrosignaali ja digitaalsignaali taastamine erinevate modulatsiooniviiside korral.

13. Faasmanipulatsiooni kasutamine infoedastusel

Eesmärk: tutvumine suhtelise faasmanipulatsiooni kasutamisega moodemites

Signaaliallikad: tekstisümbolite allikas, juhusliku signaali allikas

Muundusplokid: sümboli teisendaja järjestikkoodi, järjestikkoodi teisendaja suhtelisse järjestikkoodi, faasmanipulaator, summaatorid, filtrid, komparaatorid, viitelülid, järjestikkoodi teisendaja sümboliks, plokid erinevat tüüpi ja erineva takteerimissagedusega andmevoogude sobitamiseks

Registreerivad ja töötlevad plokid: signaali kuju ja spektri kuju jälgimine, sõltuvuste leidmine

Lühiülevaade. Moodemite tööpõhimõttega tutvumine. Kanali mõju ja häirekindluse uurimine.

Tööde nomenklatuur on hõlpsalt laiendatav vastavalt laboratoorsete tööde vajadustele. Üliõpilased saavad kasutada eelnevalt koostatud signaalitöötlussüsteemide mudeleid, samas võivad nad neid vastavalt vajadusele muuta ning täiendada või koostada ise uusi mudeleid.

Näitena võib tuua mõne laboratoorse töö detailsema kirjelduse.

5.1. Suhtelise faasmanipulatsiooniga moodem

Töö eesmärk: suhtelist faasmanipulatsiooni kasutavate moodemite tööpõhimõtte ja häirekindlusega tutvumine.

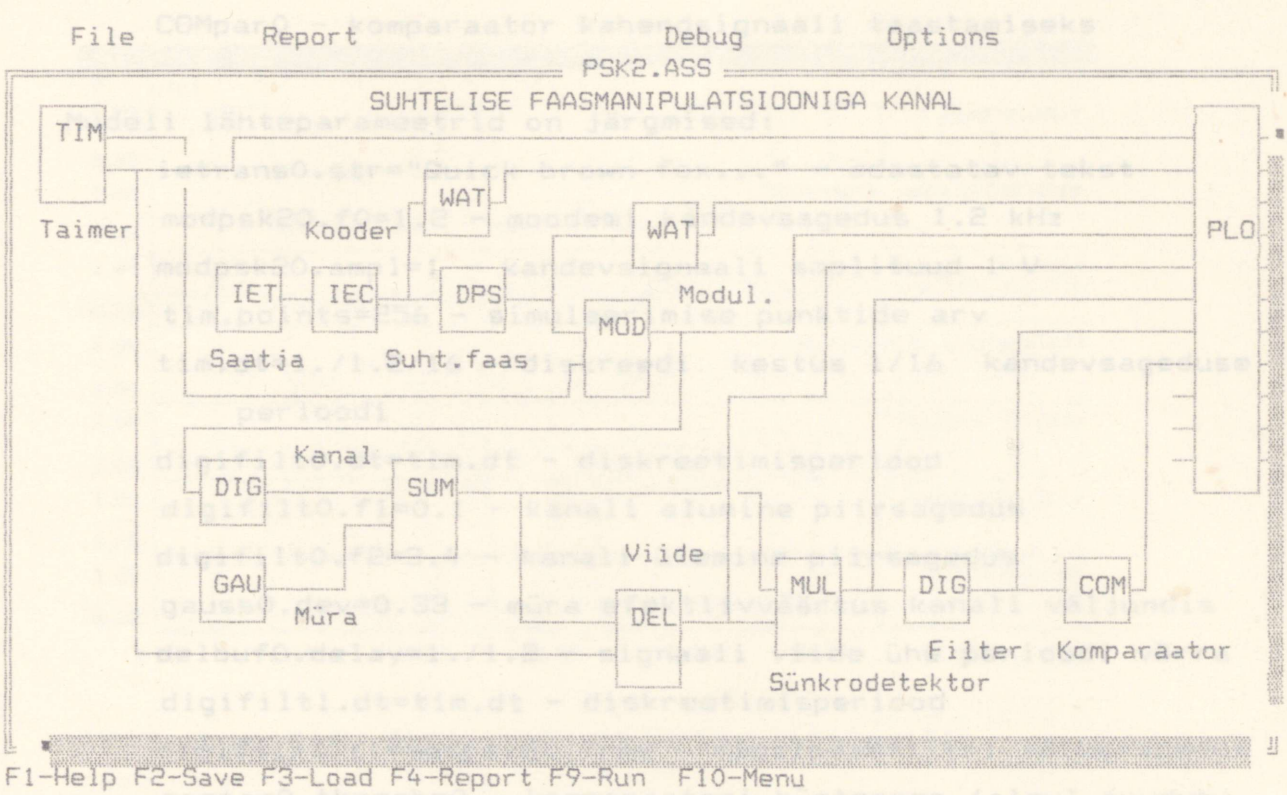
Töövahendid: MS-DOS arvuti, programm SIMULANT

Ettevalmistus tööks:

1. Tutvuge suhtelise faasmanipulatsiooni tööpõhimõttega.
2. Tutvuge ASCII kooditabeliga.
3. Tutvuge asünkroonse infoedastuse põhimõtetega.
4. Kodeerige perekonnanime kolm esimest tähte asünkroonsel infoedastusel kasutatavasse järjestikkoodi ja teisendage see suhtelise bitimuutusega koodi. Ärge unustage lisamast start- ja stopp-bitte ning paarsuse kontrolli bitte.

Töö käik.

Käivitage programm SIMULANT. Viige see dialoogreziimi ja laadige mudel PSK2 (2-Phase Shift Keying). Ekraanil joonistub mudeli struktuurskeem järgmiste plokkidega (joon. 5.1):



Joon. 5.1. Moodemiga infokanali mudel

Üldplokid:

TIM - ajaskaala formeerimine, teiste plokkide töö sünkroniseerimine

PLOt0 - plotter

WATch0,WATch1 - plotteri sisendsignaali sobitamine

Saatja mudel:

IETrans0 - edastatava teksti teisendamine baitide jadaks

IECode0 - baitide teisendamine järjestikkoodi

DPSk0 (Differential Phase Shift Keying) - koodi teisendamine suhtelistele bitimuutustele

MODpsk20 (Modulator 2-Phase Shift Keying) - faasmanipulaator

Kanali mudel:

DIGifilt0 - kanali läbilaskeriba imiteeriv ribafilter

GAUss0 - normaaljaotusega valge müra kanalis

SUM0 - signaali ja müra summa formeerimine

Vastuvõtja mudel:

