Det Teknisk-Naturvidenskabelige Fakultet
Aalborg Universitet

Institut for Elektroniske Systemer

TITEL:
Guitar effekter

TEMA:
Signalbehandlingssystemer

PROJEKTPERIODE:
SB6
1. februar - 1. juni, 2001

PROJEKT GRUPPE:
641

DELTAGERE:
Jakob Iversen
Lasse Juul Villadsen
Hans Bjergegaard
Henrik Nørby Larsen
Christian Corfits Jensen
Brian Geppert

VEJLEDER:
Sofus Birkedal Nielsen

SYNOPSIS:
Projekts formål er at konstruere lydeffekter til en elektrisk guitar. Effekterne konstrueres som analoge og digitale effekter. I kravspecifikationen vælges de enkelte enheder; equalizer og forvrænger realiseres som analoge effekter, og rumklang, flanger og kompressor realiseres som digitale effekter. Til konstruktion af de digitale effekter benyttes en digital signal processor (DSP) af typen TMS320C26 fra Texas Instruments. Det samlede signalbehandlingssystem virker efter hensigten, og lyttemæssigt lyder effekterne som projektgruppen forventede.

Oplag: 9

Abstract

This report has been written by group 641 of the 6th semester at the Institute of Electronic Systems at Aalborg University. The aim of this project is to develop sound effects to an electric guitar. The effects are constructed as analog and digital effects. The project group has been using elements of the SPU-model in the design and construction of the following effects: Equalizer (analog), distortion (analog), compressor (digital), flanger (digital) and reverb (digital). The digital part of the project is implemented on a DSP (TMS320C26) from Texas Instruments. The final signal processing system is successful and sounds as the project group would have expected.

Forord

Projektet "Guitar effekter" er udarbejdet af gruppe 641 ved Aalborg Universitet på 6. semester Signalbehandling i perioden 1. februar til 1. juni år 2001. Projektet er skrevet til vejleder og censor med henblik på evaluering af projektforløbet. Yderligere vil andre studerende have mulighed for at hente ideer og inspiration fra projektet.
Gruppens medlemmer er:

Jakob Iversen	Lasse Juul Villadsen	Hans Bjergegaard
Henrik Nørby Larsen	Christian Corfits Jensen	Brian Geppert

Læsevejledning

For at gøre projektet så læsevenligt som muligt, har gruppen valgt at anvende følgende retningslinier:

- For at sikre et konsistent layout er rapporten inddelt i tre nummerede niveauer: Kapitel, afsnit og underafsnit. Dog vil indholdsfortegnelsen kun indeholde 2 niveauer.
- Kildehenvisninger udføres med en forkortelse, der henviser til litteraturlisten samt en henvisning til sidetallet i den pågældende kilde, f.eks. [1, s. 72]. Hvis intet sidetal er angivet, ser henvisningen således ud: [1]. Litteraturlisten findes side på 101.
- Formelhenvisninger er højrestillet for formlen. De er nummereret med to numre, hvor det første angiver kapitel- eller appendiksnummer.
- Hextal indikeres med et sænket h efter tallet, f.eks. $A1B_h$, og binære tal indikeres med sænket b, f.eks. 0110_b.
- Aktiv lav angives med overstregning f.eks.: $\overline{CS}$
- Appendiks er nummereret med bogstaver.
- Bilag er nummeret med romertal, og findes efter appendiks.
- Vedlagt projektet er en CD-ROM. På denne findes simuleringer og lydfiler, kildekode, datablade mv. Se bilag VI på side 183 for yderligere information.
- Diagrammer for de enkelte moduler, er placeret på folde-ud-sider i A3 format bagerst i rapporten.

Indhold

1 Indledning **1**

1.1 Den akustiske guitar 1

1.2 Den elektriske guitar 1

1.3 Effekter til el-guitar 2

1.4 Projektet 8

2 Kravspecifikation **9**

2.1 Generel beskrivelse 9

2.2 Specifikke krav 10

2.3 Grænseflader 12

2.4 Specifikation af accepttest 13

2.5 Metodeafsnit 14

3 Equalizer **15**

3.1 Krav 15

3.2 Løsningsmuligheder 16

3.3 Design af equalizer 16

4 Forvrænger **27**

4.1 Krav til forvrænger 27

4.2 Løsningsmuligheder til forvrængeren 27
4.3 Design af forvrænger 32
4.4 Delkonklusion 33

5 Analog/Digital grænseflade 34

5.1 Indgangstrin 34
5.2 Valg af konverter 37
5.3 Design af codec 38
5.4 Yderligere tiltag 40
5.5 Delkonklusion 40

6 Hardware-opkobling 41

6.1 Den benyttede DSP 41
6.2 Den benyttede RAM kreds 42
6.3 Den benyttede debugger 44
6.4 Brugerinterface 45
6.5 Kobling af systemet 46

7 Software 50

7.1 Udviklingssoftware til DSP 50
7.2 Filer til kompilering 50
7.3 Den samlede kompilering 52
7.4 Programopbygning 52
7.5 Opdeling af adresserum 55

8 Kompressor 57

8.1 Krav til kompressoren 57
8.2 Princippet i kompressoren 57
8.3 Matlab simulering 58
8.4 Design af kompressor 59
8.5 Implementation af kompressor 68
8.6 Delkonklusion 69

9 Flanger **71**

9.1 Krav til flanger 71

9.2 Generelt om flangeren 71

9.3 Metoder til realisering af flanger 74

9.4 Valg af løsningsmetode 77

9.5 Design af flanger 78

9.6 Delkonklusion 79

10 Rumklang **81**

10.1 Krav til rumklang 81

10.2 Beskrivelse af rumklang 81

10.3 Løsningsmuligheder 82

10.4 Design af Schroeders reverberator 83

10.5 Delkonklusion 90

11 Accepttest **91**

11.1 Udførelse af accepttesten 91

11.2 Konklusion på accepttesten 94

12 Konklusion **95**

12.1 Det konstruerede system 95

12.2 Forbedringer af systemet 95

12.3 Overholdelse af projektenhedbeskrivelsen 97

12.4 Opsummering 98

Litteraturliste **100**

Appendiks **102**

A Målerapport for guitar **102**

A.1 Øvre og Nedre grænsefrekvens samt SNR 102

A.2 Måling af udgangsspænding 103

A.3 Måling af udgangsimpedans 105

B Måling af diodekarakteristikker **107**

B.1 Diode 107

C Testrapport **111**

C.1 Simulering/test af modul 1: Equalizer 111

C.2 Simuleringer/målinger modul 2: Forvrængeren 114

C.3 Simuleringer/målinger på modul 3: Codec/indgangstrin 119

C.4 Simulering/test af modul 4: Kompressor 123

C.5 Simulering/test af modul 5: Flanger 128

C.6 Simulering/test af modul 6: Rumklang 130

D Beregning af støj **132**

D.1 Støj i indgangstrinnet 132

D.2 Trunkeringsstøj 135

E Assemblerkode **143**

Bilag **169**

I Europarack **169**

I.1 Busforbindelser 170

I.2 Komponentplaceringer på Europakort 171

II Diagram Equalizer **175**

III Diagram Forvrænger **177**

IV Diagram Codec **179**

V Diagram DSP **181**

VI Den vedlagte CD-ROM **183**

KAPITEL 1

Indledning

Dette kapitel er en introduktion til den elektriske guitar samt guitareffekter. Afsnittet danner baggrund for valg af emne.

Afsnittet omhandlende guitarens historie bygger på følgende litteratur: [3] og [12].

1.1 Den akustiske guitar

Ordet "guitar" stammer fra det oldiranske "sitar", som betyder "3 tarm". Hvornår guitaren blev opfundet vides ikke præcist, men strengeinstrumenter med en krop af træ, har eksisteret i mere end 5000 år. Disse har været forgængeren for lutten og guitaren. I starten var disse instrumenter uden bånd og med et varierende antal strenge, men i løbet af 1500 års udvikling (omkring 1500 f.Kr) fremkom mandolinen og lutten. Den spanske guitar blev opfundet i perioden, hvor araberne bosatte sig i Spanien. Den blev dog ikke populær før år 1600; måske fordi andre strengeinstrumenter, såsom lutten, havde en meget kraftigere klang. I tidernes løb har guitaren haft 4-12 strenge, og har været stemt på forskellige måder. Først i 1850 fremkom den version af den klassiske (spanske) guitar, der kendes i dag. Denne guitar består af en hul trækasse, en hals med gribebræt og 6 strenge. Lyden fremkommer, når strengene sættes i svingninger og forstærkes via trækassen. Siden den er modeller som western-guitaren, jazz-guitaren, blues-guitaren osv. kommet til.

1.2 Den elektriske guitar

I 1930'erne blev de første guitarmikrofoner lavet, hvorved den elektriske guitar opstod. Det var dog en almindelig spansk guitar med mikrofon. Først i 1948 kunne Fender præsentere

den første af de elektriske guitarer, der kendes i dag (Fender Telecaster). Denne guitar var uden klanghuller, havde ingen klangbund, men var fremstillet af kompakt træ. Forstærkningen af signalet foregik via pickups placeret under hver streng. Signalet fra guitaren blev så forstærkeret via en guitarforstærker. Den akustiske guitar har sin klangkarakteristik fra trækassen, men det har den elektriske guitar ikke. Derfor laves forstærkere til elektriske guitarer ofte med forskellige forvrængningskarakteristikker, for at opnå en bestemt klang. En videreudvikling af klangændringen er guitareffekter. Disse ændrer signalet fra guitaren på forskellig vis. I det følgende beskrives forskellige guitareffekter.

1.3 Effekter til el-guitar

Til beskrivelse af effekter til guitareffekter er følgende kilder brugt: [[8], s. 95-108] samt [14].

For at opnå lydeffekter bruger guitarister ofte en pedal, som kan trykkes frem eller trækkes tilbage. Når pedalen aktiveres af guitaristen, pålægges lyden en effekt, fordi signalet fra forforstærkeren moduleres. Der er i hovedtræk tre forskellige måder at modulere signalet på:

- Amplitudemodulerede effekter.
- Frekvensmodulerede effekter.
- Tidsmodulerede effekter.

I de følgende underafsnit vil nogle hyppigt anvendte effekter fra hver af disse kategorier blive beskrevet.

1.3.1 Amplitudemodulerede effekter

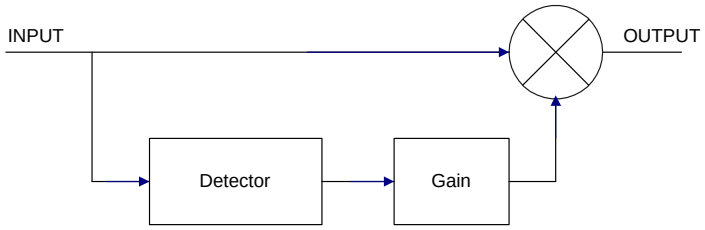
Under denne kategori findes mange forskellige effekter, som f.eks. tremolo, auto tremolo, panning, expansion/kompression og noise gate. Endvidere findes også de såkaldte "wave form" modulerede effekter, såsom asymmetrisk klipning, symmetrisk klipning, full-wave rectification og half-wave rectification. Af disse beskrives expansion/kompression, full-wave rectification og symmetrisk klipning.

Kompression/expansion

Målet med denne effekt er at komprimere/expandere dynamikken i signalet. Kompressoren enten hæver signalet, (lydtrykniveauet) ved svage sekvenser i signalet eller dæmper amplituden af kraftige signaler. På den måde komprimeres signalet, og dynamikområdet mindskes. Denne effekt er hyppigt anvendt i mikrofoner til sangere, hvor effekten gør, at afstanden mellem sangeren og mikrofonen kan variere og lydniveauet forbliver ensartet.

Denne effekt anvendes ligeledes til f.eks. FM-radio, hvilket gør, at signalet bliver bedre at lytte til i støjfyldte omgivelser.

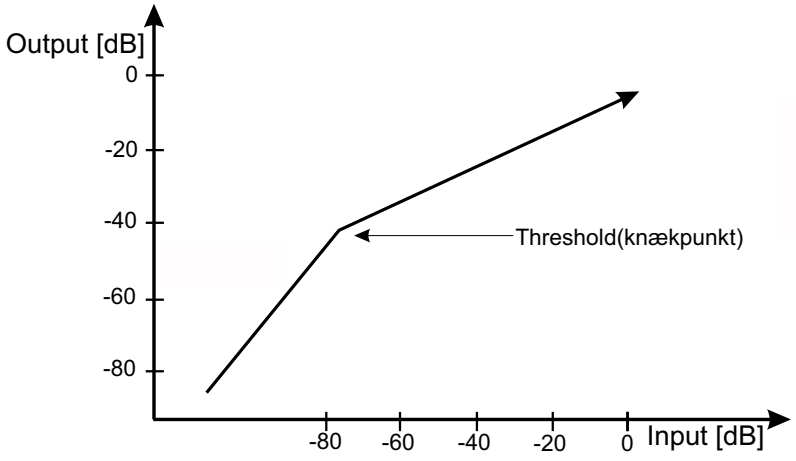
Expanderen virker lige modsat af kompressoren. Resultatet er altså en forøgelse af signalets dynamik. Dette finder ofte anvendelse ved lave niveauer som en form for støjbegrænsning. Figur 1.1 illustrerer hvorledes effekten moduleres.



Figur 1.1: *Princippet i expanderen/kompressoren.*

Regulering Det der afgør, hvordan en kompressor/expander lyder, er indreguleringstiden (attack) og tilbagereguleringstiden (release). Disse tider bestemmer, hvor hurtigt kompressoren/expanderen reagerer overfor det indkommende signal. Det menneskelige øre er ca. 4ms om at konstatere frekvensen af en tone, og derfor skal indreguleringstiden gerne være mindre end 4ms. Tilsvarende har hørelsen også en form for "tilbageløbstid" efter pludselige niveauændringer. Den er på 0,2-0,5s, men for klangfulde, harmoniske signaler er denne tid 1-2s [[8], s. 60-80] .

Knækpunkter Det niveau, hvor kompressionen starter kaldes knækpunktet/threshold (se figur 1.2).

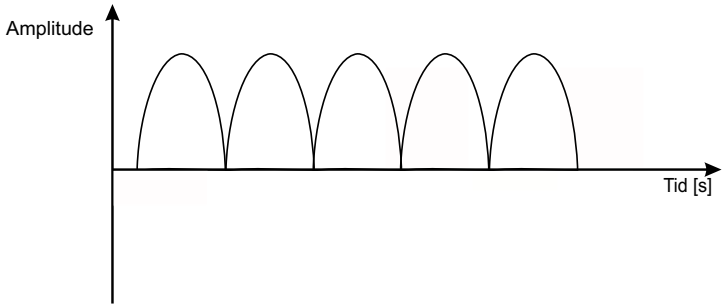


Figur 1.2: *Input/output karakteristik for en kompressor.*

De lydniveauer som ligger over dette punkt vil blive komprimeret. Ofte vælges dette niveau lavt, således at nyttesignalet komprimeres, uden at bagrundsstøjen kommer højere op end nødvendigt. Derved ændres outputniveauet samtidig med kompressionsforholdet.

Full-wave rectification

I full-wave rectification ensrettes signalet, hvilket giver et output, som minder lidt om DC (se figur 1.3). Ingen af signalets originale frekvenser er tilbage, kun anden- og højereordens harmoniske.

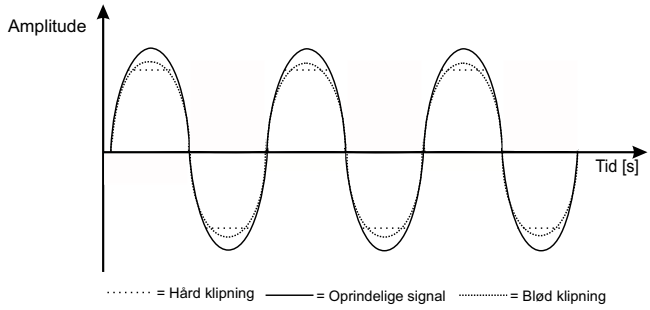


Figur 1.3: *Sinusbølge efter full-wave rectification modulation.*

Dette giver meget kraftig forstærkning af oktaverne af inputtet.

Symmetrisk klipning

Ved denne form for signalmodulering afskæres toppen og bunden af signalet. Denne afskæring foretages symmetrisk (se figur 1.4), hvilket medfører, at kun de ulige ordener af de harmoniske overtoner gengives.



Figur 1.4: *Princippet bag symmetrisk klipning illustreret på en sinusbølge.*

Dette resulterer i, at musiksignalet kommer til at lyde meget urent. Signalet kan klippes forskelligt, og alt efter hvor meget der klippes af signalets amplitude, desto flere højereordens harmoniske bølger gengives. Populært sagt: "Jo hårdere der klippes, jo mere støjfyldt bliver signalet". Hård klipning kaldes distortion, og blød klipning kaldes overdrive.

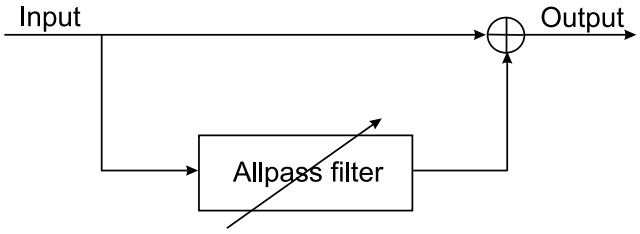
1.3.2 Frekvensmodulerede effekter

Under frekvensmodulerede effekter kan nævnes phasor, wah-wah, kabinetsimulering, resonator, treble boost, mid boost og equalizer. Af disse effekter beskrives phasor, gra-

fisk/parametrisk equalizer, wah-wah og resonator.