DELbuf0 - tugisignaali viitelüli

MULt0 - sünkroondetektori korruti

DIGifilt1 - sünkroondetektori madalpääsfilter

COMpar0 - komparaator kahendsignaali taastamiseks

Mudeli lähteparameetrid on järgmised:

ietrans0.str="Quick brown fox..." - edastatav tekst

modpsk20.f0=1.2 - moodemi kandevsagedus 1.2 kHz

modpsk20.ampl=1 - kandevsignaali amplituud 1 V

tim.points=256 - simuleerimise punktide arv

tim.dt=1./1.2/16 - diskreedi kestus 1/16 kandevsageduse perioodi

digifilt0.dt=tim.dt - diskreetimisperiood

digifilt0.f1=0.1 - kanali alumine piirsagedus

digifilt0.f2=3.4 - kanali ülemine piirsagedus

gauss0.dev=0.33 - müra efektiivväärtus kanali väljundis

delbuf0.delay=1./1.2 - signaali viide ühe perioodi võrra

digifilt1.dt=tim.dt - diskreetimisperiood

digifilt1.f1=modpsk20.freq - madalpääsfiltri lõikesagedus

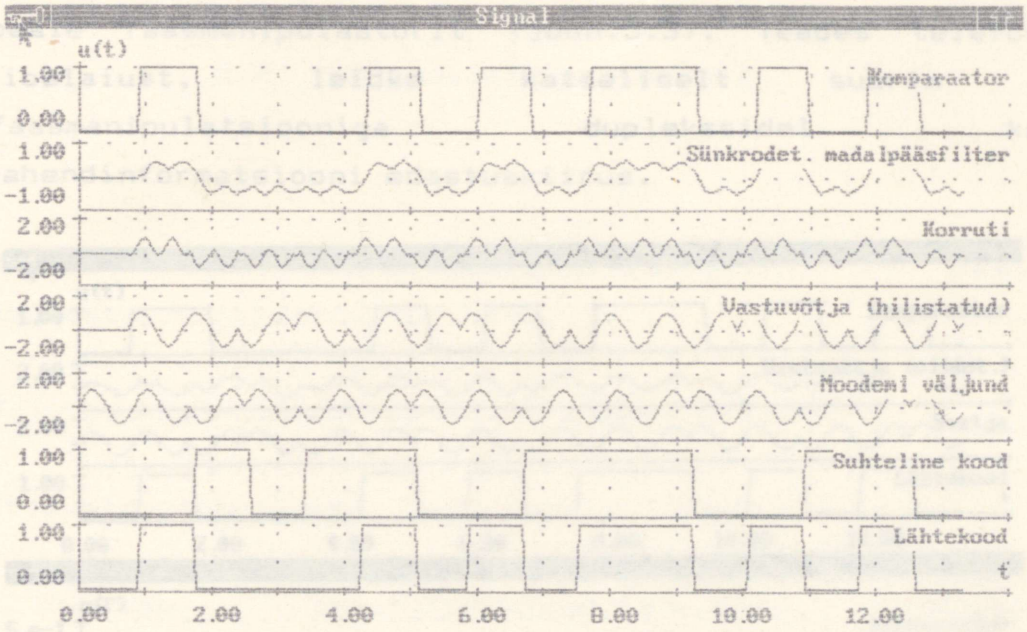
compar0.thresh=0 - komparaatori hüsterees (algul puudub)

Muutke kanali parameetrid algul sellisteks, et need ei avaldaks mõju signaalidele kanalis. Käivitage simuleerimine. Edastage tekstina perekonnanimi. Kontrollige graafikutelt (joon.5.2) ettevalmistavas osas kodeeritud järjestikkoodi õigsust. Jälgige signaale moodemite erinevates punktides ja püüdke aru saada moodemi tööpõhimõttest. Kontrollige komparaatori väljundsignaali vastavust lähtesignaalile. Leidke vastus küsimustele:

1. Kuidas formeeritakse vastuvõtja sünkroondetektori tugisignaal?

2. Kus ja kuidas toimub vastuvõtjas üleminek suhteliselt järjestikkoodilt absoluutsele järjestikkoodile?

3. Mis otstarve on madalpääsfiltril ja komparaatoril?



Joon.5.2. Signaalid moodemiga infokanalis

Seadke kanali läbilaskeriba vahemikule 0.1...3.4 kHz. Jälgige signaali kuju muutust kanali väljundis. Vähendage läbilaskeriba veelgi. Hinnake vastuvõtul tekkivaid vigasid.

Taastage kanali esialgne läbilaskeriba. Muutke edastatava signaali kandevsagedust väikestes piirides. Hinnake sealjuures vastuvõtul tekkivaid vigasid.

Muutke müra nivoo selliselt, et signaal/müra=3. Jälgige signaale vastuvõtja erinevates sõlmedes. Hinnake vastuvõtu vigu.

Vähendage signaal/müra suhet veelgi. Hinnake vigu.

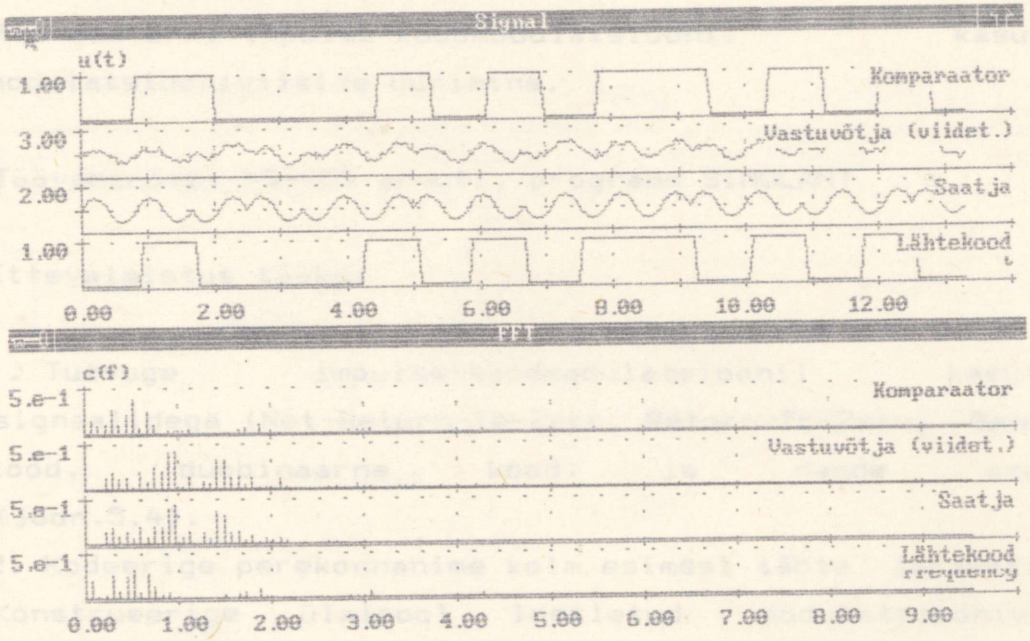
Uurige komparaatori hüstereesi mõju vastuvõetud signaalile. Määrake katseliselt sobiv hüstereesi nivoo.