Phasor

En phasor (phase-shift) er en effekt, der tidsforsinker signalet ved hjælp af faseforskydelse. Dette signal kombineres med det rene signal fra guitaren, således at det modulerede signal og det originale signal adderes (se figur 1.5).

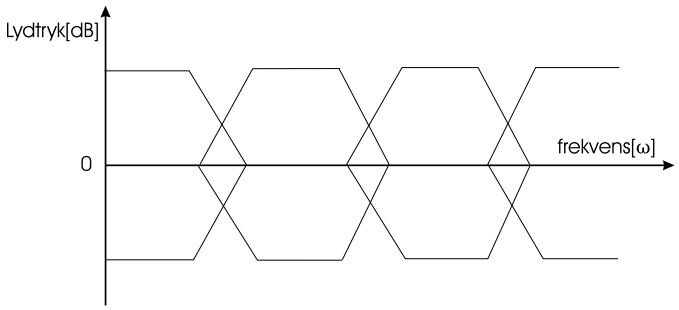


Figur 1.5: *Blokdiagram som illustrerer princippet for phasoren.*

Når guitaristen aktiverer effekten, lyder det som om to guitarister spiller det samme, lidt forsinket i forhold til hinanden. Det betyder at visse toner forstærkes og andre dæmpes.

Grafisk equalizer

En grafisk equalizer bruges til at indstille forstærkningen af specifikke frekvensområder (se figur 1.6). Dette foregår ved at man inddeler frekvensspektret i forskellige bånd.



Figur 1.6: *Båndopdelingen for en grafisk equalizer.*

Dette gøres vha. filtre, hvor forstærkningen er regulerbar. Alt efter om der ønskes at et frekvensområde skal hæves eller dæmpes i lydtryk, reguleres forstærkningen i filtret. Denne effekt findes på mange hifi- og guitarforstærkere.

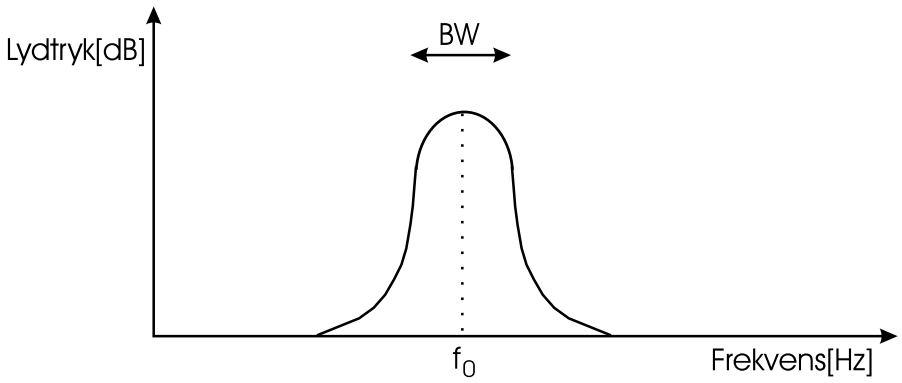
Parametrisk equalizer

Den parametriske equalizer er i realiteten blot en udvidet udgave af den grafiske. Udover at hæve forstærkningen indenfor de pågældende frekvensbånd, muliggør den ændring af

båndbredden for de enkelte filtre.

Wah-wah

Ideen bag wah-wah effekten er at forskyde en peak op og ned i frekvensspektret. På denne måde opnås en forstærkning af frekvenserne omkring peaken, når effekten aktiveres (se figur 1.7). Dette foregår ved, at man flytter et båndpasfilters centerfrekvens (f_0) op og ned i frekvensspektret.



Figur 1.7: *Amplitudekarakteristikken for wah-wah effekten.*

Frekvensspektret, hvor wah-wah pedalens centerfrekvens varieres indenfor, er ofte 400-2000Hz, da der indenfor dette område findes størst dynamik. Udover at regulere f_0, er der, i nogle varianter af wah-wah, mulighed for at regulere godheden (Q) af filtret, hvilket udgør bredden af peaken (båndbredden BW).

Resonator

En resonator bygger på samme princip som wah-wah effekten. Forskellen består i, at centerfrekvensen for båndpasfiltret ikke kan ændres. Effekten giver et løft i det pågældende frekvensområde, når den aktiveres.

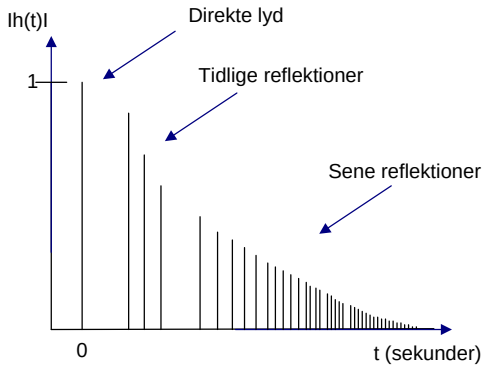
1.3.3 Tidsmodulerede effekter

Der findes mange forskellige måder, hvorpå et signal kan moduleres vha. tidsforsinkelse. Som eksempel kan nævnes ekko, rumklang, delay, flanger samt chorus. I det følgende beskrives rumklang, flanger og chorus.

Rumklang

Idéen bag denne effekt er at konstruere en effekt, der kan efterligne et rums impulsrespons. Der er stor forskel på, hvordan lyden udbredes i det fri og i et lukket rum. I det fri lytter man tilnærmelsesvist til den direkte lyd, som kommer fra lydgiveren, mens man i et lukket

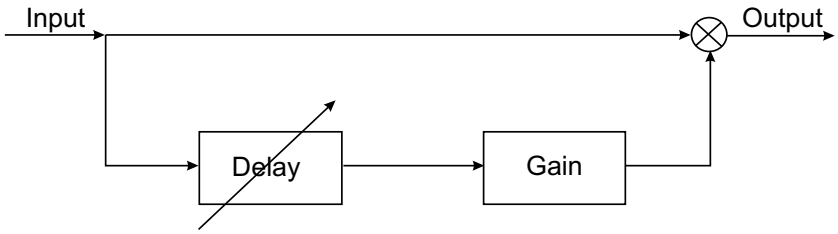
rum oplever både det direkte lydfelt og det diffuse lydfelt (refleksioner fra vægge, loft, gulv og genstande).



Figur 1.8: *Lydudbredelse i et rum.*

Flanger

Flanger er en effekt som er baseret på en variabel forsinkelse af signalet. Forsinkelsen varieres med en frekvens på op til 1Hz. Den maksimale forsinkelse er ca. 10ms. På figur 1.9 ses flangereffektens blokdiagram.



Figur 1.9: *Blokdiagram til illustration af princippet bag flangereffekten.*

Det forsinkede signal kombineres med det oprindelige signal og det resulterer i at nogle toner (frekvenser) faseforskydes i forhold til de oprindelige. For lytteren resulterer det i, at nogle toner fremhæves og andre dæmpes i forhold til det øvrige signal.

Chorus

Chorus-effekten minder meget om flanger. Den eneste forskel er blot at der er to eller flere fremadkoblede forsinkelser istedet for én.

1.4 Projektet

Der er fra Aalborg Universitet stillet en elektrisk guitar og en guitarforstærker til rådighed. Med udgangspunkt i det udleverede materiale, ønskes der i dette projekt, at designe og konstruere effektenheder, der ændrer lyden fra guitar og forstærker. Dette gøres vha. analog- og digital signalbehandling. Der tages udgangspunkt i de førbeskrevne effekter.

1.4.1 Problemformulering

Der vil i dette projekt blive designet og konstrueret hardware og software, der har til formål at påføre effekter til lyden fra en guitar.

Dette giver anledning til følgende problemformulering:

Hvorledes konstrueres effektenheder til en guitar?

1.4.2 Projektafgrænsning

For at begrænse projektet er det valgt at designe og konstruere 5 forskellige effekter. Dette gøres primært af tidsmæssige årsager. Det vælges at konstruere effekter, der stammer fra de førnævnte 3 forskellige effektformer. Dette er effekter, der er amplitude-, frekvens- og tidsmodulerede. Disse er valgt til at være:

- Equalizer (frekvensmodulation).
- Forvrænger/klipning (amplitudemodulation).
- Flanger (tidsmodulation).
- Kompressor (amplitudemodulation).
- Rumklang (tidsmodulation)

Der ses således bort fra andre effekter end de valgte.

KAPITEL 2

Kravspecifikation

Kravspecifikationen tager udgangspunkt i relevante delelementer fra SPU modellen [5]. Kravspecifikationens formål er, at klarlægge kravene til det endelige system, og det er udfra disse krav, at systemet skal designes og konstrueres.

2.1 Generel beskrivelse

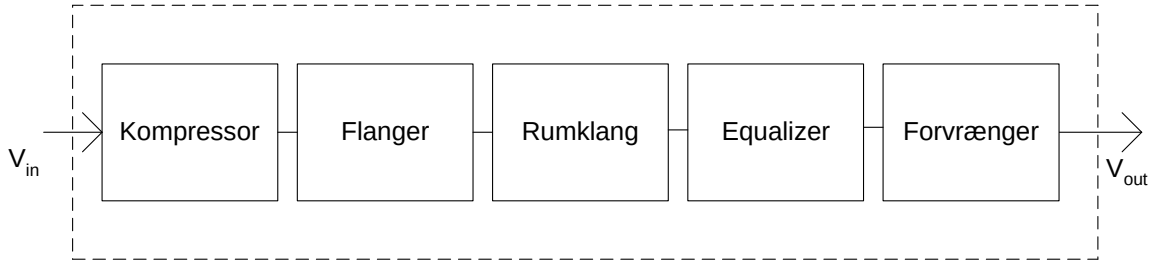
2.1.1 Systembeskrivelse

Systemet, der skal designes og konstrueres, består af lydeffektenheder, der kan bruges til en el-guitar. Systemet vil bestå af 5 forskellige effekter, der seriekobles. Der vil blive konstrueret analoge og digitale effekter.

I design og konstruktion af lydeffekterne er følgende stillet til rådighed af AAU:

- Elektrisk guitar af mærket: Axtech Vintage Plus.
- Forstærker af mærket: **ROLAND DAC-15XD**.
- DSP TMS320C26 fra Texas Instruments.
- Debugger: D/M3202XV udviklet på AAU.
- Diverse aktive og passive analoge komponenter.
- IC'ere.

Blokdiagrammet for systemet ses på figur 2.1, hvor effektenhederne er omkranset af en stiplet linie. Det skal således være muligt at spille på guitaren, hvorefter signalet behandles af en eller flere effektenheder, og sendes videre til forstærkeren. Selve signalbehandlingen foretages efter de valg brugeren har truffet. Dog bliver brugerinterfacet holdt til et minimum, da det ikke er det essentielle i dette projekt.



Figur 2.1: *Et blokdiagram af det samlede system.*

2.1.2 Begrænsninger

Begrænsningerne i dette projekt består primært af antallet af effektenheder og udformningen af enhederne. Det er af tidsmæssige årsager ikke muligt at konstruere mange effektenheder, og derfor er det valgt at fokusere på enheder, der dels passer til studieordningen for 6. semester, og samtidig omfatter alle 3 former for lydmodulation. Angående den fysiske udformning og design af enhederne bliver dette holdt på et minimum af ressource-mæssige årsager. Projektgruppen finder det vigtigere at bruge ressourcer på at lave funktionelle enheder end et pænt design.

2.2 Specifikke krav

I dette afsnit fastsættes de specifikke krav, der skal opfyldes, ved design af guitar effektenhederne. Først opstilles de overordnede krav til det samlede system, hvorefter kravene til de enkelte effektenheder vil blive opstillet. Disse krav er bl.a. opstillet på baggrund af målinger på det udleverede hardware.

2.2.1 Krav til det samlede system

Her opstilles kravene til det samlede system:

- De enkelte effektenheder skal modulopbygges.
- Afvikling skal foregå i realtid: Maksimalt delay på 1 sampling.
- Ændring på brugerinterfacet må maksimalt tage 1 sekund at opdatere.
- Der arbejdes med monogengivelse.

- Det samlede frekvensområde er fra 60Hz-11kHz (se evt. appendiks A på side 102, Målinger på guitar).
- Et samlet signal/støj forhold (SNR) på mindst 60dB (se evt. appendiks A på side 102, Målinger på guitar).
- THD (total harmonisk forvrængning) på maksimalt 0,7%. dette gælder for trin der ikke har til formål at forvrænge, taget fra DIN 45500 normen for hifi-forstærkere.
- Enheder skal kunne fungere uafhængigt af hinanden.
- Der må ikke kunne forekomme ustabilitet i de enkelte effektenheder.
- Indgangsimpedans i de enkelte analoge effektenheder skal være mindst $10k\Omega$.
- Udgangsimpedans i de enkelte analoge effektenheder skal være maksimalt $1k\Omega$.

2.2.2 Krav til kompressoren

- Valg af forskellige foruddefinerede kompressionsrater.
- Valg af threshold-værdier mellem 0 og -30dB.

2.2.3 Krav til flangeren

- Variabel delay frekvens op til 1Hz.
- Delay varieres mellem 1-10ms.

2.2.4 Krav til rumklang

- Efterklangstid på 0,8-3,0s.

2.2.5 Krav til Equalizeren

- Maksimal flad frekvenskarakteristik i pasbåndene.
- Maksimal dæmpning i pasbåndet på 3dB.
- Hæve eller sænke bassen med +/-12dB.
- Hæve eller sænke mellemtonen med +/-12dB.
- Hæve eller sænke diskanten med +/-12dB.
- Bassens frekvensområde strækker sig til 200Hz.
- Mellemtonens pasbåndsområde strækker sig fra 400 til 1250Hz.

- Diskantens frekvensområde strækker sig fra 2,5kHz.
- Overlap i andre bånd må maksimalt være 25%.

2.2.6 Krav til forvrængeren

- Forskellige forvrængningsniveauer.
- Forvrængning ved både små og store signaler.
- Valg mellem symmetrisk og asymmetrisk forvrængning/klipning.

Hvorvidt de opstillede krav er overholdt for de enkelte moduler, bestemmes i de modultests, som forefindes i appendiks C på side 111.

2.3 Grænseflader

I dette afsnit vil grænsefladerne til systemet blive defineret.

2.3.1 Brugergrænsefladen

Som før nævnt vil brugergrænsefladen i dette projekt blive nedprioriteret. Den vil bestå af justerbare analoge og digitale komponenter. Det vil således ikke være muligt for en guitarist direkte at bruge de konstruede effektenheder, da selve fodpedalerne ikke designes. Istedet justeres effekterne vha. potentiometre og kontakter.

2.3.2 Hardware grænseflader

Da effektenhederne skal virke uafhængigt af hinanden, kan de også konstrueres uafhængigt af hinanden. For at gøre dette er grænsefladerne mellem disse nødt til at blive fastlagt.

Eksterne Grænseflader

Grænsefladen fra systemet til omverdenen består på inputsiden af udgangen på en elguitar (jack-stik) og indgangen på en guitarforstærker på outputsiden (jack-stik).

Analog grænseflade

Som nævnt i de generelle krav ønskes en indgangsimpedans på mindst 10kΩ og en udgangsimpedans på maksimalt 1kΩ. Dette gøres for ikke at opnå uønskede effekter, såsom forvrængning og signaltab.

Digital grænseflade

Denne grænseflade findes internt på det benyttede digitale hardware. Det er vigtig, at der ikke forekommer overlap i de benyttede digitale enheder. Derfor skal den digitale grænseflade fastlægges. Dette gøres i kapitel 7 på side 50.

PC/DSP grænseflade

Der bruges RS-232 seriel kommunikation mellem PC og DSP-systemet. Kommunikationen på DSP'en bliver varetaget af en debugger D/M3202x-V. PC-siden bruger et standard terminal program, der understøtter VT100 standarden. Som compiler bruges Texas Instruments assembler-compiler samt diverse text-editorer og linker værktøj.

Analog/Digital grænseflade

Grænsefladen mellem de analoge og digitale enheder er AD/DA konvertere med dertilhørende analog signaltilpasning. Det vælges at indsætte de digitale effektenheder som de første enheder, hvorefter disse bliver efterfulgt af de analoge enheder. Grunden til dette er, at de analoge enheder, kan tilføre signalet frekvenser over 11kHz. Ved at placere de analoge enheder efter de digitale, bliver disse frekvenser ikke filtreret væk i anti-aliaseringsfiltret.

2.4 Specifikation af accepttest

Når det endelige system er opstillet, og systemets funktioner er bestemt, kan en specifikation af accepttest udarbejdes. Denne specifikation indholder krav til de tests, som skal udføres på systemet, for at demonstrere at systemet virker efter hensigten.

Specifikation af krav til systemet:

Krav 1: Det skal være muligt, at vælge en vilkårlig effekt, hvorefter dennes parametre skal kunne ændres inden for 1s.

Krav 2: Det skal være muligt, at vælge alle effekter samtidigt uden tab af information.

Krav 3: Signal/støj forhold over 60dB

Krav 4: THD under 0,7%

Accepttesten af systemet deles op i 4 tests, som skal vise, at de ovenstående krav er opfyldt. Disse tests er:

Test 1: Programmet loades over i RAM på DSP-kortet. Systemet startes, og effekterne vælges én efter én. Det konkluderes, om systemet kan skifte mellem effektenhederne, ændre parametre og om de virker efter hensigten.

Test 2: Programmet loades over i RAM på DSP-kortet. Systemet startes, og alle effekterne vælges. Der konkluderes, om systemet kan køre alle effekterne samtidigt.

Test 3: Der konstrueres ekstra software til programmet, der indikerer hvornår et interrupt er i gang. Dette loades over i RAM på DSP-kortet og systemet startes. Alle effekter startes, og det konkluderes om systemet kan nå at færdiggøre interrupt efter hver sample. Ved dette konkluderes om der er tab af data.

Test 4: Måling af SNR og THD.

2.5 Metodeafsnit

I kravspecifikationen er kravene til det samlede system fastlagt. Det er som før nævnt muligt at konstruere effektenhederne uafhængigt af hinanden. Det er valgt at designe og konstruere de analoge effektenheder først, og derefter de digitale. Grunden til dette er, at det stemmer bedst overens med tilrettelæggelsen af de tilhørende PE-kurser. Før dokumentationen af de digitale effektenheder vil der være et kapitel, der beskriver den analoge/digitale grænseflade, et kapitel, der beskriver selve hardwareopkoblingen og et kapitel der omhandler software.

Der vil, efter hver konstruktion, blive foretaget en test af den pågældende enhed. Når alle enheder er konstrueret og testet, samles disse og accepttesten udføres. Derefter kan det konkluderes om projektet er vellykket. Derved bliver opbygningen af resten af rapporten:

- Equalizer.
- Forvrænger.
- AD/DA konvertering.
- Hardwareopkobling.
- Software.
- Kompressor.
- Flanger.
- Reverb.
- Accepttest.
- Konklusion.

KAPITEL 3

Equalizer

Dette kapitel omhandler design og konstruktion af en equalizer. Equalizeren skal kunne hæve eller sænke et ønsket frekvensområde. Det er valgt, at konstruere equalizeren med tre bånd, der repræsenterer bassen, mellemtonen og diskanten.
Kapitlet er baseret på [[4], s. 33-40,242-66]

3.1 Krav

De opstillede krav er følgende:

- Maksimal flad frekvenskarakteristik i pasbåndene.
- Maksimal dæmpning i pasbåndet på 3dB.
- Hæve eller sænke bassen med +/-12dB.
- Hæve eller sænke mellemtonen med +/-12dB.
- Hæve eller sænke diskanten med +/-12dB.
- Bassens frekvensområde strækker sig til 200Hz.
- Mellemtonens pasbåndsområde strækker sig fra 400 til 1250Hz.
- Diskantens frekvensområde strækker sig fra 2,5kHz.
- Overlap i andre bånd må maksimalt være 25%.
- Øvrige generelle krav fra kravspecifikationen.

3.2 Løsningsmuligheder

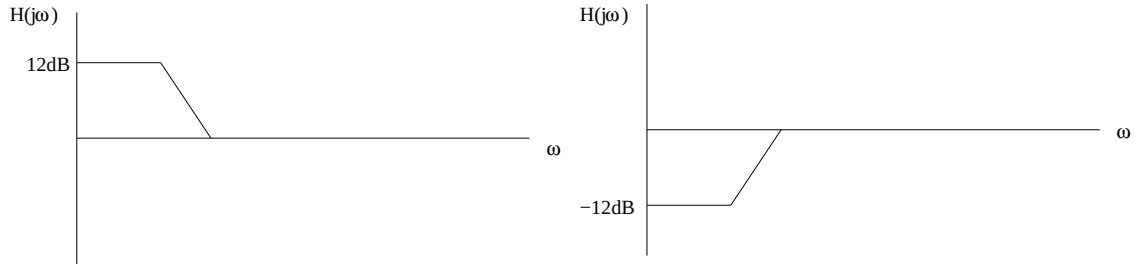
Der er flere måder at realisere equalizeren på. Der kan vælges den parametriske eller den grafiske. Disse kan så konstrueres både analogt og digitalt. Det vælges at konstruere equalizeren analogt, da den analoge equalizer i stor udstrækning anvendes i hifi-forstærkere og guitarforstærkere (f.eks. Marshall Valve). Fordelen ved at konstruere equalizeren analogt er, at forstærkningen i de givne pasbånd vil kunne varieres direkte med et potentiometer (dvs. kontinuert), hvorimod den digitale equalizer foretager ændringen i forstærkningen i step. Ulempen ved den analoge equalizer er, at der bruges komponenter, der kunne spares ved en digital løsning, idet en DSP allerede benyttes.

Ydermere er det valgt at konstruere filtrene grafisk, dvs. med fastlagte knækfrekvenser (se evt. Indledningen).

3.3 Design af equalizer

Det vælges at konstruere equalizeren som 3 kaskadekoblede delsystemer, der hver varetager et frekvensområde. Inden for hver af disse områder skal forstærkningen kunne varieres mellem +/-12dB.

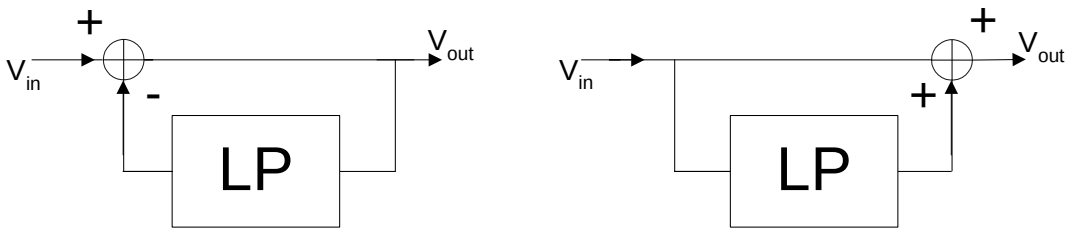
For at konstruere elektronik, der varetager delsystemernes funktioner, ses der på equalizeren i dens to yderpositioner, dvs. ved +12dB og ved -12dB. Der tages udgangspunkt i baskontrollen. Ved +12dB skal det oprindelige signal forstærkes 1 gang samtidigt med at bassens pasbåndsområde skal lægges til det oprindelige signal. Ved -12dB skal bassens pasbåndsområde trækkes fra det oprindelige signal. Dette vil give to frekvenskarakteristikker, som ses på figur 3.1.



Figur 3.1: *Frekvensrespons ved +12dB og -12dB i baskontrollen.*

Dette kan også illustreres ved at tegne et blokdiagram for de to yderpositioner. Dette er vist på figur 3.2 på næste side. Her ses det, at lavpas filtreringen enten skal ske ved inputtet (+12dB) eller ved det negative output (-12dB), og at filtrering enten skal lægges til det oprindelige signal eller trækkes fra.

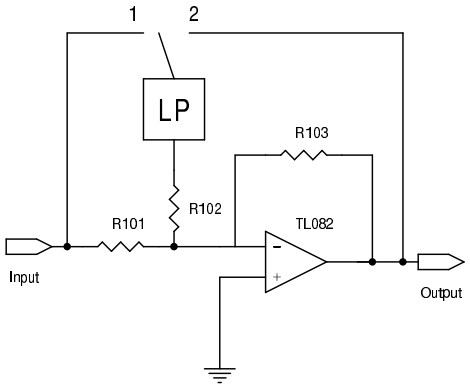
Til realiseringen af dette bruges en inverterende summationskobling. Denne er illustreret på figur 3.3 på modstående side. Kontakten i koblingen kan enten koble inputtet (1), eller det negative output (2) gennem filtret. G er gainfaktoren, der bestemmer hvor meget



Figur 3.2: *Blokdiagram for hhv. hævning og sænkning af bassen.*

basområdet maksimalt skal kunne hæves eller sænkes. Denne forstærkning sker enten i filtret eller ved den modstand, der sidder lige efter filtret (R_{102}, se evt. figur 3.3).

Koblingen vil i de to yderpositioner kunne hæve eller sænke bassens frekvensområde med 12dB. Ved at erstatte kontakten med et potentiometer opnås en glidende overgang mellem de to yderpositioner. Denne kobling kan bruges til at konstruere bas-, mellemtone- og diskantkontrollen, dog med forskellige filtre.



Figur 3.3: *Princippet i den benyttede summationskobling.*

3.3.1 Valg af operationsforstærker

I realiseringen af equalizeren benyttes operationsforstærkere. Derfor ses der kort på hvilke, der er umiddelbart tilgængelige, hvorefter den benyttede operationsforstærker vælges. Denne operationsforstærker vil blive benyttet i equalizeren, men også i de andre analoge moduler. De undersøgte operationsforstærker er opstillet med nøgledata samt pris (taget fra *http://www.rs-components.dk*) i tabel 3.1 på næste side.

I valget af operationsforstærkeren ses der på det behov operationsforstærkeren skal opfylde samt på prisen. Argumentet for at bruge en dyr operationsforstærker er, at operationsforstærkeren skal bruges til musiksignaler, hvorved det er ønskeligt, at signalerne kommer så urørt igennem forstærkeren som muligt. Argumentet for at bruge en billig operationsforstærker er, at signal/støj forholdet på guitaren kun er omkring 60dB, samt at operationsforstærkeren kun skal behandle signaler op til 11kHz.

	Slew Rate	CMRR	A_V	BW	I_{offset}	V_{offset}	Pris/stk.
LM324	ikke opg.	85dB	100dB	1MHz	5nA	2mV	1,15 kr
TLE2074	38 V/μs	89dB	106dB	9.4MHz	5pA	0.34mV	11,37kr
OP27	2,8 V/μs	120dB	125dB	8MHz	12nA	10uV	49,50kr
TL084	13 V/μs	86dB	126dB	3MHz	5pA	3mV	2,25kr
μA741	0,5V/μs	90dB	126dB	ikke opg.	20nA	1mV	6,30kr

Tabel 3.1: Nøgledata for operationsforstærkere.

Det er valgt at benytte en operationsforstærker med en lav offset strøm og en lav offset spænding, og som samtidig ikke koster meget. Denne er TL08x, som kommer i pakninger med 1, 2 og 4 operationsforstærkere (hvor x er antallet af operationsforstærkere). Denne operationsforstærker vil blive brugt i samtlige designede kredsløb.

3.3.2 Valg af filterorden

I designet af filtret tages der først udgangspunkt i kravet om en maksimal flad frekvenskarakteristik i pasbåndet, samt det enkelte pasbånds indvirken på andre bånd. For at imødekomme kravet om en maksimal flad frekvenskarakteristik benyttes Butterworth filtre *[[4], s. 29]* . For at finde ud af flankestejlheden på disse filtre, ses der på kravet om, at det enkelte pasbånds indvirken på andre bånd skal begrænses til maksimalt 25%, og at dæmpningen i pasbåndet må være maksimalt 3dB. Derved kan filtrets orden findes. Da der forstærkes 4 gange (12dB) må amplituden i det næste filter maksimalt være 1 gang (0dB). Dette betyder, at signalet skal dæmpes 12dB før det efterfølgende filter. Ud fra dette kan ordenen findes. Denne er ved et Butterworth filter givet ved (*[[4], s. 38-39]*):

$$N = \frac{logM}{log\Omega} \tag{3.1}$$

hvor M:

$$M = \sqrt{\frac{10^{0,1K_s} - 1}{10^{0,1K_p} - 1}} \tag{3.2}$$

K_s og K_p er dæmpning i henholdsvis stopbånd og pasbånd, frekvensensforholdet Ω findes ved:

$$\Omega = \frac{f_s}{f_p} \tag{3.3}$$

hvor f_s og f_p er henholdsvis stopbåndsfrekvensen og pasbåndsfrekvensen. Udfra disse ligninger ses det, at dæmpningen i stop- og pasbånd er den samme for alle tre bånd, hvilket betyder at M er ens for alle bånd. Denne er (se formel 3.3):

$$M = \sqrt{\frac{10^{1,2} - 1}{10^{0,3} - 1}} = 3,86 \tag{3.4}$$

Det ses også (jvf. kravene), at forholdet Ω er fælles for alle båndene. Denne er (udregnet for bassen):

$$\Omega = \frac{400}{200} = 2 \tag{3.5}$$

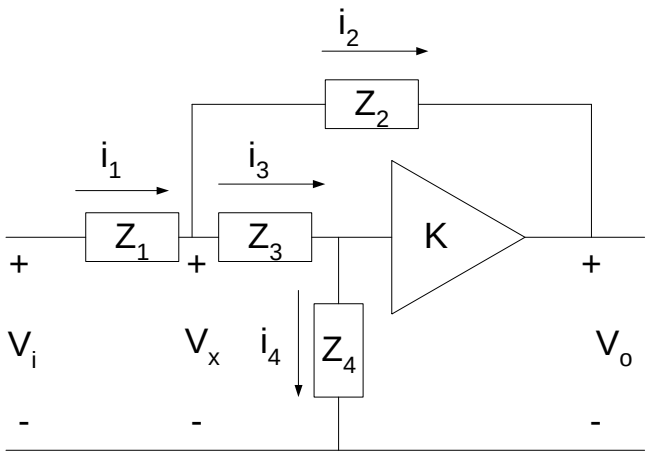
Derved bliver den nødvendige orden:

$$N = \frac{logM}{log\Omega} = 1,94 \approx 2 \tag{3.6}$$

Da ordenen på filtrene kendes, kan de tre filtre nu designes udfra grundkoblingen på figur 3.3 på side 17.

3.3.3 De benyttede filtre

Dette afsnit fastlægger de benyttede filtre. Der benyttes 2. ordens Butterworth Sallen & Key filtre til realisationen af bas-, mellemtone- og diskantkontrollen. Der tages udgangspunkt i koblingen, der ses på figur 3.4.



Figur 3.4: *Sallen & Key 2. ordens filter*

Overføringsfunktionen for filtret findes ved at se på strømmene i systemet.

$$i_1 - i_2 - i_3 = 0 \tag{3.7}$$

hvor

$$i_3 = i_4 \tag{3.8}$$

Udfra disse to ligninger kan overføringsfunktionen opstilles:

$$i_1 - i_2 - i_4 = \frac{V_i - V_x}{Z_1} - \frac{V_x - V_o}{Z_2} - \frac{\frac{V_o}{K}}{Z_4} = 0 \tag{3.9}$$

$$\Updownarrow$$

$$\frac{V_i}{Z_1} - \frac{V_x}{Z_1} - \frac{V_x}{Z_2} + \frac{V_o}{Z_2} - \frac{V_o}{K \cdot Z_4} = 0 \tag{3.10}$$

Så findes V_x:

$$i_3 = i_4 = \frac{V_x - \frac{V_o}{K}}{Z_3} = \frac{\frac{V_o}{K}}{Z_4} \tag{3.11}$$

$$V_x = V_o \cdot \frac{Z_4 + Z_3}{K \cdot Z_4} \tag{3.12}$$

Indsættes V_x i formel 3.10 findes overføringsfunktionen:

$$\frac{V_i}{Z_1} - \frac{V_o \cdot (Z_4+Z_3)}{K \cdot Z_1 \cdot Z_4} - \frac{V_o}{K \cdot Z_4} - \frac{V_o \cdot (Z_3+Z_4)}{Z_2 \cdot Z_4 \cdot K} + \frac{V_o}{Z_2} = 0 \tag{3.13}$$

$$\Updownarrow$$

$$\frac{V_i}{Z_1} = V_o \cdot \left(\frac{Z_4+Z_3}{K \cdot Z_1 \cdot Z_4} + \frac{1}{K \cdot Z_4} + \frac{Z_3+Z_4}{Z_2 \cdot Z_4 \cdot K} - \frac{1}{Z_2} \right) \tag{3.14}$$

$$\Updownarrow$$

$$\frac{V_i}{Z_1} = V_o \cdot \left(\frac{Z_2 \cdot (Z_3+Z_4) + Z_1 \cdot (Z_3+Z_4) + Z_1 \cdot Z_2 - Z_1 \cdot Z_4 \cdot K}{Z_1 \cdot Z_2 \cdot Z_4 \cdot K} \right) \tag{3.15}$$

$$\Updownarrow$$

$$\frac{V_o}{V_i} = \frac{Z_4 \cdot Z_2 \cdot K}{(Z_3+Z_4) \cdot (Z_1+Z_2) + Z_1 \cdot Z_2 - Z_1 \cdot Z_4 \cdot K} \tag{3.16}$$

Bas-, mellemtone-, og højpasfiltrene

For at lave et lavpasfilter sættes $Z_1 = R_1$, $Z_2 = \frac{1}{s \cdot C_2}$, $Z_3 = R_3$ og $Z_4 = \frac{1}{s \cdot C_4}$. Derved fås følgende overføringsfunktion:

$$\frac{V_o}{V_i} = \frac{\frac{K}{s^2 \cdot C_2 \cdot C_4}}{(R_3 + \frac{1}{s \cdot C_4}) \cdot (R_1 + \frac{1}{s \cdot C_2}) + \frac{R_1}{s \cdot C_2} - \frac{K \cdot R_1}{s \cdot C_4}} \tag{3.17}$$

$$\Updownarrow$$

$$\frac{V_o}{V_i} = \frac{\frac{K}{C_2 \cdot C_4 \cdot R_1 \cdot R_3}}{s^2 + s \cdot (\frac{1}{C_2 \cdot R_1} + \frac{1}{R_3 \cdot C_4} + \frac{1}{R_3 C_2} - \frac{K}{R_3 \cdot C_4}) + \frac{1}{R_1 \cdot R_3 \cdot C_2 \cdot C_4}} \tag{3.18}$$

Højpasfiltret findes ved $Z_1 = \frac{1}{s \cdot C_1}$, $Z_2 = R_2$, $Z_3 = \frac{1}{s \cdot C_3}$ og $Z_4 = R_4$. Derved fås følgende overføringsfunktion:

$$\frac{V_o}{V_i} = \frac{s^2 \cdot K}{s^2 + s \cdot (\frac{1}{C_1 \cdot R_2} + \frac{1}{R_4 \cdot C_3} + \frac{1}{R_4 \cdot C_1} - \frac{K}{C_1 \cdot R_2}) + \frac{1}{R_2 \cdot R_4 \cdot C_1 \cdot C_3}} \tag{3.19}$$

Til at konstruere båndpasfiltret bruges en lavpas- og en højpas-sektion.