Leidke ligikaudselt bitivigade tõenäosuse hinnangu sõltuvus signaal/müra suhtest vahemikus s/n=10...s/n=1.

Modulatsiooniviisi häirekindlust iseloomustatakse signaali amplituudväärtuse suhtega müra spektraaltihedusse. Modelleerimisel kasutatakse mürasignaalina juhuslikku normaaljaotusega diskreetsignaali. Teades diskreetimisperioodi, leidke simuleerimisel kasutatud mürasignaali efektiivväärtustele vastavad spektraaltihedused.

Tutvuge faasmanipulatsioonil kasutatavate signaalide spektritega. Selleks kasutage ploki PLOt asemel plokki

FFTplot. Leidke signaalide spektrite sagedusvahemikud enne ja peale faasmanipulaatorit (joon.5.3). Teades telefonikanali ribalaiust, leidke katseliselt suurim kahese faasmanipulatsiooniga duplekssidel kasutatav kahendinformatsiooni edastuskiirus.



Joon.5.3. Signaalid ja nende spektrid moodemiga kanalis

Kahest faasmanipulatsiooni kasutatakse ka pooldupleks-sidel saatja edastuskiirusel 1200 b/s ja vastuskanali kiirusel 300 b/s. Modelleerige saatekanali ja vastuskanali signaalid ning kontrollige nende spektrite ribalaiusi.

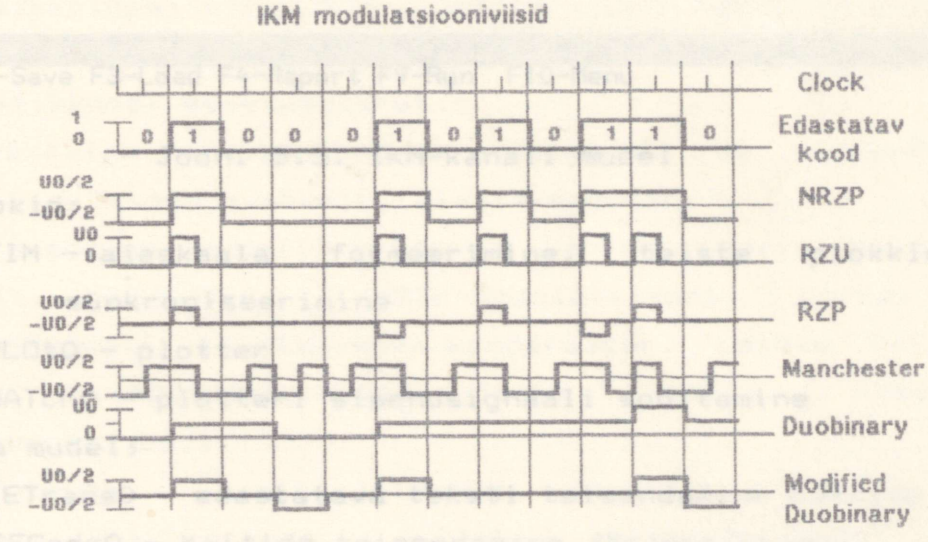
5.2. IKM modulatsiooniviisid

Töö eesmärk: impulss-koodmodulatsioonil kasutatavate modulatsiooniviiside uurimine.

Töövahendid: MS-DOS arvuti, programm SIMULANT

Ettevalmistus tööks:

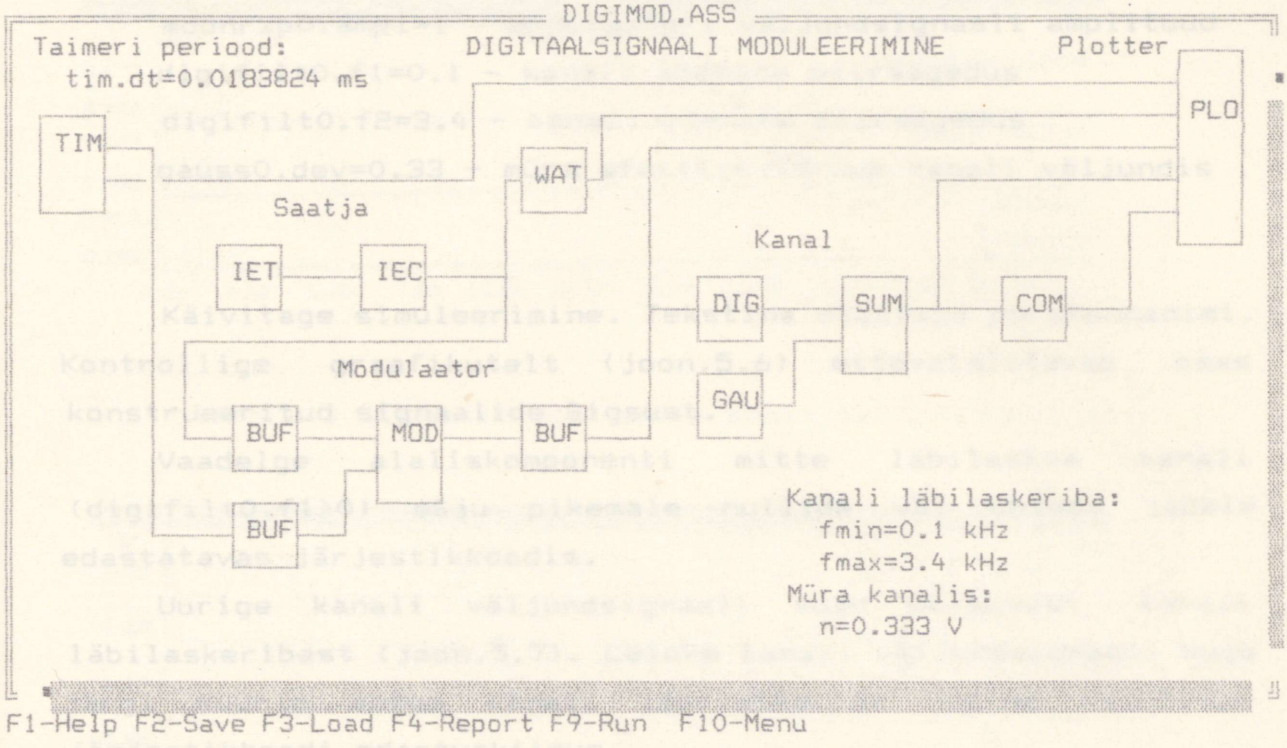
1. Tutvuge impulss-koodmodulatsioonil kasutatavate signaalidega (Not-Return-To-Zero, Return-To-Zero, Manchesteri kood, duobinaarne kood) ja nende omadustega (joon.5.4).
2. Kodeerige perekonnanime kolm esimest tähte järjestikkoodi. Konstrueerige ülalpool loetletud modulatsiooniviisidele vastavate signaalide graafikud.