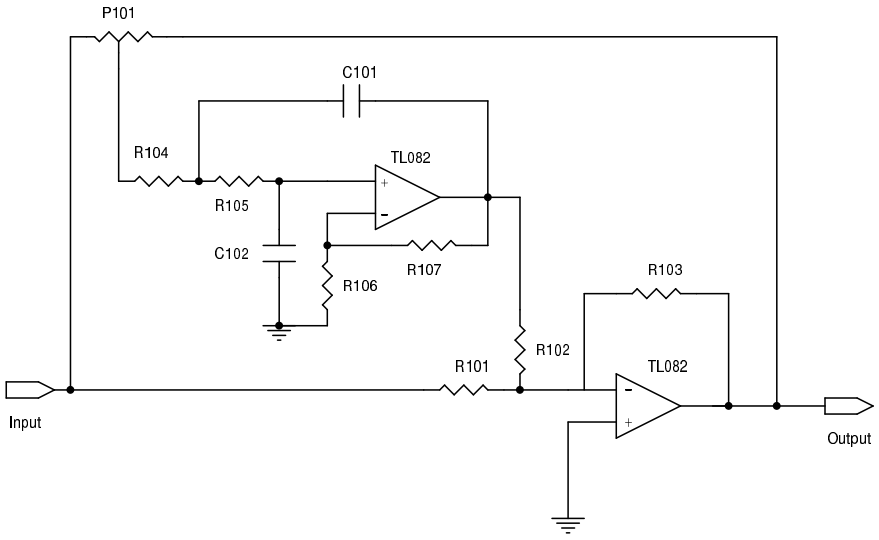
3.3.4 Baskontrollen

Da der benyttes 2. ordens Butterworth Sallen & Key filtre til lavpasfiltret til baskontrollen, bliver løsningsmodellen som på figur 3.5 på modstående side. Før filtret kan designes og konstrueres, er det nødvendigt at kende knækfrekvensen for filtret.

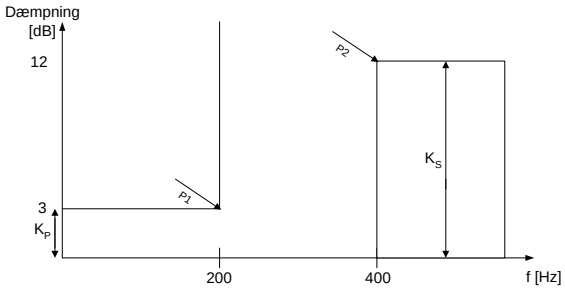
Filtrets knækfrekvens
Først ses der på kravet til dæmpningen af pasbånd og stopbånd. Dette er illustreret på figur 3.6 på næste side.

Som det, ses skal dæmpningen være maksimalt 3dB i pasbåndet samt 12dB i stopbåndet. Ved at udregne knækfrekvensen for de to yderpunkter (P_1 og P_2) kan den endelige knækfrekvens (f_n) findes mellem de to knækfrekvenser.

$$L(f) = 10 \cdot log\left(1 + \left(\frac{f}{f_n}\right)^{2 \cdot N}\right) \tag{3.20}$$



Figur 3.5: *Baskontrollens løsningsmodel.*



Figur 3.6: *Kravet om dæmpning.*

Hvor N er filtrets orden. Indsættes 3dB ved 200Hz og 12dB ved 400Hz fåes følgende to knækfrekvenser:

$$3dB = 10 \cdot log\left(1 + \left(\frac{200Hz}{f_{P1}}\right)^4\right) \tag{3.21}$$

$$\Updownarrow$$

$$f_{P2} = 200Hz \tag{3.22}$$

$$12dB = 10 \cdot log\left(1 + \left(\frac{400Hz}{f_{P2}}\right)^4\right) \tag{3.23}$$

$$\Updownarrow$$

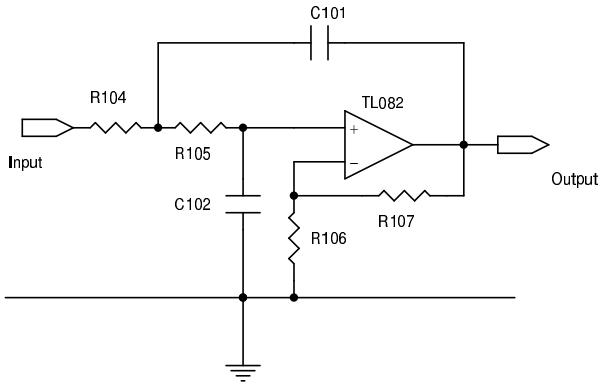
$$f_{P2} = 284Hz \tag{3.24}$$

Det vælges at lægge 3dB knækfrekvensen ved f_n=242Hz.

Lavpasfiltret
Der tages udgangspunkt i det normerede kredsløb for et 2. ordens system (dvs. pasbåndsforstærkning lig 1 og knækfrekvens ω_n=1 rad/s):

$$H(s) = \frac{1}{s^2 + \sqrt{2} \cdot s + 1} \tag{3.25}$$

Dette sammenlignes med kredsløbet for et 2. ordens Sallen & Key lavpasfilter, der ses på figur 3.7.



Figur 3.7: *Sallen & Key lavpasfilter.*

Overføringsfunktionen for Sallen & Key filtret er (se formel 3.18):

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{K}{R_{104} \cdot R_{105} \cdot C_{101} \cdot C_{102}}}{s^2 + s \cdot (\frac{1}{R_{105} \cdot C_{102}} + \frac{1}{R_{104} \cdot C_{101}} + \frac{1}{R_{105} \cdot C_{101}} - \frac{K}{R_{105} \cdot C_{102}}) + \frac{1}{R_{104} \cdot R_{105} \cdot C_{101} \cdot C_{102}}} \tag{3.26}$$

Ved at normere filtret således at $C_{101} = C_{102} = 1F$, $K = 2$ og $\omega_n = 1rad/s$ kan de normerede R_{104} og R_{105} findes:

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{K}{R_{104} \cdot R_{105}}}{s^2 + s \cdot \frac{1}{R_{104}} + \frac{1}{R_{104} \cdot R_{105}}} \tag{3.27}$$

Da godheden Q for 2. ordens Butterworth filtre er $\frac{1}{\sqrt{2}}$ og knækfrekvensen ω_n kendes, kan R_{104} og R_{105} findes:

$$R_{104} = \frac{1}{\sqrt{2}} \qquad R_{105} = \sqrt{2} \tag{3.28}$$

Filtret frekvensdenormeres med $\omega_n = 484 \cdot \pi rad/s$ og impedansdenormeres med $Z_m = 65760$. Konstanten til denormeringen af impedansen er valgt for at få en standard kondensatorværdi. Ved denne denormering fås følgende komponenter:

$$C_{101} = C_{102} = \frac{1}{\omega_n \cdot Z_m} = \frac{1}{484 \cdot \pi \cdot 65760} \approx 10nF \tag{3.29}$$

$$R_{104} = \frac{1}{\sqrt{2}} \cdot Z_m = 46.5k\Omega \doteq 46.3k\Omega \tag{3.30}$$

$$R_{105} = \sqrt{2} \cdot Z_m = 93k\Omega \doteq 93.1k\Omega \tag{3.31}$$

For at filtret kan få forstærkningen 2 sættes $R_{106} = R_{107} = 10k\Omega$, idet den ikke-inverterende forstærkning i koblingen er givet ved:

$$\frac{V_{out}}{V_{in}} = 1 + \frac{R_{107}}{R_{106}} \tag{3.32}$$

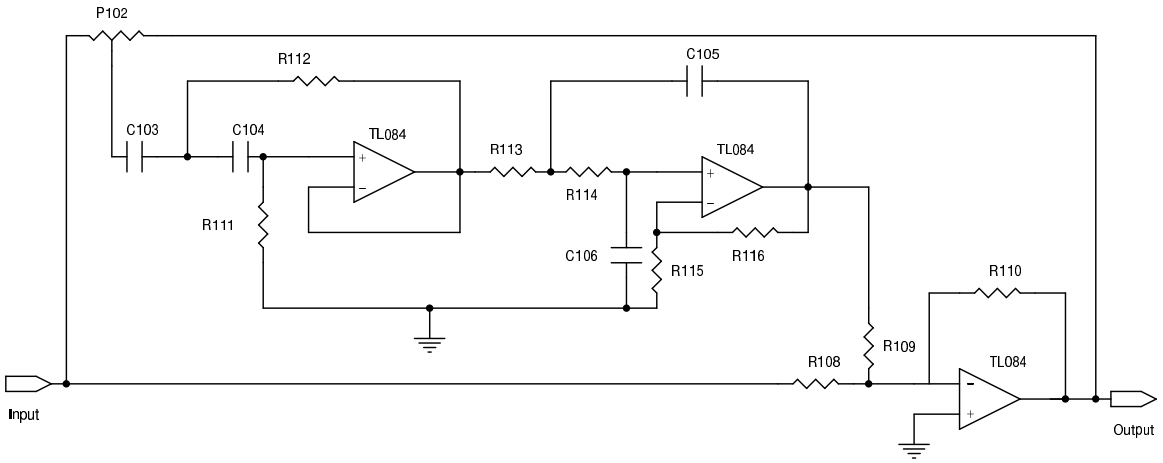
Summationskobling
Inden filtret kan konstrueres, skal forstærkningen i summationskoblingen bestemmes (se evt. figur 3.5 på side 21). Signalet skal passere uhindret igennem og filterforstærkningen skal ligge på sammenlagt 3 (12dB i yderpositionerne). Da forstærkningen i selve filtret er på 2, skal der yderligere forstærkes 1,5 gange. Forstærkningen i summationskoblingen er:

$$V_{out} = -\frac{R_{103}}{R_{101}} \cdot V_{in} - \frac{R_{102}}{R_{101}} \cdot V_{lp} \tag{3.33}$$

V_{lp} er output fra lavpasfiltret. Ved at vælge $R_{103} = R_{101} = 60k\Omega$ og $R_{102} = 40k\Omega$ fås den ønskede forstærkning. Ydermere vælges potentiometret P_{101} til $100k\Omega$, da dette er en standard værdi. Værdien af dette potentiometer er ikke kritisk.

3.3.5 Mellemtonekontrollen

Filtret til mellemtonen konstrueres af et højpas- og et lavpasfilter. Løsningsmodellen ses på figur 3.8.



Figur 3.8: *Mellemtone kontrollen.*

Q værdien for et Butterworth filter er $\frac{1}{\sqrt{2}}$ og knækfrekvenserne er fundet i MATLAB ved:

```
[N,wn] = BUTTORD[wp,ws,Rp,Rs]
```

Hvor ω_p er pasbåndsfrekvensen (400-1250Hz), ω_s er stopbåndsfrekvensen, R_p og R_s er dæmpningen i hhv. pasbåndet og stopbåndet. Funktionen leverer den nødvendige orden samt knækfrekvenser. Disse er ved N=2:

$$\omega_{n1} = 2089 rad/s \qquad \omega_{n2} = 9451 rad/s \tag{3.34}$$

Højpasfiltret

Formlen for et standard 2. ordens højpasfilter er:

$$H(s) = \frac{s^2}{s^2 + \frac{\omega_n}{Q} \cdot s + \omega_n^2} \tag{3.35}$$

Da $Q = \frac{1}{\sqrt{2}}$ og $\omega_n = 2089 rad/s$ kan overføringsfunktionen findes:

$$H(s) = \frac{s^2}{s^2 + 2089 \cdot \sqrt{2} \cdot s + 2089^2} \tag{3.36}$$

Denne sammenlignes med overføringsfunktionen for et Butterworth 2. ordens højpas filter (se evt. formel 3.19):

$$\frac{s^2}{s^2 + 2089 \cdot \sqrt{2} \cdot s + 2089^2} = \frac{s^2 \cdot K}{s^2 + s \cdot (\frac{1}{C_{103} \cdot R_{112}} + \frac{1}{R_{111} \cdot C_{104}} + \frac{1}{R_{111} \cdot C_{103}} - \frac{K}{C_{103} \cdot R_{112}}) + \frac{1}{R_{112} \cdot R_{111} \cdot C_{103} \cdot C_{104}}} \tag{3.37}$$

Vælges K=1 fås:

$$\frac{s^2}{s^2 + 2089 \cdot \sqrt{2} \cdot s + 2089^2} = \frac{s^2 \cdot 1}{s^2 + s \cdot (\frac{1}{R_{111} \cdot C_{104}} + \frac{1}{R_{111} \cdot C_{103}}) + \frac{1}{R_{112} \cdot R_{111} \cdot C_{103} \cdot C_{104}}} \tag{3.38}$$

Så vælges kondensatorerne til 33nF, hvorved:
$R_{111} = 8,55k\Omega \doteq 8,45k\Omega$
$R_{112} = 17,09k\Omega \doteq 16,9k\Omega$

Lavpasfiltret

Filteret konstrueres med $Q = \frac{1}{\sqrt{2}}$ og $\omega_n = 9451$ rad/s og en pasbåndsforstærkning på 2 gange. Dette giver følgende overføringsfunktion:

$$H(s) = \frac{2 \cdot 9451}{s^2 + 9451 \cdot \sqrt{2} \cdot s + 9451^2} \tag{3.39}$$

Denne sammenholdes med overføringsfunktionen for 2. ordens Sallen & Key:

$$\frac{2 \cdot 9451}{s^2 + 9451 \cdot \sqrt{2} \cdot s + 9451^2} =$$

$$\frac{\frac{K}{R_{113}\cdot R_{114}\cdot C_{105}\cdot C_{106}}}{s^2 + s\cdot(\frac{1}{R_{114}\cdot C_{106}} + \frac{1}{R_{113}\cdot C_{105}} + \frac{1}{R_{114}\cdot C_{105}} - \frac{K}{R_{114}\cdot C_{106}}) + \frac{1}{R_{113}\cdot R_{114}\cdot C_{105}\cdot C_{106}}} \tag{3.40}$$

Vælges K=2 og $C_{105} = C_{106} = 10nF$ fås:
$R_{113} = 8,837k\Omega \doteq 8,87k\Omega$
$R_{114} = 17,67k\Omega \doteq 17,8k\Omega$
$R_{115} = 10k\Omega$
$R_{116} = 10k\Omega$

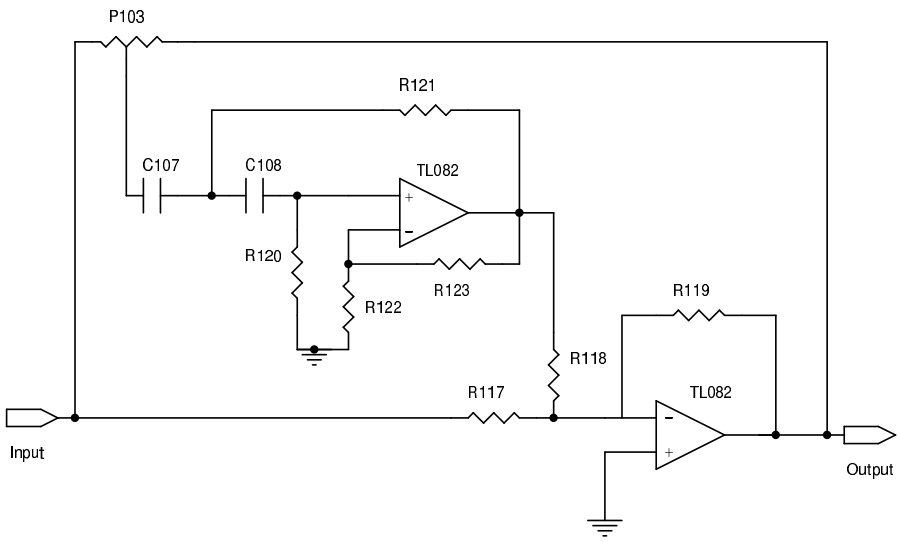
Summationskobling

Forstærkningen i pasbåndet er ikke helt på 2 gange, da lavpasfiltret knækker før højpasfiltret er på slutværdien. Derfor er det valgt at have 2 ganges forstærkning i summationskobling. Derved er valgt følgende modstandsværdier:
$R_{108} = 60k\Omega$
$R_{109} = 30k\Omega$
$R_{110} = 60k\Omega$

Potentiometret P_{102} vælges igen til standardværdien $100k\Omega$.

3.3.6 Diskantkontrollen

Til diskantkontrollen bruges et 2. ordens Butterworth Sallen & Key højpasfilter. Løsningsmodellen ses på figur 3.9. Knækfrekvensen findes igen vha. MATLAB til $\omega_n = 9532$ rad/s.



Figur 3.9: *Diskantkontrollens løsningsmodel.*

Dette giver følgende overføringsfunktion:

$$H(s) = \frac{s^2}{s^2 + s \cdot \frac{\omega_n}{Q} + \omega_n^2} \tag{3.41}$$

Hvor $Q = \frac{1}{\sqrt{2}}$ og $\omega_n = 9532\text{rad/s}$. Dette sammenlignes med formlen for Sallen & Key højpasfilteret (se evt. fml 3.19):

$$\frac{s^2}{s^2 + s \cdot \sqrt{2} \cdot 9532 + 9532^2} =$$

$$\frac{s^2 \cdot K}{s^2 + s \cdot (\frac{1}{C_{107} \cdot R_{121}} + \frac{1}{R_{120} \cdot C_{108}} + \frac{1}{R_{120} \cdot C_{107}} - \frac{K}{C_{107} \cdot R_{121}}) + \frac{1}{R_{120} \cdot R_{121} \cdot C_{107} \cdot C_{108}}} \tag{3.42}$$

K vælges til 2 og $C_{107} = C_{108} = 2nF$. Derved fås:
$R_{120} = 28,3k\Omega \doteq 28,7k\Omega$
$R_{121} = 37,1k\Omega \doteq 37,4k\Omega$
$R_{122} = 10k\Omega$
$R_{123} = 10k\Omega$

Summationskobling

Da pasbåndsforstærkningen er 2 findes:
$R_{117} = 60k\Omega$
$R_{128} = 40k\Omega$
$R_{129} = 60k\Omega$
Potentiometret P_{103} vælges til $100k\Omega$.

Det samlede diagram for equalizeren, kan ses i bilag II.

3.3.7 Delkonklusion

Equalizeren testes og simuleres i appendiks C.1.2, hvor det konkluderes, at equalizeren virker efter hensigten. Dog fremkommer der nogle over/undersving i starten af pasbåndet, der gør at indstilling af alle 3 parametre på een gang ikke giver den frekvenskarakteristik, der forventes.

KAPITEL 4

Forvrænger

Formålet med dette kapitel er at designe et lydmodul, der skal forvrænge guitarsignalet. I dette afsnit konstrueres og realiseres løsningsmodellen.

Afsnittet tager udgangspunkt i kilden: [15] og [11].

4.1 Krav til forvrænger

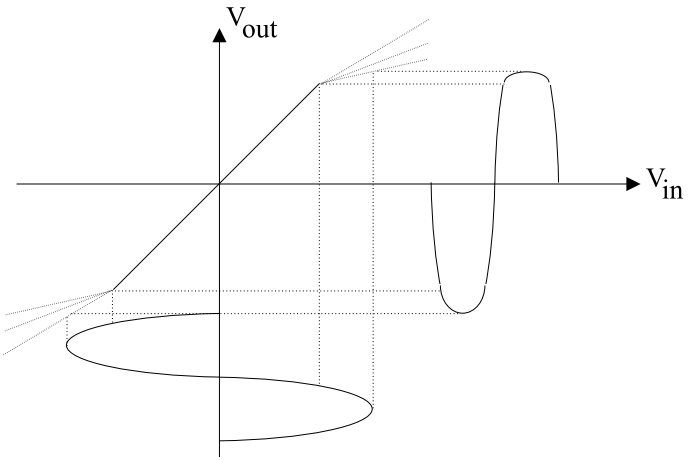
Der opstilles følgende krav til forvrængeren:

- Forskellige forvrængningsniveauer.
- Forvrængning af både små og store signaler.
- Valg mellem symmetrisk og asymmetrisk forvrængning.
- Øvrige hardwarekrav fra kravspecifikationen.

Med de forskellige forvrængningsniveauer skal det være muligt at vælge hårdheden af klipningen, således der kan vælges mellem "blød" (overdrive) og "hård" (distortion) klipning. Der ønskes en klipningskarakteristik som vist på figur 4.1 på den følgende side, hvor udgangssignalet klippes over et bestemt spændingsniveau.

4.2 Løsningsmuligheder til forvrængeren

Ud fra den ønskede klipningskarakteristik som vist på figur 4.1 på næste side, ses det, at der skal en ulineær forstærkning til at realisere forvrængeren. Af ulineære komponenter

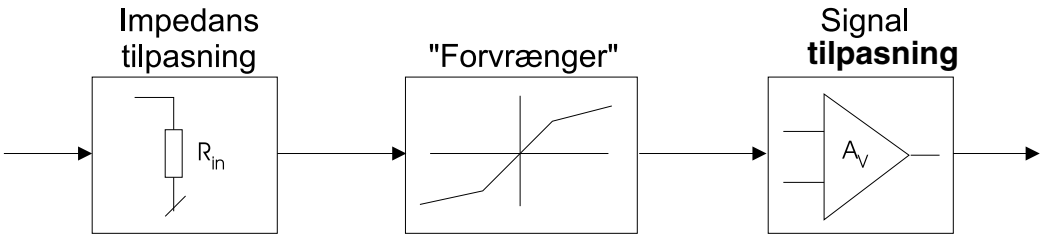


Figur 4.1: *Den ønskede klipningskarakteristik.*

kan nævnes:

- Dioden.
- Transistoren.

Ulineariteten i transistoren udgøres af diodestrækningen, og derfor vælges der kun at kigge nærmere på dioden. Se appendiks B for måling af de forskellige diodetypers karakteristikker.
Ud fra disse målinger vælges der, pga. indre modstand, at arbejde videre med silicium dioden (1N4148) og germanium dioden (AA119).
For at imødekomme kravet om en indgangsimpedans på mindst 10kΩ (se hardwarekrav kapitel 2 afsnit 2.2.1 på side 10), skal forvrængeren have en impedanstilpasning på indgangen. Det generelle løsningsforslag til forvrængeren, ses på figur 4.2.



Figur 4.2: *Blokdiagram over generel løsningsforslag til forvrængermodulet.*

Som nævnt tidligere kan dioden bruges til at forvrænge et signal ved hjælp af dens ulinearitet. Der vælges at koble dioderne med forstærkning, da indgangssignalet også skal forvrænges ved små amplituder. Forvrængeren kan kobles op på to måder: Enten med dioderne i tilbagekoblingen (se figur 4.3 på modstående side), eller med dioderne på udgangen (se figur 4.4 på næste side).
Der kan anvendes 2 typer operationsforstærkerkoblinger: Den inverterende og den ikke-inverterende kobling. Den ikke-inverterende kobling har en høj indgangsimpedans, og en

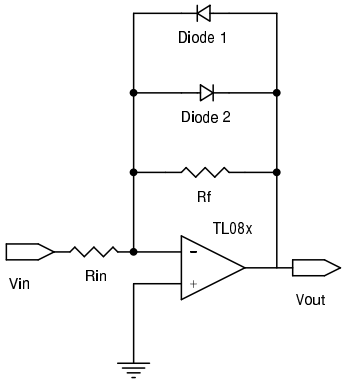
forstærkning på:

$$A_V = 1 + \frac{R_f}{R_{in}} \tag{4.1}$$

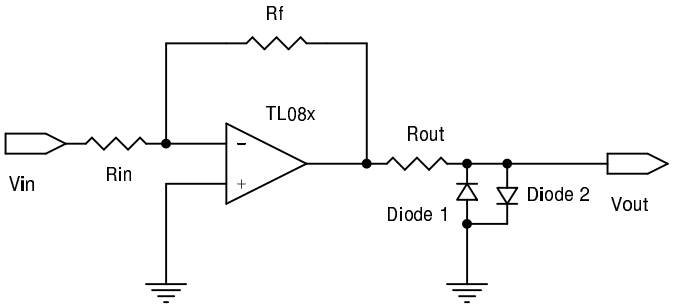
Den inverterende kobling har en indgangsimpedans på R_{in} og en forstærkning på:

$$A_V = -\frac{R_f}{R_{in}} \tag{4.2}$$

Som udgangspunkt vælges det at se på den inverterende kobling, da den både kan forstærke og dæmpe.



Figur 4.3: *Forvrænger med dioder i tilbagekoblingen (design 1).*

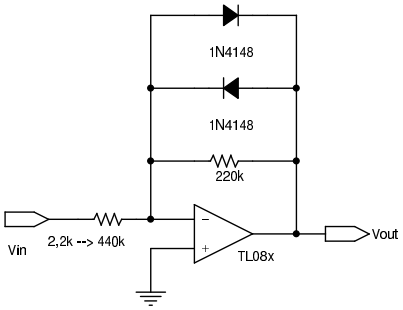


Figur 4.4: *Forvrænger med dioder på udgangen (design 2).*

Desuden er der indsat to modsatrettede dioder for at få både de positive og negative spændingsniveauer forvrænget.

4.2.1 Dioder i tilbagekobling (design 1)

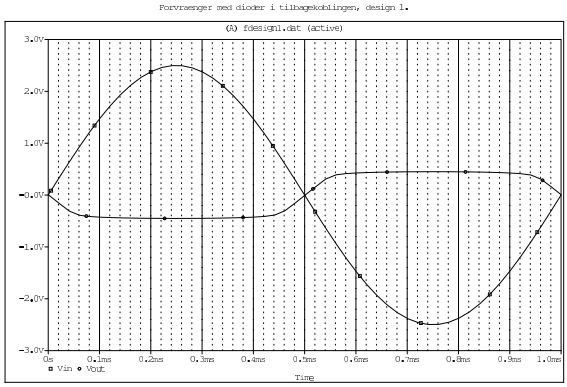
I design 1 sidder dioderne over tilbagekoblingsmodstanden på operationsforstærkeren. Ved at ændre indgangsmodstanden, R_{in}, ændres koblingens forstærkning. Funktionen af design 1 er, at inputsignalet forstærkes lineært op, indtil spændingen over dioderne er over diodespændingen, hvorefter dioderne klipper signalet. Det vælges at simulere design 1 (figur 4.5 på den følgende side) i Spice, for at se hvilken effekt ændringen af forstærkningen i koblingen giver.



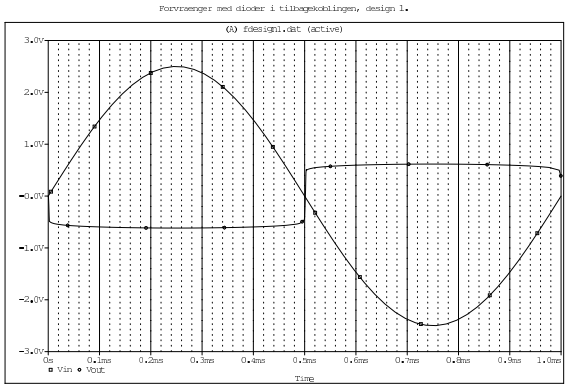
Figur 4.5: *Design 1 med valgte komponentværdier.*

Der blev simuleret med en sinus på $2{,}5V_p$ amplitude på input og med henholdsvis en forstærkning på 0,5 gange og 100 gange.

Komponentværdierne er valgt som vist på figur 4.5, og resultaterne ses på figur 4.6 og 4.7, hvor både ind- og udgangssignalet er afbildet.



Figur 4.6: *Design 1 med en forstærkning på 0,5 gange.*



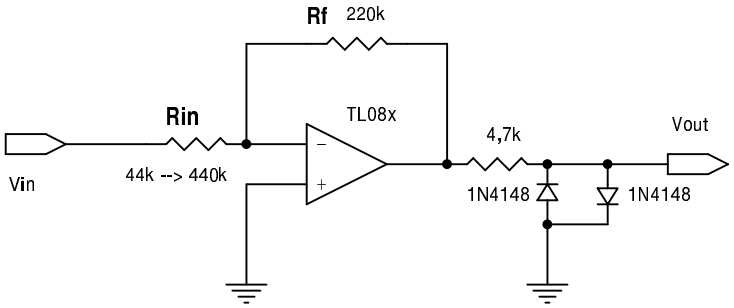
Figur 4.7: *Design 1 med en forstærkning på 100 gange.*

Sammenlignes de 2 figurer, ses det, at ved forstærkningen på 0,5, er flankerne på udgangssignalet "pæne", hvorimod forstærkningen på 100 gange skaber meget stejle flanker. Det

er flankestejlheden der giver forvrængningsgraden. Design 1 giver altså mulighed for at ændre forvrængningsgraden fra næsten uforvrænget til kraftigt forvrænget.

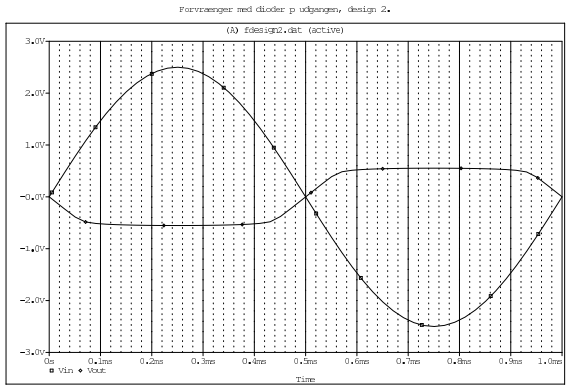
4.2.2 Dioder i udgangen (design 2)

I design 2 sidder dioderne på udgangen, og dermed kan udgangsspændingen aldrig komme over diodespændingsfaldet (Se figur 4.4 på side 29). Først forstærkes signalet op i operationsforstærkerkoblingen, og når udgangsspændingen kommer op over diodespændingen, åbner dioderne, og klipper signalet.
Der laves en Spice simulering for at finde ud af, hvad der sker, når forstærkningen i koblingen ændres (R_{in} på figur 4.8 ændres).



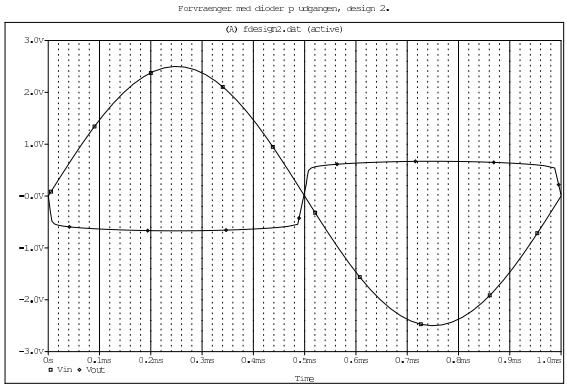
Figur 4.8: *Design 2 med valgte komponentværdier.*

Der simuleres med $2{,}5V_p$ sinus på inputtet, og forstærkningen sættes til henholdsvis 0,5 og 5 gange. Med et signal på $2{,}5V_p$ må forstærkningen ikke komme over 5 gange, da operationsforstærkeren ellers vil klippe på grund af at signalet overstiger forsyningsspændingen.



Figur 4.9: *Design 2 med en forstærkning på 0,5 gange.*

Figur 4.9 viser design 2 med en forstærkning på 0,5 gange, og figur 4.10 på næste side viser design 2 med 5 ganges forstærkning. Ændringen på flankestejlheden, og dermed forvrængningsgraden, ændres ikke tilstrækkeligt på den sidste simulering (design 2).



Figur 4.10: *Design 2 med en forstærkning på 5 gange.*

4.2.3 Valg af forvrængertype

På baggrund af simuleringerne foretaget i afsnit 4.2.1 på side 29 og afsnit 4.2.2 på foregående side vælges design 1 til forvrængeren. Grunden hertil er, at forvrængningsgraden kan justeres ved at ændre på indgangsmodstanden, og der opnås kraftig forvrængning.
Forvrængningen i design 2 bliver ikke nær så kraftigt som i design 1, da stejlheden på udgangssignalets flanker, ikke kan blive stor nok i design 2 - på grund af den lave forstærkning.

4.3 Design af forvrænger

Kravet om en indgangsimpedans $\geq 10\text{k}\Omega$ opnås ved at koble en buffer foran forvrængeren. Denne buffer realiseres med en spændingsfølger, hvor det inverterende input kobles til outputtet, og inputsignalet sendes ind på den ikke-inverterende indgang.
Kravet om både lige og ulige forvrængning (symmetrisk og asymmetrisk forvrængning) kan opnås ved at anvende 2 forskellige diodetyper. Symmetrisk forvrængning opnås ved at benytte ens diodetyper. Den asymmetriske forvrængning laves ved at kombinere forskellige dioder. I afsnit 4.2 på side 27 blev det valgt, at benytte henholdsvis AA119 og 1N4148 som dioder til forvrængeren. Den asymmetriske forvrængning opnås ved at bruge AA119 som DIODE1 og 1N4148 som DIODE2 (se figur 4.3 på side 29).
Der sættes en modstand fra indgangen på operationsforstærkeren til stel, R_{203}, for at der kan trækkes biasstrøm. Denne modstand sættes til $1\text{M}\Omega$.

4.3.1 Valg af forvrængningsgrad

Kravet om at kunne ændre forvrængningsgraden realiseres ved at gøre indgangsmodstanden R_{in} variabel. Det vælges at have en maksimal forstærkning i trinnet på 440 gange, og en minimumsforvrængning på 1 gang. Det realiseres ved at have en tilbagekoblingsmodstand, R_{202}, på $220\text{k}\Omega$ og en indgangsmodstand på:

$$A_{v-max} = \frac{220k\Omega}{R_{in-min}} = 440 \tag{4.3}$$

$$A_{v-min} = \frac{220k\Omega}{R_{in-max}} = 1 \tag{4.4}$$

$$\Rightarrow R_{in-min} = 500\Omega, \qquad \Rightarrow R_{in-max} = 220k\Omega \tag{4.5}$$

Indgangsmodstanden realiseres med en modstand, R_{201}, på 499Ω og et potmeter, P_{201}, på 220kΩ.

Det samlede diagram over forvrængeren ses i bilag III på side 177.