Joon.5.4. IKM modulatsiooniviisid

Töö käik:

1. Käivitage programm SIMULANT. Viige see dialoogreziimi ja laadige mudel DIGIMOD. Ekraanil joonistub mudeli struktuurskeem järgmiste plokkidega (joon.5.5):



Joon. 5.5. IKM-kanali mudel

Üldplokid:

TIM - ajaskaala formeerimine, teiste plokkide töö sünkroniseerimine

PLOt0 - plotter

WATch0 - plotteri sisendsignaali sobitamine

Saatja mudel:

IETrans0 - edastatava teksti teisendamine baitide jadaks

IECode0 - baitide teisendamine järjestikkoodi

BUFie0,BUFfer0,BUFfer1 - modulaatori puhverdamine (nende vahele saab ühendada erinevaid modulaatoreid)

MODnrzp - NRZP modulaator

Kanali mudel:

DIGifilt0 - kanali läbilaskeriba imiteeriv ribafilter

GAUss0 - normaaljaotusega valge müra kanalis

SUMO - signaali ja müra summa formeerimine

Mudeli lähteparameetrid on järgmised:

ietrans0.str="Quick brown fox..." - edastatav tekst

tim.points=256 - simuleerimise punktide arv

tim.dt=1./3.4/16 - diskreedi kestus 1/16 kanali läbilaskeribast

modnrzp0.f0=3.4 - järjestikkoodi edastuskiirus Kbit/s

modnrzp0.ampl=1 - modulaatori väljundsignaali amplituud

digifilt0.f1=0.1 - kanali alumine piirsagedus

digifilt0.f2=3.4 - kanali ülemine piirsagedus

gauss0.dev=0.33 - müra efektiivväärtus kanali väljundis

Käivitage simuleerimine. Tekstina edastage perekonnanimi. Kontrollige graafikutelt (joon.5.6) ettevalmistavas osas konstrueeritud signaalide õigsust.

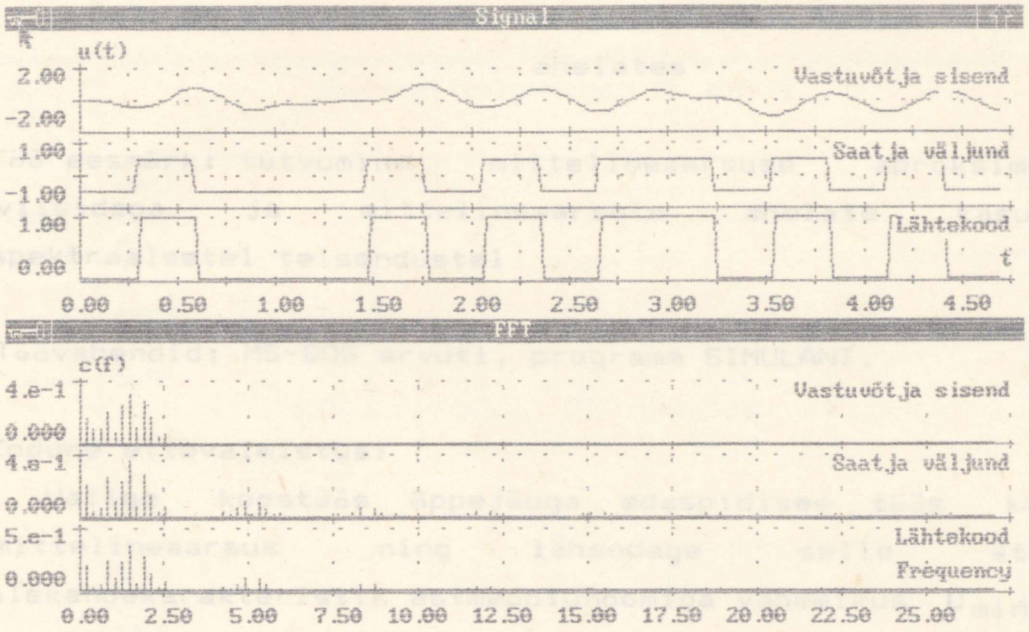
Vaadelge alaliskomponenti mitte läbilaskva kanali (digifilt0.f1>0) mõju pikemale nullide või ühtede jadale edastatavas järjestikkoodis.

Uurige kanali väljundsignaali kuju sõltuvust kanali läbilaskeribast (joon.5.7). Leidke kanali väljundsignaali kuju järgi suurim antud kanali läbilaskeriba juures kasutatav järjestikkoodi edastuskiirus.

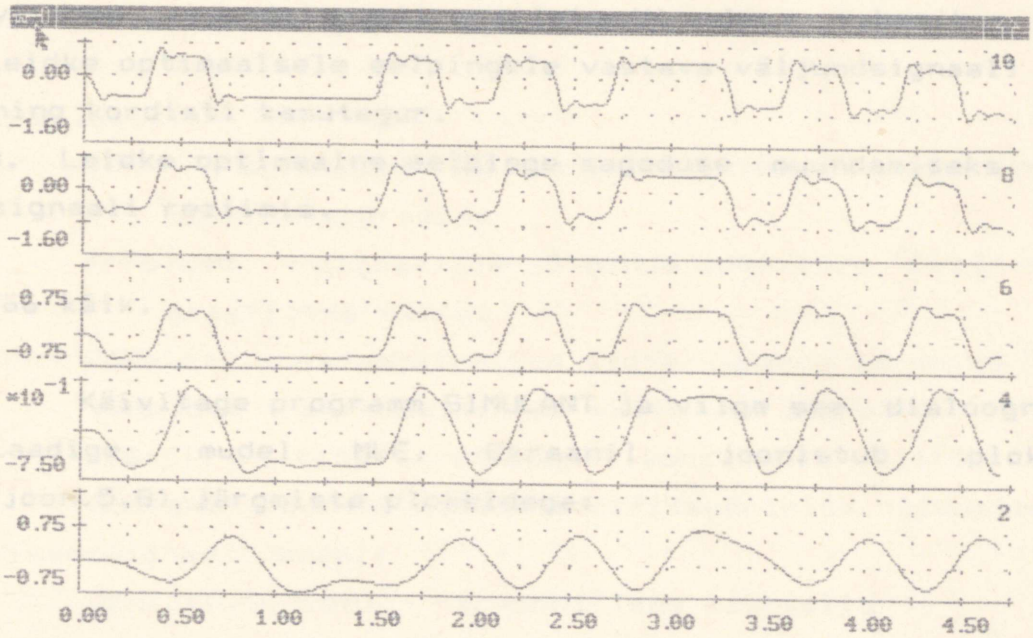
Võrrelge modulaatori sisendsignaali ja väljundsignaali spektreid (sagedusvahemik, alaliskomponent).

Täiendage mudelit skeemiga kahendsignaali taastamiseks kanali väljundis. (NRZP-modulatsiooni korral sobib kahendsignaali taastamiseks komparaator. Leidke katseliselt kanali väljundsignaali jaoks sobiv komparaatori hüstereesinivoo.)

Tutvuge ülalkirjeldatud viisil ka teiste modulatsiooniviisidega. Selleks asendage mudelis olev modulaator teist tüüpi modulaatoriga ja korrake ülalkirjeldatud uuringuid. Võrrelge erinevate modulatsiooniviiside suurimaid järjestikkoodi edastuskiiruseid.



Joon.5.6.Signaalid ja nende spektrid IKM-kanalis



Joon.5.7. Signaali sõltuvus
kanali läbilaskeribast

5.3. Spektraalteisendused mittelineaarsetes ahelates

Töö eesmärk: tutvumine mittelineaarsuse aproksimeerimisviisidega ja mittelineaarsete ahelate kasutamisega spektraalsetel teisendustel

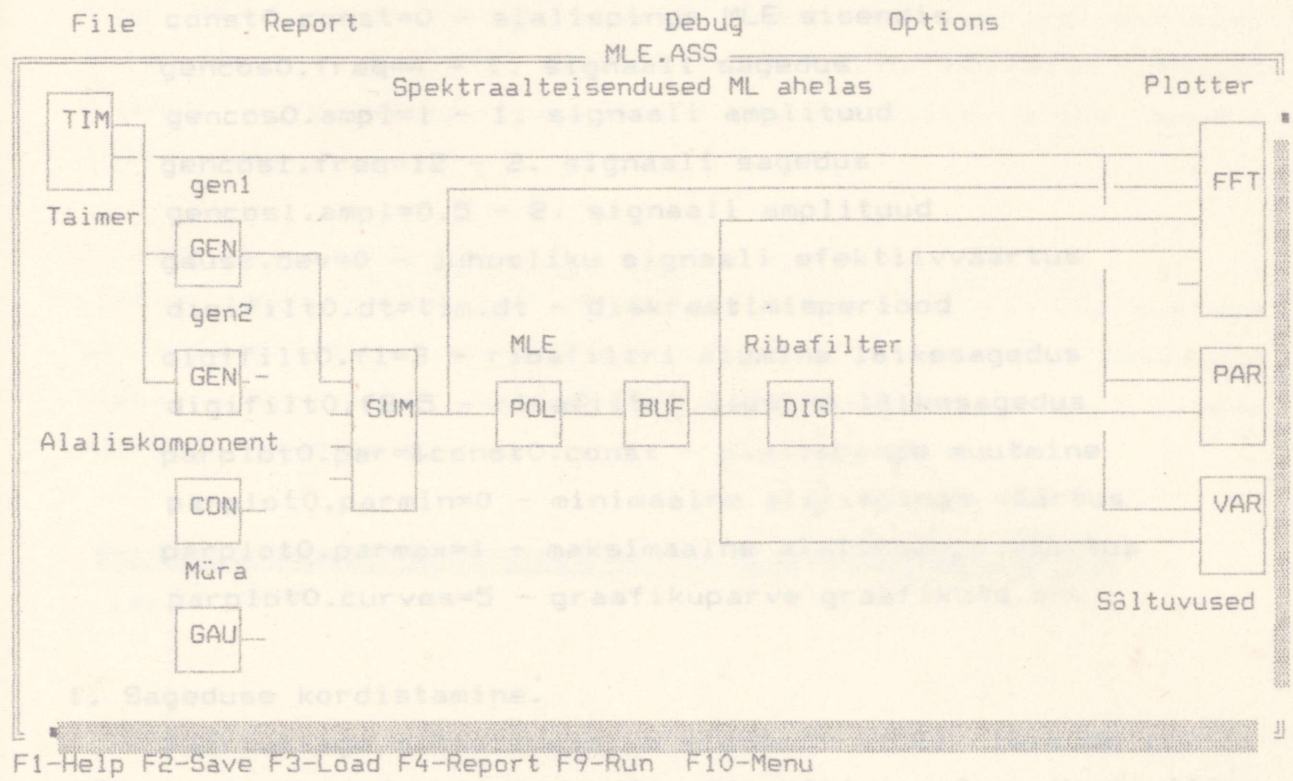
Töövahendid: MS-DOS arvuti, programm SIMULANT.

Kodune ettevalmistus:

1. Valige koostöös õppejõuga edaspidises töös kasutatav mittelineaarsus ning lähendage selle staatiline ülekandekarakteristik astmepolünoomiga vahemikus $U_{min}...U_{max}$. U_{min}, U_{max} valige lähtudes edaspidises töös kasutatavatest sisendsignaali nivoodest ja tööpunktist. Kasutage selleks keskkonnas SIMULANT töötavat programmi APROX.SIM.
2. Leidke optimaalne eelpinge sageduse N-kordistamiseks väikese sisendsignaali reziimis (N valige vahemikus 2...5). Leidke optimaalsele eelpingele vastava väljundsignaali spekter ning kordisti kasutegur.
3. Leidke optimaalne eelpinge sageduse muundamiseks väikese signaali reziimis.

Töö käik.