4.4 Delkonklusion

Forvrængeren virker efter hensigten og overholder hardwarekravene. Forvrængeren er blevet simuleret i Spice (se appendiks C.2.1 på side 115) og det er eftervist, at den ønskede effekt opnås. Forvrængeren er ligeledes testet i laboratoriet (se appendiks C.2.2 på side 115) og målingerne viste, at forvrængeren virker efter hensigten.
Efter at have lyttet til forvrængeren, konstateres det, at der ikke er så stor lydmæssig forskel ved de forskellige forvrængningstyper. Men som det ses af FFT analyserne i appendiks C.2.2 på side 115 er der ved de tre diodekombinationer en lille variation af de lige harmoniske overtoner (hård forvrængning), der ved silicium dioderne er med cirka samme amplitude, og med germanium dioden er svagt stigende med frekvensen, og til sidst med den asymmetriske er de lige harmoniske faldende i amplituden ved højere frekvenser.

KAPITEL 5

Analog/Digital grænseflade

Dette kapitel omhandler den analoge/digitale grænseflade. Her beskrives den konverter, der varetager ændringen fra analogt til digitalt signal samt fra digital til analog. Dette værende sig et indgangstrin og den benyttede konverter.

Kapitlet bygger på [11] og databladet for den benyttede konverter.

5.1 Indgangstrin

Som det ses af kravspecifikationen, er de digitale effektenheder de første i signalvejen. Derfor skal der konstrueres et indgangstrin, der tager signalet fra guitaren, og tilpasser det til den benyttede konverter, samt overholder kravspecifikationens krav. I appendiks A på side 102 er der foretaget målinger på guitaren. Ud fra denne målerapport kan det ses, at guitaren typisk har en udgangsimpedans på omkring 100kΩ, men at der er en impedans på ca. 1MΩ ved 3,12kHz. Ud fra målingerne på guitaren mht. spændingsniveauer for forskellige anslag, kan det ses, at spændingsniveauer på over $1{,}4V_{pp}$ er meget sjældne, og kun forekommer ved meget voldsomme anslag. Derfor vurderes det, at det er unødvendigt at sende spændingsniveauer over $1{,}4V_{pp}$ videre i kredsløbet. Ud fra dette kan der opstilles følgende krav til indgangsforstærkeren:

- Indgangsmodstand på mindst 1MΩ.
- "Blød" klipning af indgangssignalet, hvis det er over $1{,}4V_{pp}$.
- Spændingstilpasning af guitarens signal til $5V_{pp}$.
- Øvrige hardwarekrav fra kravspecifikationen.

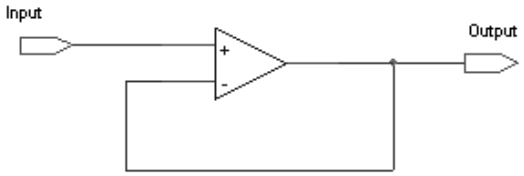
Spændingstilpasningen til $5V_{pp}$ er foretaget før valget af konverter. Derfor kan det blive nødvendigt at tilpasse signalet til konverteren.

5.1.1 Design af indgangsforstærker

Indgangsforstærkeren består af 2 kredsløb: En indgangsbuffer og et forstærkningskredsløb.

Indgangsbuffer

For at sikre en indgangsmodstand der er større end 1MΩ, bruges en operationsforstærker koblet som spændingsfølger. Denne spændingsfølger er vist på figur 5.1.



Figur 5.1: *Princip-diagram over spændingsfølger.*

Der kobles en modstand, R_{313}, på 1MΩ fra signalet til stel før spændingsfølgeren for at sikre at operationsforstærkeren kan trække biasstrøm. Ved at vælge 1MΩ er kravet om indgangsmodstanden opfyldt.

Forstærkningskredsløb/Limiterkredsløb

For at opfylde kravene om signaltilpasning, "blød" klipning og lav udgangsimpedans, bruges en inverterende operationsforstærkerkobling med 2 dioder og 4 modstande. Denne kobling ses på figur 5.2 på den følgende side. Koblingen klipper signaler over en vis amplitude samt forstærker signalet.

Ved lav inputspænding forstærker koblingen lineært, som en operationsforstærker koblet som en inverterende forstærker. Derfor er forstærkningen givet ved:

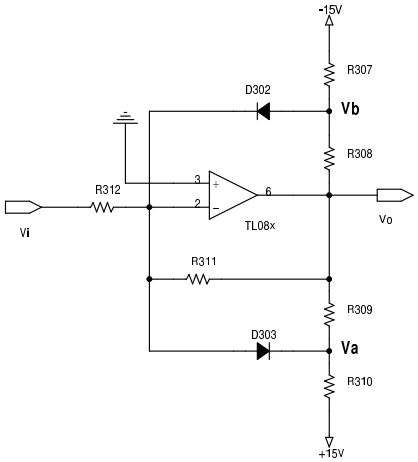
$$V_o = -V_i \cdot \frac{R_{311}}{R_{312}} \tag{5.1}$$

Udfra superposition kan spændingerne V_A og V_B findes:

$$V_A = V_- \cdot \frac{R_{309}}{R_{309} + R_{310}} + V_o \cdot \frac{R_{310}}{R_{309} + R_{310}} \tag{5.2}$$

$$V_B = V_+ \cdot \frac{R_{308}}{R_{308} + R_{307}} + V_o \cdot \frac{R_{307}}{R_{308} + R_{307}} \tag{5.3}$$

I tilfælde af at inputspændingen er positiv, er udgangsspændingen negativ, og udfra ovenstående ligninger ses det, at V_B bliver mere negativ og V_A bliver mindre positiv. Idet



Figur 5.2: *Diagram for limiterkredsen.*

V_A når et negativt spændingsniveau, på diodens tærskelspænding, begynder D_{303} at lede. Det antages, at der findes et konstant spændingsfald over dioden, V_D, hvorved der kan opstilles en formel for, hvilken spænding der skal være på outputtet før der klippes.

$$L_- = V_- \frac{R_{309}}{R_{310}} - V_D \cdot \left(1 + \frac{R_{309}}{R_{310}}\right) \tag{5.4}$$

Den spænding som kræves på indgangen for at klippe udgangssignalet kan findes udfra følgende formel:

$$V_i = \frac{L_-}{R_{311}/R_{312}} \tag{5.5}$$

Hvis V_i øges, udover dette niveau, løber der mere strøm gennem D_{303} og V_A har en spænding på omkring $-V_D$. Dette betyder, at strømmen gennem R_{310} er konstant, og at der derudover løber en strøm gennem R_{309}, hvilket medfører at R_{309} sidder i parallel med R_{311}. For outputtet betyder dette at når L_- nås, klippes signalet med en hældning (a) på:

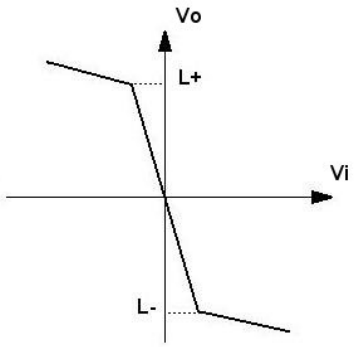
$$a = -\frac{R_{311} \parallel R_{309}}{R_{312}} \tag{5.6}$$

Ud fra denne sammenhæng ses det, at der skal vælges en lav værdi for R_{309} for at få en "hurtig" og "blød" klipning. Det samme princip gælder, når V_i er negativ, blot med den anden diode. Se figur 5.3 på næste side for at se input/output-karakteristikken for limiter kredsløbet.

Der indskydes, før limiter kredsløbet, et RC-led til højpasfiltrering, for at fjerne lavfrekvent støj.

Realisering af indgangstrin

Det vælges at bruge to "shottky barrier" dioder fra Philips af typen BAT85. De har en tærskelspænding på ca. 0,3V. Dernæst bestemmes den spænding, L_-, hvor signalet skal klippes. Det er valgt at outputtet maksimalt må være $5V_{pp}$, og at det derefter skal klippes. Derfor skal L_- være -2,5V. Det betyder at modstandene R_{309} og R_{308} skal være $0{,}144 \cdot R_{307}$ eller $0{,}144 \cdot R_{310}$, da der ønskes symmetrisk klipning. R_{307} og R_{310} vælges til



Figur 5.3: *Input/output relation for limiter kredsløbet.*

10kΩ og derfor skal R_{308} og R_{309} være ca. 1440Ω. Der vælges at indsætte et potentiometer for R_{308} og R_{309}, så man kan variere spændingen, hvor der klippes. Udfra sammenhængen mellem indgangsspændingen og L_- kan forstærkningen for kredsløbet findes:

$$V_{in} = \frac{L_-}{G} \Rightarrow G = \frac{L_-}{V_{in}} = \frac{2,5V}{0,7V} = 3,57V/V \tag{5.7}$$

Der ønskes en "blød" klipning, med en hældning på ± 0,3, hvorved følgende formel kan opstilles:

$$R_{312} \cdot 0,3 = (R_{311} \parallel R_{309}) \Rightarrow R_{312} = 3,1 \cdot R_{309} \tag{5.8}$$

Det betyder, at R_{312} skal være 4,4kΩ og R_{311} skal være 15,7kΩ. Der vælges to modstande på henholdsvis 4,42kΩ og 15,8kΩ. Filteret dimensioneres udfra et ønske om at filtrere eventuel DC-støj fra, sammenholdt med kendskabet til guitarens dybeste toner. Udfra disse betragtninger vælges en afskæringsfrekvens på 20Hz. Ved at sætte en kondensator i serie med R_{312} fåes det ønskede højpasfilter. Kondensatorens værdi beregnes til:

$$C_{319} = \frac{1}{2\pi \cdot 4.42k\Omega \cdot 20Hz} \approx 1.81\mu F \tag{5.9}$$

Kondensatoren vælges til 2,2μF, hvorved afskæringsfrekvensen bliver ca. 16Hz.

Indgangstrinnet er nu designet således, at det forstærker signaler op til 1,4V_{pp} med 3,57 gange. Signaler over denne størrelse klippes "blødt". På dette stadie er indgangstrinnet tilpasset 0-5V_{pp} og skal konverteres til et digitalt signal. I næste afsnit vælges og dimensioneres konverteren.

5.2 Valg af konverter

Da der i kravspecifikation er opstillet et krav om, at frekvensområdet strækker sig fra 60Hz-11kHz, kræves der ifølge Nyquist, at samplingsfrekvensen skal være mindst 22kHz [2]. Derudover er det ønskeligt at kunne både A/D- og D/A-konvertere, da der skal bruges et analogt signal til guitarforstærkeren. Da den udleverede DSP (TMS320C26), er en 16 bit fixed point processor er det fordelagtigt med en 16 bit konverter. På baggrund heraf opstilles følgende krav.

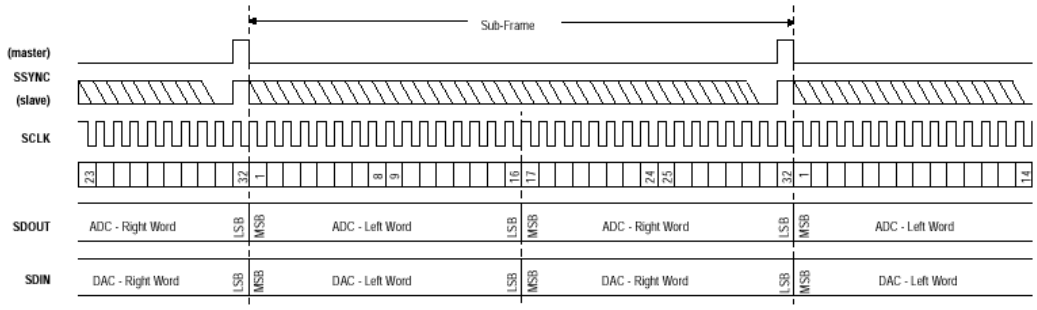
- A/D og D/A konvertering.
- 16 bit konvertering.
- Samplefrekvens på mindst 22kHz.
- Øvrige krav fra kravspecifikationen.

På baggrund af de opstillede krav til konverteren, blev der ud fra de tilgængelige modeller i komponentudleveringen, valgt at anvende CS4218-KL fra Crystal (datablad findes på CD-ROM). Følgende karakteristika gælder for denne model:

- 2 kanals seriel 16-bit Stereo Audio codec.
- Delta sigma A/D og D/A konvertere.
- Samplingfrekvens fra 4kHz til 50kHz.
- Input anti-aliasing og output smoothing filtre.
- THD $< 0{,}02\%$.

5.3 Design af codec

Ved brug af CS4218 er der tre forskellige opsætninger af serielporten: SM3, SM4 og SM5 (SM = Serial Mode). SM4 er, ifølge fabrikanten, specielt velegnet til brug sammen med forholdsvis simple DSP'ere. Det skyldes, at der i denne mode lige efter synkroniseringspulsen udsendes, kommer en 32 bit frame med 16 bit data fra den ene kanal efterfulgt af 16 bit data fra den anden kanal. Dette minimerer unødvendig kommunikation mellem CS4218 og TMS320C26, og derfor vælges det at bruge SM4. På figur 5.4 ses, hvorledes der sendes over bussen mellem DSP og codec.



Figur 5.4: *Seriel transmission mellem DSP og codec.*

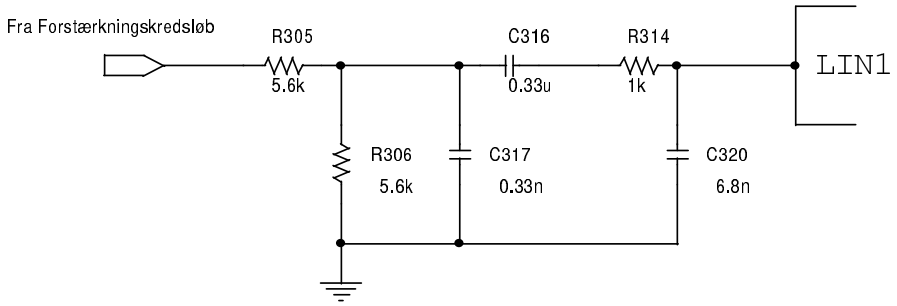
Ved brug af SM4 er der mulighed for at foretage kontrol af data ved at aflæse hvad der står på pin 40. Dette gøres dog ikke, da der ikke ønskes fejlkontrol, eftersom det ville belaste DSP'en yderligere. SM4 er yderligere inddelt i 4 submodes, der omhandler interfacet mellem konverteren og DSP'en. Denne er valgt til "Master sub-mode". Denne mode betyder at konverteren er master på bussen mellem det serielle interface til DSP'en og konverteren.

5.3.1 Opkobling af codec samt analog signaltilpasning

Det vælges at koble codec'en op som det er anbefalet i databladet (se databladet på CD-ROM). For selve opkoblingen henvises der til bilag IV på side 179.

Codec'en (CS4218-KL) er internt offset kompenseret (DC-offset på $2{,}1V_{DC}$), således at indgangssignalet ikke skal påføres et offset. Ifølge databladet skal indgangssignalet svinge mellem $\pm 1{,}25$ ($2{,}5V_{pp}$) for at få fuld udstyring.

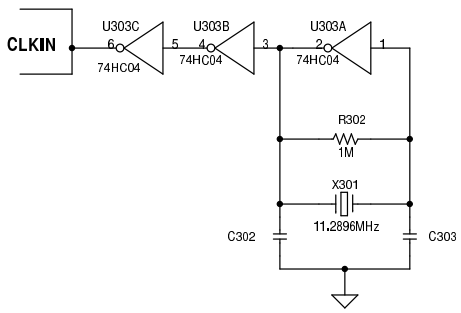
For at halvere spændingen på indgangen, som det anbefales i databladet, indsættes to modstande, R_{305} og R_{306} på 5,6kΩ. Der sættes endvidere et RC-led ind mellem kondensatoren på $0{,}33\mu$F og pin 27 for at filtrere noget højfrekvent støj fra. På figur 5.5 ses hvordan signalet kobles fra indgangstrinnet og til det analoge input.



Figur 5.5: *Indgangskredsen til analogt input.*

5.3.2 Oscillatorkredsløb

For at opnå en ønsket samplefrekvens på 22,05kHz benyttes et krystal på 11,2896MHz. Oscillatorkredsløbet er opbygget ved at afkoble krystallet gennem to kondensatorer og sætte det i parallelkobling med en 10MΩ modstand. Til at drive krystallen med anvendes en trigger af typen: HC74LS04. Den virker som en forstærker, der tilpasser niveauet til 0-5V. Den gennemløbes 2 gange for at gøre firkantsignalet bedre. På figur 5.6 ses eldiagram for oscillatorkredsløbet.



Figur 5.6: *Oscillatorkredsløbet.*

5.3.3 Reset kredsløb

For at nulstille alle interne registre på codec, skal reset-benet holdes lavt i minimum 50ms. Til reset kredsløbet er valgt at bruge afladningsegenskaberne for en kondensator i et RC-led, kombineret med en trykknap. Der er brugt en modstand på 10kΩ, samt en kondensator på $10\mu F$, som resulterer i en resettid på 100ms. Dette er tilstrækkeligt til at kalibrere codec'en.

5.4 Yderligere tiltag

Da der er koblet en modstand på 1MΩ indgangstrinnet, og der arbejdes med meget lave spændinger før forstærkningen, er der risiko for at få problemer med EMC og anden støj. Ved højimpedante netværk er det især elektriske LF-felter, der kan forsårsage støj. Støjen kan både kobles gennem induktive, kapacitive og galvaniske forbindelser. Det potentielle EMC-problem er løst ved brug af en metalskærm omkring indgangstrinnet, som er stellet med forsyningsspændingens stel. Den fjerner/dæmper elektriske felter af alle frekvenser. Endvidere er der koblet en afkoblingskondensator, C_{318} på udgangen af indgangtrinnet på 47nF.