Käivitage programm SIMULANT ja viige see dialoogreziimi. Laadige mudel MLE. Ekraanil joonistub plokk-skeem (joon.5.8) järgmiste plokkidega:



Joon.5.8. Spektraalteisendused ML ahelas

Üldplokid:

TIM – ajaskaala formeerimine, teiste plokkide töö sünkroniseerimine

FFTplot0 – signaalide ja nende spektrite leidmine ning graafiline esitus

PARplot0 – parameetrilise signaaliparve leidmine

VARplot0 – parameetrilise sõltuvuse leidmine

DIGifilt – ribafilter spektrikomponendi filtreerimiseks

BUFfer0 – puhver mudeli modifitseerimise hõlbustamiseks

Sisendsignaali mudel:

GENcos0,GENcos1 – harmoonilised signaalid

CONst0 – alaliskomponent

GAUss0 – juhuslik normaaljaotusega signaal

SUM40 – sisendsignaalide summaator

Mittelineaarsuse mudel:

POLyline0 – tükati-lineaarne MLE

Mudeli lähteparameetrid on järgmised:

tim.points=256 – signaali punktide arv

tim.dt=1./256=... - diskreetimisperiood
const0.const=0 - alalispinge MLE sisendis
gencos0.freq=4 - 1. signaali sagedus
gencos0.ampl=1 - 1. signaali amplituud
gencos1.freq=12 - 2. signaali sagedus
gencos1.ampl=0.5 - 2. signaali amplituud
gauss.dev=0 - juhusliku signaali efektiivväärtus
digifilt0.dt=tim.dt - diskreetimisperiood
digifilt0.f1=3 - ribafiltri alumine lõikesagedus
digifilt0.f2=5 - ribafiltri ülemine lõikesagedus
parplot0.par=&const0.const - alalispinge muutmine
parplot0.parmin=0 - minimaalne alalispinge väärtus
parplot0.parmax=1 - maksimaalne alalispinge väärtus
parplot0.curves=5 - graafikuparve graafikute arv

1. Sageduse kordistamine.

Väärtustage mittelineaarse elemendi (MLE) lõikude arv ja käänupunktid vastavalt kodusele ettevalmistusele. Kontrollige selle staatilist pingeülekandekarakteristikut. Ühendage selleks alalispinge allikas MLE sisendisse ja plokk VARPLOT MLE väljundisse. Väärtustage karakteristiku leidmisel kasutatavad parameetrid ja käivitage simuleerimine (joon.5.9).

Ühendage esimene generaator ning alalispinge allikas summaatori sisendisse. Väärtustage alalispinge allikas vastavalt arvutuste tulemusena leitud optimaalsele eelpingele.

Käivitage simuleerimine ploki FFTPLOT0 suhtes. Leidke signaalide spektritelt kordistamise kasutegur ning võrrelge arvutuslikult leitud tulemusega (joon.5.10).

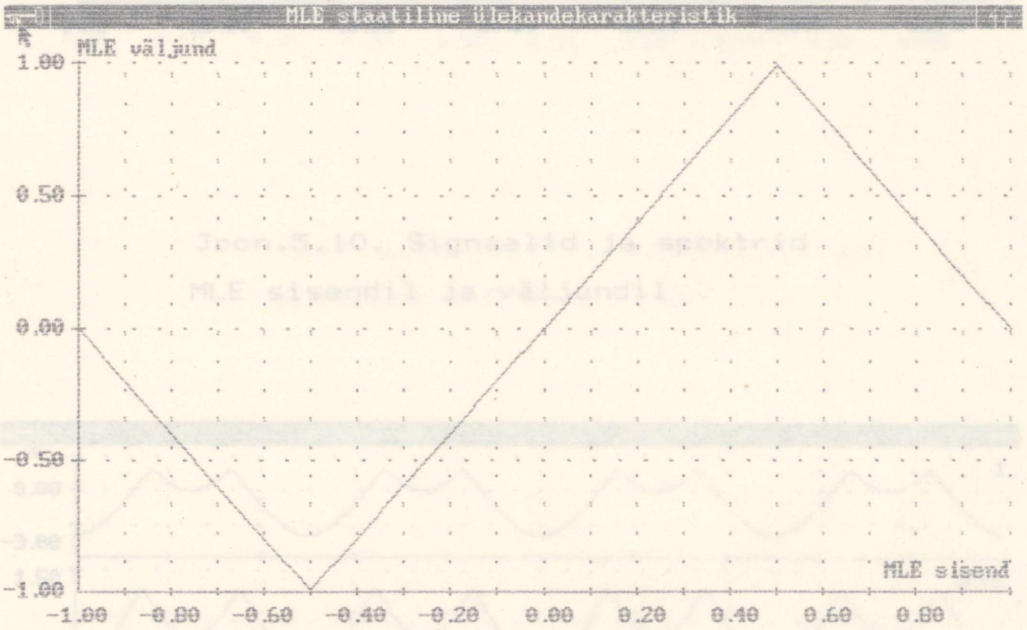
Jälgige ning võrrelge erinevatele sisendsignaali amplituudidele ning erinevatele eelpingetele vastavaid väljundsignaale ning nende spektreid.

Leidke soovitava spektrikomponendi sõltuvus MLE eelpingest. Seadistage selleks ribafiltri sagedusvahemik (plokk DIGIFILT0) vastavalt leitava spektrikomponendi sagedusele ning alalispinge muutmise piirid (plokk PARPLOT0) ning käivitage simuleerimine ploki PARPLOT0 suhtes (joon.5.11, 5.12). Võrrelge simuleerimisel leitud optimaalse eelpinge väärtust arvutustel saadud tulemusega.

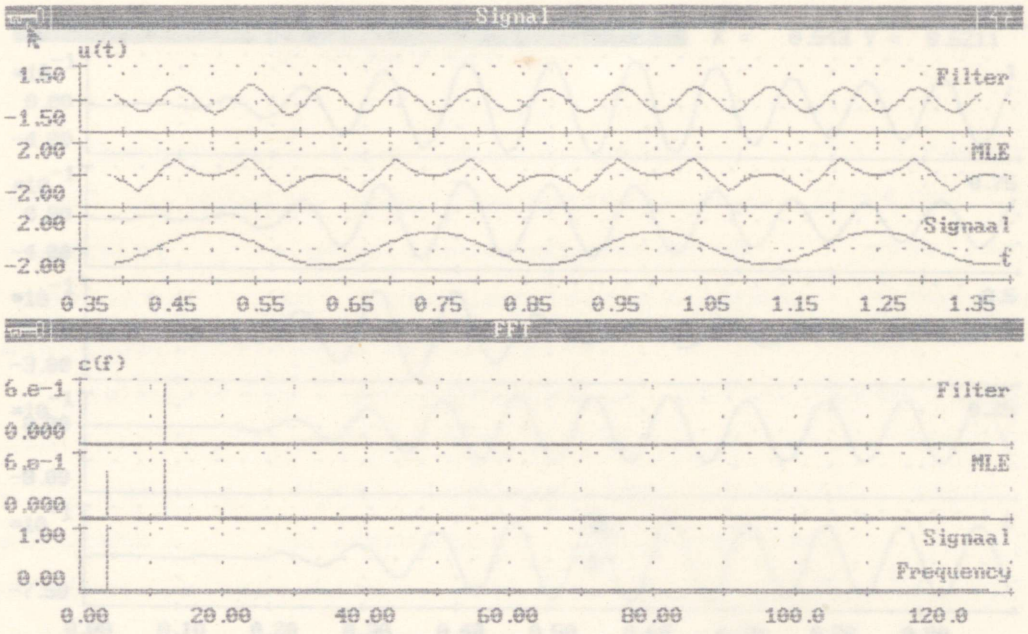
Asendage mittelineaarsus astmepolünoomiga (plokk POLYNOM) ning väärtustage selle kordajad vastavalt aproksimeerimise tulemustele. Korrake simuleerimist ning vôrrelge saadud tulemusi arvutuslike ning eelneva simuleerimise käigus saadud tulemustega.