Yderligere er der indsat en kontakt, der kan frakoble hele DSP-systemet. Derved kan signalvejen holdes helt analog. Selve kontakten og hele koblingen kan ses i bilag IV på side 179.

5.5 Delkonklusion

Der er i dette kapitel blevet designet hardware, der sørger for at indgangssignalet fra guitaren tilpasses til den benyttede codec. Der er også designet codec, der sørger for AD/DA konverteringen. Disse funktioner er testet i appendiks C.3 på side 119, hvor det konkluderes, at codec og signaltilpasning virker efter hensigten. Det er muligt, at konvertere analoge signaler til digitale signaler og derefter konvertere dette digitale signal tilbage til et analogt signal igen. Det kan derfor konkluderes, at den analoge til digitale grænseflade virker efter hensigten.

KAPITEL 6

Hardware-opkobling

Dette kapitel omhandler hardware-opkoblingen af det benyttede digitale system. Kapitlet vil indeholde en generel beskrivelse af de benyttede elementer i opkoblingen (DSP, RAM/ROM etc.). Der vil blive redegjort for systemets timing, adressedekodning samt sammenkoblingen af de forskellige delelementer.

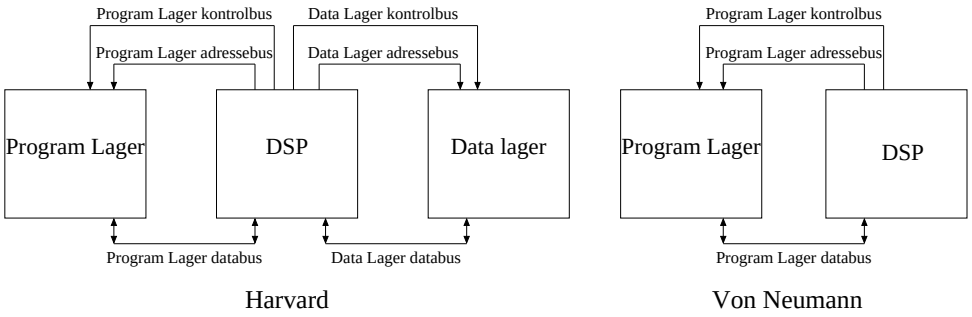
6.1 Den benyttede DSP

I projektet benyttes en DSP (Digital Signal Processor) af typen TMS320C26 fra Texas Instruments. Nøgledata for denne DSP er følgende (taget fra [[13], s. 1.4] og [[1], s. 6.37]):

- 16 bit fixed point processor.
- 10MIPS (million instructions per second).
- 40MHz clockfrekvens.
- instruktionstid 100ns.
- 16 bit databus.
- 16 bit programbus.
- Intern "Harvard" struktur.
- Ekstern "Von Neumann" struktur.
- 256word onchip ROM.

- 16 I/O porte.
- Serielt interface (recieve/transmit).
- 3 interrupts.
- 1,5kword intern onchip RAM.

Som nævnt ovenfor er TMS320C26 opbygget internt efter "Harvard" strukturen. Det betyder, at data og program hentes til processoren med hver sin bus. Dog er der kun 1 bus udadtil til eksterne enheder (RAM/ROM etc.). Dette kaldes "Von Neumann" princippet. Forskellen på de 2 strukturer er vist på figur 6.1.



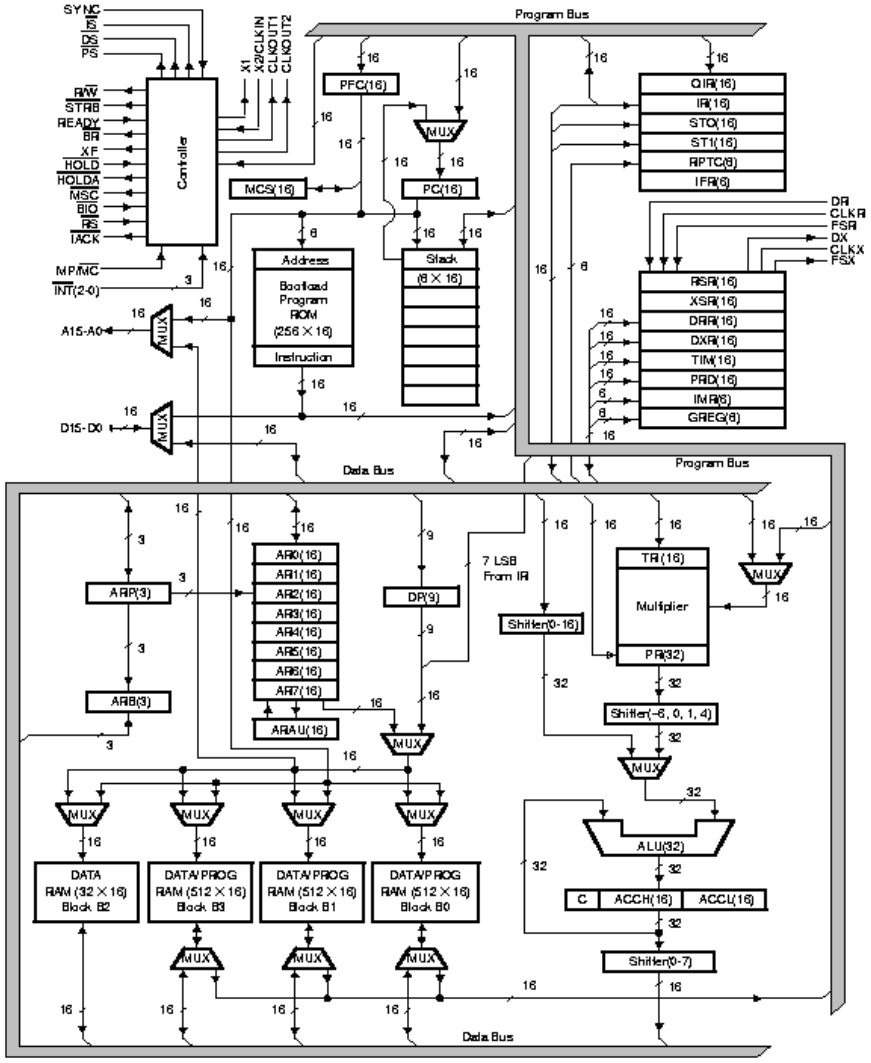
Figur 6.1: *Blokdiagram for "Harvard" og "Von Neumann" strukturen.*

TMS320C26 bruger pipeline, der hele tiden henter instruktioner og data ind. En pipeline henter de næste instruktioner ind til processoren, mens den er igang med at eksekvere. Derved køres flere processer samtidigt, hvilket øger afviklingshastigheden. Dog vil der ved "hop" (f.eks. branch) opstå et problem, da pipelinen er fyldt med instruktioner, der ikke skal eksekveres. Derfor skal pipelinen tømmes, hvilket kræver ekstra tid. Dette betyder, at DSP-processorer er bedst egnede til applikationer uden for mange "hop". Dette kunne være signalbehandling (deraf navnet DSP-processor). Opbygningen af DSP-processoren er vist på figur 6.2.

Undersøges benforbindelserne til DSP'en ses det, at det er muligt at selektere 64kword med den 16 bit adressebus. Der findes 3 ben på DSP'en, der gør, at den kan skelne mellem program, data og I/O. Disse er: $\overline{PS}$, $\overline{DS}$ og $\overline{IS}$. De gør det muligt at selektere: 64kword program, 64kword data og 16word I/O.

6.2 Den benyttede RAM kreds

Til DSP-koblingen benyttes 128kB (64kword) RAM af typen CY7C199, der er 8 bit adresserbar. Valget af CY7C199 blev truffet, da den har den hurtigste accesstid af de RAM kredse, der umiddelbart var tilgængelige. Da RAM kredsene kun er 32kB og 8 bit adresserbare, bruges 4 RAM kredse, der kobles 2 og 2 i en "RAM-blok". Derved kobles den ene RAM kreds til de nederste 8 bit (LOW) på databussen og den anden RAM kreds kobles til de øverste 8 bit (HIGH). Selve selekteringen af kredsene er ens. Det skal også



Figur 6.2: *Strukturen i den benyttede DSP (TMS320C26).*

nævnes, at da en "RAM-blok" kun udgør 32kword, skal adressedekodning sørge for at selektere mellem de 2 "RAM-blokke". Derved forbindes det øverste adresseben (A15) ikke til RAM kredsene. CY7C199 har følgende nøgledata:

- 32kB RAM.
- 5V spændingsniveau.
- 15ns access tid.
- 8 bit adresserbar.
- Mulighed for "Power Down".

6.3 Den benyttede debugger

Den udleverede debugger består af en EPROM af typen M27C1024 og en PEEL-kreds 22CV10 samt diverse kontakter, modstande, dioder og en RS232 driver (MAX232A). Selve opkoblingen er beskrevet i [[7], s. 15] og er vist i bilag V på side 181. Debuggeren bruges i forbindelse med udvikling af software til TMS320C26. Den gør det muligt at debugge den benyttede software direkte på processoren. Dette indbefatter mulighed for at overføre programmet til DSP'ens RAM og starte eksekveringen af programmet. Det er også muligt at få tilgang til RAM området for at udlæse eller substituere data.

Debuggeren

Debuggeren benytter ét interrupt (INT0) til reset af systemet samt I/O port 15, og har lagt beslag på adresseområdet til program fra 0_h-10FF$_h$, hvor det første kword er til ROM'en og de sidste 256words er RAM, som debuggeren bruger. Det betyder, at debuggeren ikke benytter intern- som ekstern-data RAM.

Angående RS232 (forbindelsen mellem debuggeren og PC'en) er denne konstrueret således, at der ikke benyttes den serielle kanal, der findes på TMS320C26, men det hele kører via de øverste 5 databen til processoren (D11-15). Derved er denne fri til codec. Selve den serielle forbindelse kommunikerer via følgende protokol:

- Baudhastighed på 19200baud.
- 1 startbit.
- 7 databit.
- 1 paritetsbit.
- 1 stop bit.

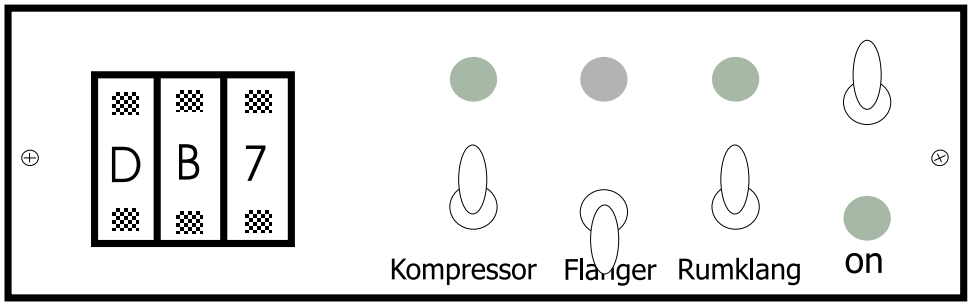
Ovenstående betyder, at den benyttede debugger er "næsten usynlig". Under eksekvering af program, benyttes debuggeren ikke, med mindre trap eller reset aktiveres (trap kun hvis interrupt er enabled).

Den EPROM som debuggeren er brændt i, er af typen M27C1024 og har følgende nøgledata:

- 64kword EPROM.
- 5V spændingsniveau.
- 35ns access tid.
- 16 bit adresserbar.

6.4 Brugerinterface

For at kunne indstille og selektere de forskellige DSP-effekter laves et brugerinterface, således at der kan vælges om guitarsignalet skal behandles af flangeren, rumklangen eller kompressoren. Desuden skal der være mulighed for justering af de forskellige effekter. Der benyttes en on/off switch til hver effekt, der aktiverer/deaktiverer effekten, og til at justere effekterne, benyttes en thumbleswitch til hver effekt. Dermed kan effekten justeres binært op og ned i 16 forskellige trin for hver thumbleswitch. Brugerinterfacet ses på figur 6.3



Figur 6.3: *Det benyttede brugerinterface.*

Som det ses på figur 6.3 består brugerinterfacet af 4 kontakter og 3 thumbleswitches. Disse kontakter og switches sætter et register, der latches ud på databussen, når registret chipselecteres. Dette register hentes ind i hovedprogrammet til DSP'en, der arbejder når interrupt proceduren ikke er i gang (dvs. når effekterne er processeret). Derved tager brugerinterfacet ikke unødig tid for selve den digitale signalbehandling af guitarsignalet.

6.4.1 Valg af effekt

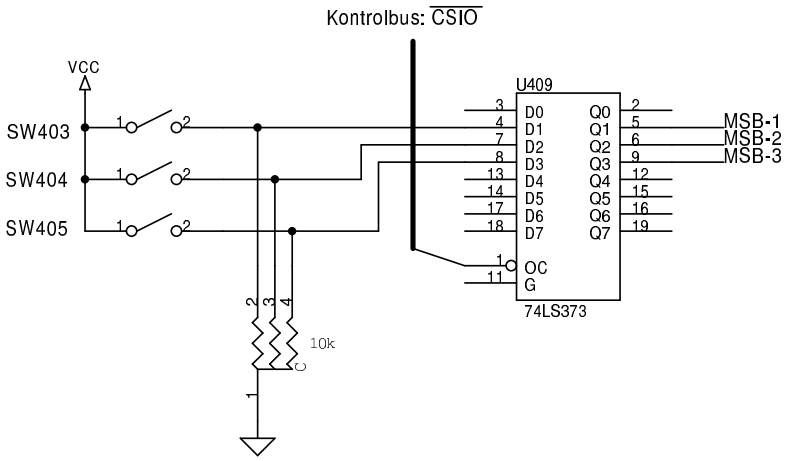
Brugeren vælger hvilke effekter, der skal afvikles ved at sætte on/off switchene. Den valgte indstilling sætter inputtet på brugerinterfaceregisteret, på de øverste 3 registeradresser (d1 til d3). Inputtene er forsynet med pull-down modstande, der trækker d1 til d3 lavt, når de ikke er valgt. Det vælges at lade de 3 effektvalg repræsenteres som: D12 til D14. Switchenes betydning ses i tabel 6.1.

Effekt	D14	D13	D12
Reverb	1	0	0
Flanger	0	1	0
Kompressor	0	0	1

Tabel 6.1: De 3 on/off switches betydning for brugerinterfaceregistret.

On/off switchene forbindes som vist på figur 6.4 på den følgende side.

Når latchen chipselectes, sendes indholdet ud på databussen, og DSP'en læser switchenes indstilling. Når chipselect fjernes, tri-states latchens output.



Figur 6.4: *Diagram over on/off switche til valg af DSP effekter.*

Juster effekt nummer	Latches til databussen:	Latch adresse	Latch nummer
1	D11 - D8	d5 - d8	1
2	D7 - D4	d1 - d4	2
3	D3 - D0	d5 - d8	2

Tabel 6.2: Brugerinterfaceregistrets forbindelse med databussen.

Som latch vælges 74HC373, der indeholder en 8-bit latch der er tri-statet, når der ikke modtages chipselect.

6.4.2 Justering af effekt

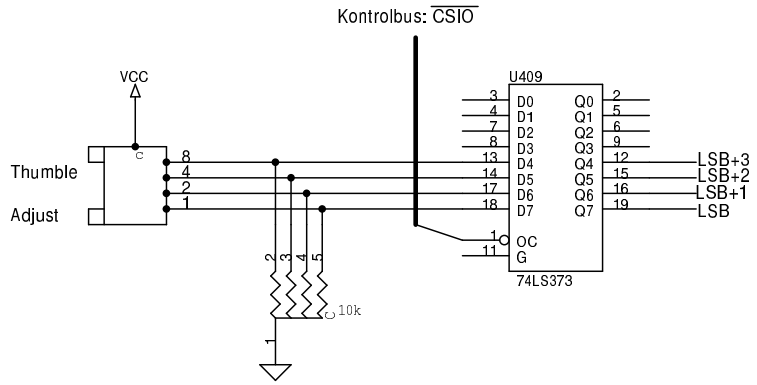
Til koefficientjustering af den valgte effekt, benyttes thumbleswitches. Der benyttes en thumbleswitch til hver effekt. Thumbleswitch'en har 16 instillinger, der vælges ved tryk på mekanisk switch. Indstillingen udlæses binært i 4 bit. Outputtet sendes ind på brugerinterfaceregistret, der latcher dem ud på databussen, ved $\overline{CSIO}$.
I tabel 6.2 ses en oversigt over registrets forbindelse med databussen:

Som det ses af figur 6.5 på modstående side er alle latches koblet med pull-down modstande.

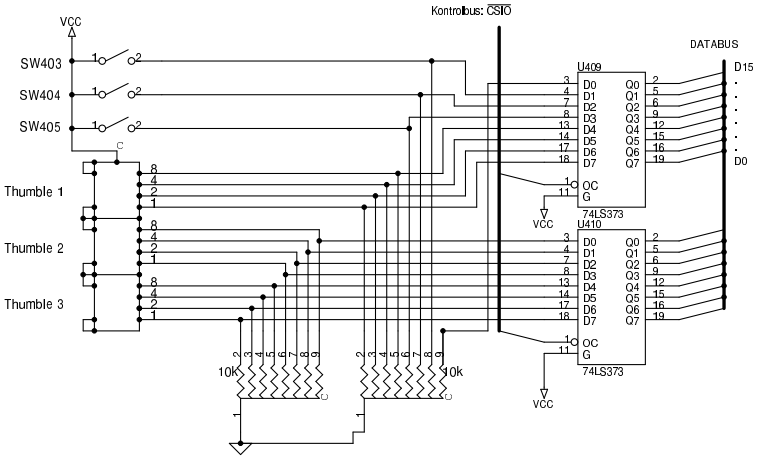
På figur 6.6 på næste side ses det samlede brugerinterface.