2. Sageduse muundamine.

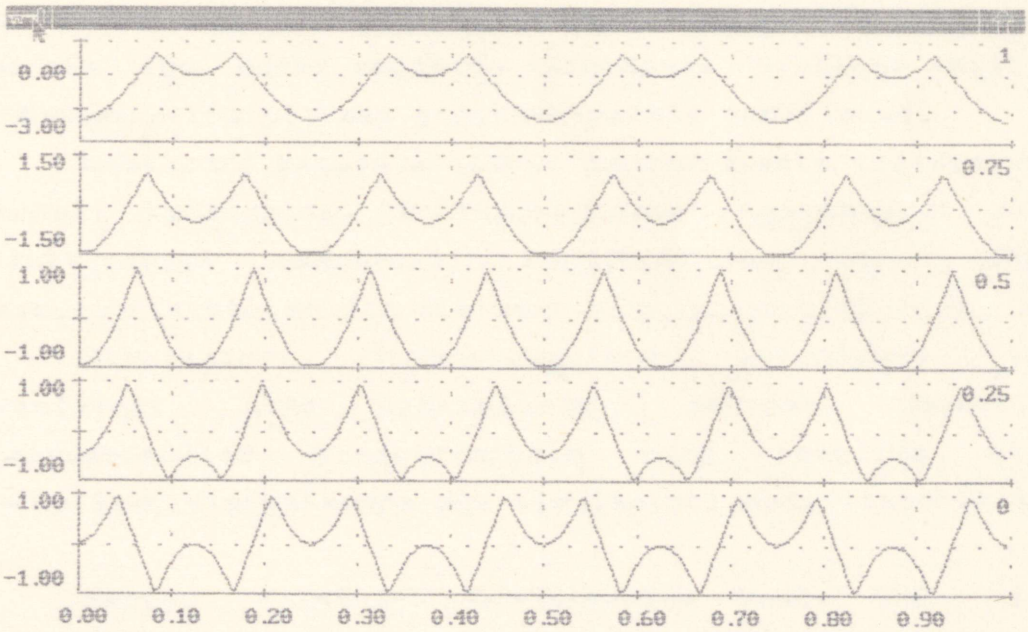
Lisage teine generaator summaatori sisendisse ja väärtustage selle parameetrid. Korrake p.1 kirjeldatud tegevusi vastavalt kodusele ülesandele. Uurige nôrga signaali mahasurumist tugeva signaali olemasolul.



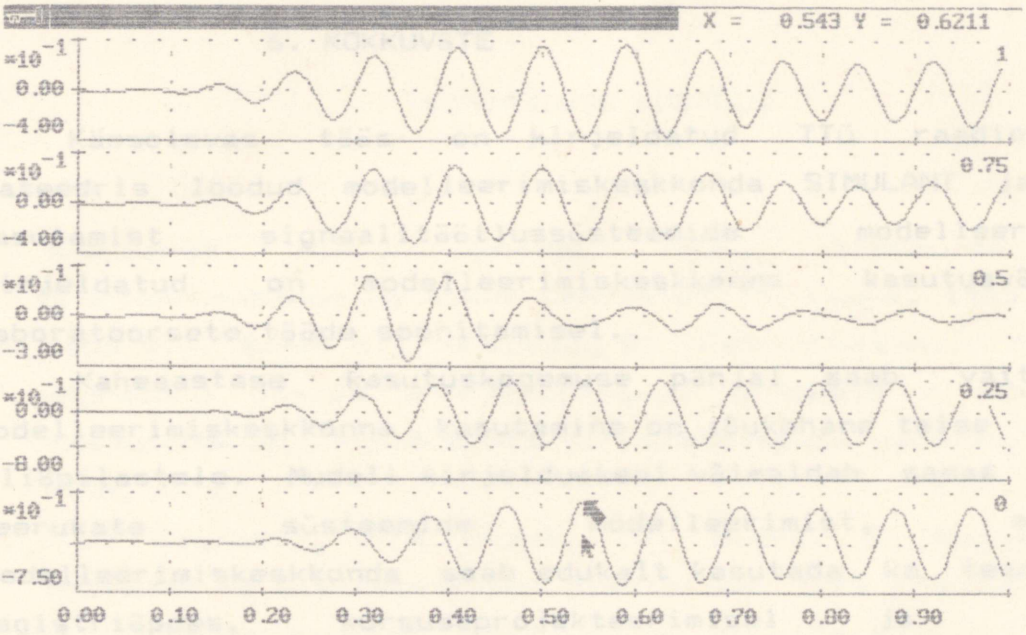
Joon.5.9. MLE staatiline pingeülekandekarakteristik



Joon.5.10. Signaalid ja spektrid
MLE sisendil ja väljundil



Joon.5.11. MLE väljundsignaali sõltuvus eelpingest



Joon.5.12. Filtri väljundsignaali
sõltuvus MLE eelpingest

6. KOKKUVÕTE

Käesolevas töös on kirjeldatud TTÜ raadiotehnika kateedris loodud modelleerimiskeskkonda SIMULANT ja selle kasutamist signaalitöötlussüsteemide modelleerimisel. Kirjeldatud on modelleerimiskeskkonna kasutusvõimalusi laboratoorsete tööde sooritamisel.

Kaheaastase kasutuskogemuse põhjal saab väita, et modelleerimiskeskkonna kasutamine on jõukohane teise kursuse üliõpilastele. Mudeli kirjelduskeel võimaldab samas küllalt keerukate süsteemide modelleerimist, mistõttu modelleerimiskeskkonda saab edukalt kasutada ka teadustöös, magistriõppes, kursuseprojekteerimisel ja diplomprojekteerimisel.

Simulaatori töökiiruse tõstmiseks vajaks translaatori genereeritav kood optimeerimist. Oluliselt tõstaks töökiirust kompilaatori kasutamine, kuid see teeks mudelite muutmise simuleerimise käigus mõnevõrra keerukamaks. Sel juhul teisendataks mudel enne simuleerimist mõnda üldlevinud protseduursesse keelde või otseselt masinkoodi. Esimesel juhul vajaks iga mudel seetõttu täiendavat transleerimist, teine lähenemisviis piiraks programmipaketi mobiilsust.

Keskkonna kasutajaliidese parandamiseks töötatakse välja mudeli koostamisel ja simuleerimisel kasutatavat graafilist plokk-skeemi redaktorit. Lõplikult on veel lahendamata simuleerimistulemuste arhiveerimine ja järeltöötlus.

Simulaatori funktsionaalsed võimalused täienevad vastavalt uute plokitüüpide, makrode, mudelite ja laademoodulite kirjeldamisele ning reeglina ei nõua modelleerimiskeskkonna ega simulaatori modifitseerimist.

Seoses riistvara kirjeldamise keele VHDL laialdase levikuga ja muutumisega standardiks ([14]) tuleks mõelda VHDL keele alamhulga realiseerimisele simulaatoris. See nõuaks paraku simulaatori töö olulist muutmist.

KIRJANDUSE LOETELU

1. Андрианов А.Н., Бычков С.П., Хорошилов А.И. Программирование на языке СИМУЛА-67. М.:Наука, 1985, 288 с.
2. Арро И., Юхмен В., Нийнсалу В. Радиосистемы. Индикатор кругового обзора и моделирование угломера с коническим сканированием: Руководство по лабораторным работам. Таллинн 1978, 44 с.
3. R.L. Bagrodia, K.M. Chandy, J. Misra. A Message-Based Approach to Discrete-Event Simulation. IEEE Transactions on Software Engineering, 1987, no. 6, pp. 654-665.
4. P. Corsini, L. Rizzo. SSCSSC: A Tool For the Teaching of Digital Circuits. IEEE Trans. on Education, Feb. 1991, pp.70-75.
5. A.El-Hajj, K.Y.Kabalan. A Spredsheet Simulation of Logic Networks. IEEE Trans. on Education, 1991, vol.34, No 1, pp. 43-46.
6. P. Guernic, A. Benveniste, P. Bournai. SIGNAL - A Data Flow-Oriented Language for Signal Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1986, no 2, pp. 362-374
7. V. Heinrichsen, G. Leibur. Spektraalanalüüs mittelineaarsetes ahelates. Metoodiline juhend raadiotehniliste ahelate ja signaalide laboratoorsete tööde sooritamiseks. Tallinn, TTÜ rotaprint, 1990.a., 28 lk.
8. Киндлер Е. Языки моделирования. Пер. с чеш. М.:Энергоатомиздат, 1985, 288 с.
9. G.A. Korn, J.V. Wait. Digital continuous-system simulation. Prentice-Hall 1978, Inc. Englewood Cliffs, 212 p.
10. Lumberg, T. Sobitatud filtri modelleerimine arvutil: Diplomitöö. Tallinn 1979, 161 lk.
11. J. McLead. Should we be more discreet? Simulation 1988 vol.51, No 4, pp.174-177
12. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II: Пер. с англ. М.:Мир, 1987, 646 с.
13. W. Schaufelberger. Design and Implementation of Software

for Control Education. IEEE Trans. on Education, 1990, vol.33, No 3, pp. 291-297.

14. K. Tammemäe. Riistvara kirjeldamise keel VHDL. Arvutustehnika ja Andmetöötlus, 1992, 4, lk.1-11.
15. The APROS software for process simulation and model development. Espoo: Technical Research Centre of Finland, 1989, 106 p.
16. Тыугу Э.Х. Концептуальное программирование. М.: Наука, 1984, 256 с.
17. E.H. Tyugu. Knowledge-based programming environments. Tallinn:Est.Acad. of Sci. Inst. of Cybernetics, 1990, 30 p.

L I S A D

A K T

signaalitöötluse süsteemide simulaatori programmi "SIMULANT" autorite kohta.

Käesolev akt on koostatud määramaks simulaatori (programmi "SIMULANT") põhiliste autorite osalust. Simulaator on valminud 2.5 aastase töö tulemusena põhiliselt Aarne Roosi ja Tõnu Lumbergi poolt. Üldiseks töö mahukuseks loeme 3.5..4 inimaastat. Lisaks on töö valmimisel kasutatud Villem Vannase poolt loodud menüüde süsteemi, inglisekeelsete tekstide ettevalmistamisel ja mõningate probleemide lahendamisel on abi osutanud Madis Kaal, süsteemi testimisel ja tekstide korrigeerimisel Ene Lääne. Venekeelsete tekstide tõlkimisel ja sisestmisel on kasutatud üliõpilaste abi. Simulandis kasutatav dünaamiline laadur pärineb süsteemist "SPADE".

Osaluse hindamine on teostatud eksperthinnangute alusel ja on toodud tabelites 1,2.

Tabel 1.

Töö osad	üldine kaal	Aarne Roosi	Tõnu Lumberg	Teised
Projekt	0.2	0.14	0.06	
Realisatsioon (vt.tabel 2.)	0.4	0.22	0.11	0.07
Testimine	0.2	0.12	0.05	0.03
Dokumenteerimine	0.2	0.04	0.13	0.03
Kokku	1.0	0.52	0.35	0.13

Tabel 2.

Realisatsiooni osad	Üldine kaal	Aarne Roosi	Tõnu Lumberg	Teised
Simuleeriv osa sellest:	0.70			
Menüüd	0.14	0.07		0.07
Translaator	0.14	0.14		
Mudeli täitmine	0.14	0.14		
Ülejäänud keskkond	0.28	0.20	0.05	0.03
Graafiline esitus sellest:	0.30			
Kasutajaliides	0.12		0.04	0.08
Jooniste esitus	0.18		0.18	
Kokku	1.00	0.55	0.27	0.18
Üldine osalus (x0.4, 2. rida tabelis 1)		0.22	0.11	0.07

Kooskõlastatud:

..... E.Lääne

..... M.Kaal

..... V.Vannas

..... A.Roosi

..... T.Lumberg

Käesolev akt on koostatud TTÜ Raadiotehika kateedri poolt esitamiseks magistrieksami sooritamise komisjonile. Akt on kinnitatud kateedri koosolekul 22.jaanuaril 1992.a. prot. Nr1.

Raadiotehnika kateedri juhataja I.Arro