6.5 Kobling af systemet

Systemet, bestående af 4 x 32kB RAM, debugger og brugerinterface, skal kobles sammen således, at de tilsammen udgør et komplet DSP-system. Før sammenkoblingen ses der på timingen i systemet.



Figur 6.5: *Diagram over indstillingsmulighederne.*



Figur 6.6: *Diagram over det samlede brugerinterface.*

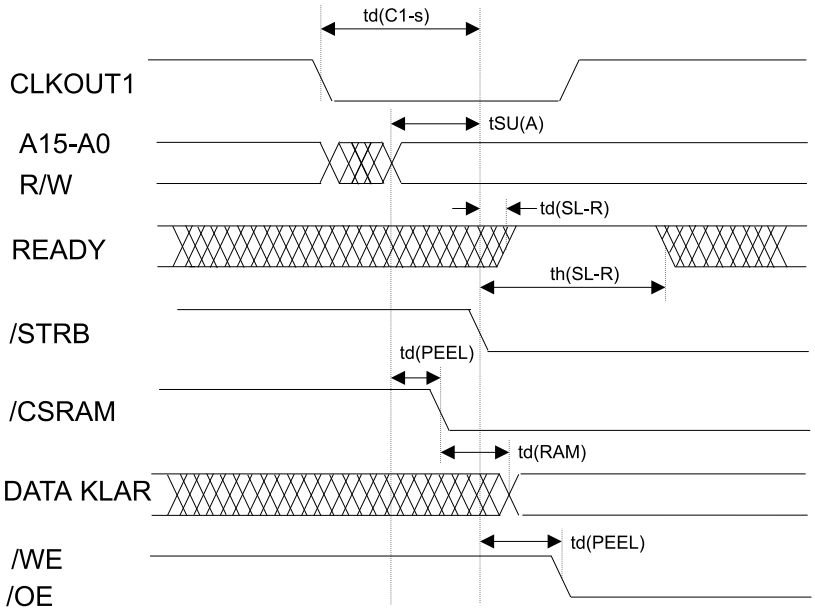
6.5.1 Timing

Her beskrives timingen for TMS320C26. Den tager udgangspunkt i værdien Q som er 1/4 af clocktiden (25ns v. 40MHz clock). Timingsdiagrammet ses på figur 6.7 og de specifikke tider til timingsdiagrammet ses i tabel 6.3.

Symbol	Forklaring	Tid
$t_{d(C1-S)}$	clk lav til $\overline{STRB}$ lav	Q±6ns =19→31ns
$t_{SU(A)}$	ADR klar til $\overline{STRB}$ lav	Q-12ns = 13ns
$t_{d(SL-R)}$	READY klar efter $\overline{STRB}$ lav	Q-22ns = 3ns
$t_{h(SL-R)}$	holdetid for READY efter $\overline{STRB}$ lav	Q+3ns = 28ns
$t_{d(PEEL)}$	tidsdelay gennem PEEL-kreds	10ns
$t_{d(RAM)}$	fra $\overline{CS}$ lav til data klar	15ns
$t_{d(EPROM)}$	fra $\overline{CS}$ lav til data klar	35ns

Tabel 6.3: Tidsangivelser for timingen af TMS320C26.

Det forudsættes i timingsdiagrammet, at der er indkoblet en 10ns PEEL kreds som adres-



Figur 6.7: *Timingsdiagram over DSP'en.*

sedekoder. Det ses, at for at undgå wait states skal de tilkoblede enheder have en adresseringstid på under 31ns ($t_{SU(A)} + t_{h(SL-R)} - t_{d(PEEL)}$). Det betyder, at de benyttede RAM kredse og brugerinterfacet ikke benytter waitstates, men da access tiden for EPROM'en er 35ns, betyder det, at det er nødvendigt med et waitstate, når der selekteres ROM.

6.5.2 Adressedekodning

Adressedekodning tager sig af samspillet mellem ydre enheder og DSP'en. Tabel 6.4 viser de eksterne enheders adresseområder.

Hardware	Startadresse	Slutadresse
EPROM	0000h	0FFFh
RAM1	1000h	7FFFh
RAM2	8000h	FFFFh
I/O	Lige adr. på IO-port	

Tabel 6.4: Adresseområdet.

Der benyttes en PEEL 18CV8 kreds til at styre adressedekodningen. Den har 10 input og 8 input/output (kan vælges softwaremæssigt).

For at kunne selectere det rigtige hardware, opstilles en række boolske udtryk, der beskriver hvornår de forskellige kredse skal chipselectes. Foruden de 4 chipselects (EPROM, RAM1, RAM2 og I/O), skal adressedekoderen lave $\overline{OE}$, $\overline{WE}$ og $READY$. $\overline{OE}$ (output enable - aktiv lav) bruges, når der skal læses fra eksterne enheder, $\overline{WE}$ (write enable - aktiv lav) bruges, når der skal skrives til eksterne enheder og $READY$ (klar - aktiv høj) bruges

til at generere wait-states ved skrivning til langsomme eksterne enheder (EPROM).
Som inputs vælges de højeste adresseben, samt $R/\overline{W}$, $\overline{MSC}$ og $\overline{STRB}$. For at vælge antallet af nødvendige adresseben, opstilles en tabel, hvor det er muligt at se, hvordan adressebenenes bit ændres i de ønskede adresseområder:

Hardware	A15 A14	adresse i hex
EPROM	0000 0000b	0000h
	0000 1111b	0FFFh
RAM1	0001 0000b	1000h
	0111 1111b	7FFFh
RAM2	1000 0000b	8000h
	1111 1111b	FFFFh

Tabel 6.5: Adressebenenes ændring ved de ønskede adresseområder.

Ud fra tabellen ses, at det er nødvendigt at medtage de 4 mest betydende bit på adressebussen, for at kunne skelne RAM1's slutadresse fra RAM2's startadresse. Det betyder, at A15 til A12 benyttes til input til adressedekoderen. I/O chipselectes via A0 og $\overline{IS}$ (IO select). Der opstilles boolske ligninger til de forskellige output. Den færdige PEEL kode findes på CDROM'en under PEEL.

6.5.3 Sammensætning med codec

For at sammensætte codec med resten af DSP-systemet, ses der på hvilket output codec sender ud. Denne er koblet i SM4 (Serial Mode 4), hvor der sendes 32 bit pakker. Disse pakker sendes og modtages med codec'ens clockfrekvens. Der sendes et synkroniseringsbit ud for hver 32 bit. Output fra codec (ADC) sættes til processorens DR ben, og input til codec (DAC) sættes til DX benet. Clocken fra codec sættes på CLKR og CLKX som er sende og modtage clockfrekvensen for den serielle kommunikation. Synkroniseringsbittet sættes på processorens FSR og FSX som er "frame synchronization". Derved sendes og modtages der fra codec med fast clockfrekvens, og da der kun kommer et synkroniseringsbit for hver 32 bit betyder det, at de sidste 16 bit ikke vil blive registreret. Derved benyttes kun 1 kanal (den venstre).

Til sidst skal det nævnes at den benyttede clock er en standard 40MHz clock, der forbindes til CLKIN. Det samlede diagram for hardware-opkoblingen kan ses i appendiks V på side 181.

KAPITEL 7

Software

Dette kapitel vil omhandle opbygningen af den benyttede software. Denne opbygning indbefatter en generel opbygning af den benyttede software samt muligheden for at justere lydeffekternes parametre vha. brugerinterfacet. Ud over dette vil det benyttede udviklingsværktøj fra Texas Instruments kort blive beskrevet. Til sidst vil det benyttede adresserum blive fastlagt.

7.1 Udviklingssoftware til DSP

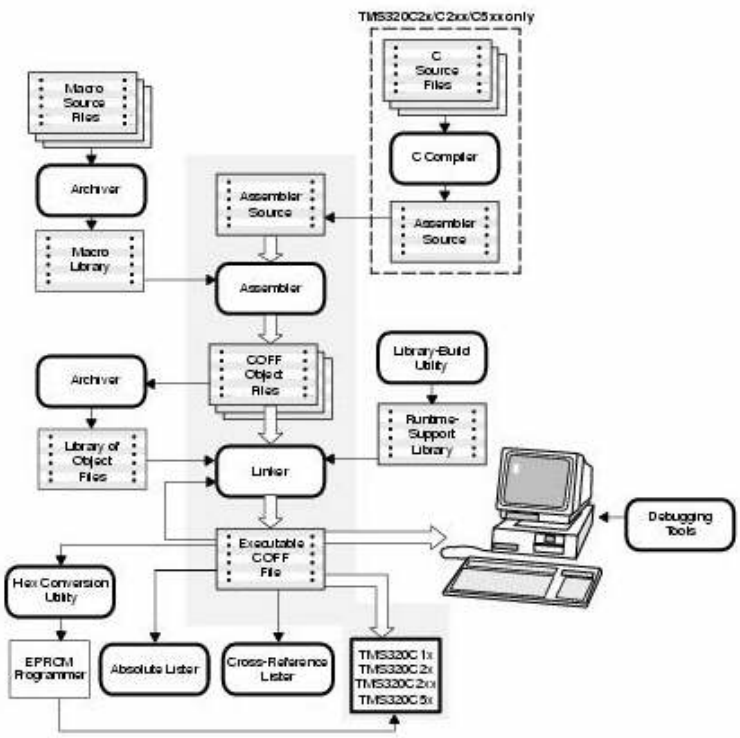
Til at programmere DSP-processoren bruges udviklingsværktøjet fra Texas Instruments. Dette værktøj indeholder både C- og assembler cross compiler og linkerværktøj. Herunder beskrives compileringsværktøjet, pånær C-compileren, da denne ikke benyttes.

7.2 Filer til kompilering

De filer som følger med til TMS320C26, for at kompilere et assembler program, er angivet i tabel 7.1:

Filer	Input-fil(er)	Output-fil(er)
dspa	.asm eller .axx	.ovj evt. .lst
dsplnk	.obj .cmd	.out .map
dspabs	.out	.abs
dsphex	.cmd .out .map	.hex

Tabel 7.1: *Tabel over filer brugt til assemblering og filer de laver.*



Figur 7.1: *Oversigt over forløbet for kompilering af en assemblerfil.*

7.2.1 dspa

Som det ses på figur 7.1 startes der med en assembler fil. Her benyttes programmet *dspa* der skal have at vide at assemblerprogrammet er til en DSP af typen C26. Dette gøres ved at sætte "-v26" efter. Endvidere kan der vælges, at outputtet skal lave en "list" fil. Denne fil kan, efter eksekvering af *dspa*, benyttes til at efterse, hvor i programmet der var fejl eller advarsler.

7.2.2 dsplnk

Næste led er at sammensætte ("linke") .obj filen med en .cmd (command) fil. Denne .cmd fil bestemmer hvilke RAM områder der skal anvendes og deres længde. Et eksempel på en .cmd fil:

```
test.obj            /*input file*/
-o test.out         /*executable output file*/
-m test.map         /* map file */

MEMORY
{
    PAGE 0 :            /* PROGRAM MEMORY fra 01000h - 0FFFFh*/
         INTS       : origin = 01052h , length  =  0Ch
         Prog_RAM   : origin = 01100h , length  =  05000h
         KOEFS      : origin = 07000h , length  =  04000h


    PAGE 1 :            /*DATA MEMORY*/
```

```
        Regs      : origin =    0h  , length  =     06h
        Block_B2  : origin =  060h  , length  =    020h
        Int_RAM   : origin = 0200h  , length  =   0600h
        Ext_RAM   : origin = 01100h , length  =  0EF00h
}

SECTIONS
{
    INT_VEC        :  > INTS       PAGE 0
    .text          :  > Prog_RAM   PAGE 0
    .data          :  > Ext_RAM    PAGE 1
    .bss           :  > Prog_RAM   PAGE 0
    KOEF           :  > KOEFS      PAGE 0
}
```

Heraf ses f.eks. at koefficienter (KOEFS) til flangeren, ligger fra adresse 2100_h og fylder $DF00_h$ i program RAM. Endvidere ses der at input filen er test.obj. Output fra dsplnk bliver: .out og .map. I .map filen kan der ses, hvad der kommer til at ligge i RAM og hvad rækkefølgen er. Hvis der ønskes at skulle benytte simulatoren til at debugge med, tilføjes -v0 ved eksekvering.

7.2.3 dspabs

Hvis en eller flere assembler filer skal samles, sætter *dspabs* dem sammen i en samlet .lst (list) fil når der igen bliver brugt kommandoen *dspa*. Den benytter sig af .out som var output fra *dsplnk*.

7.2.4 dsphex

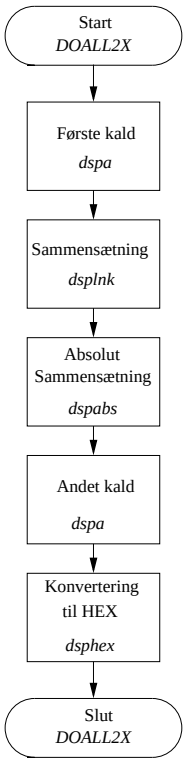
For at bruge *dsphex*, som er det sidste led i kompileringen, bruges der en command (.cmd) fil og en .map fil. I .map skrives der hvilket format .hex filen (output) skal have. Der vælges i hex.map at formatet skal være Intel. I command filen gives der besked til debuggeren om, hvilket format den skal forvente at modtage.

7.3 Den samlede kompilering

For at gøre det nemmere at kompilere en assembler fil, laves der en fil *DOALL2X.bat*, der kalder alle de filer og kommandoer der skal bruges, i den rækkefølge som angivet på figur 7.2 på næste side.

7.4 Programopbygning

Her fastlægges programopbygningen. Denne er delt op i en initialisering, et hovedprogram og en interrupt rutine. Disse er beskrevet i det følgende.



Figur 7.2: *Flow over kompilering med DOALL2X.*

7.4.1 Initialisering

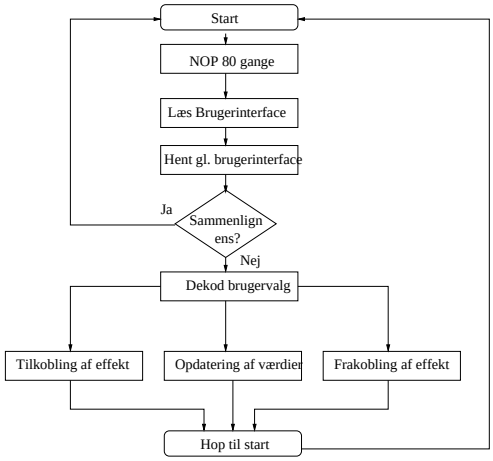
Denne del af programmet sørger for at alle variable bliver initialiseret. Her nulstilles også alle ringbufferne.

7.4.2 Hovedprogrammet

Hovedprogrammet har to funktioner: At opdatere koefficienterne til de forskellige effekter samt at tilslutte/frakoble de forskellige effekter. Dette skal ske inden for 1 sekund. Hovedprogrammet læser brugerinterfaceregistret og sammenligner det med det tidligere. Er de ens, foretages der ikke mere. Er der sket en ændring, findes denne og en effekt tilsluttes/frakobles eller parametre ændres.

For at undgå at smide interne registre (f.eks. auxillery registre) ud i RAM, hver gang interruptet bliver kaldt, disables interrupt hver gang, der bruges interne registre. I selve hovedprogrammet indlægges en del NOP instruktioner, der har til formål, at give plads til at interruptet kan afvikles. Opdatering af koeffiecienter til de forskellige effekter sker som vist på flowchartet på figur 7.3 på den følgende side.

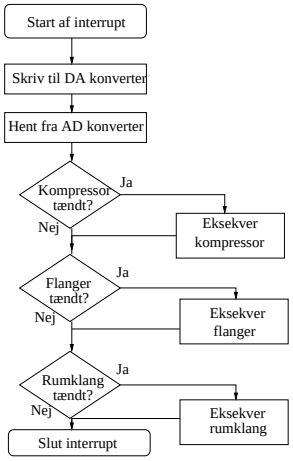
Opdateringen af de forskellige effekters koefficienter foretages ved at brugerinterfaceregistret læses og dekodes, og værdien mellem 0-F_h bruges til at opdatere de nye koefficienter. Måden de enkelte enheders opdatering sker på beskrives i de respektive afsnit.



Figur 7.3: *Flowchart for opdateringen af koefficienterne.*

7.4.3 Interrupt service rutinen

I projektet bruges det interrupt, der genereres af serielporten, til at processere guitareffekterne. Måden dette interrupt er bygget op på er vist på figur 7.4.



Figur 7.4: *Interrupt service rutinen.*

Som det ses af figur 7.4 skrives der til DA konverteren i starten af interruptet. Dette sker for at sikre sig at skrivningen til DA konverteren foretages på samme tid hver sample. Skulle der skrives til DA konverteren i slutningen af interruptet, ville det afhænge af hvor mange effekter, der var tilsluttet. Samtidig kan de enkelte effekter også variere i tid (ved f.eks. nulstilning af buffere). Efter skrivning til DA konverter hentes det analoge signal fra AD konverteren. Denne lægges ind i et input register. Derefter undersøges der først om compressoren er slået til, hvorefter den "hopper" til eksekveringen af kompressoren, hvis dette er tilfældet. På samme måde undersøges også de resterende effekter.

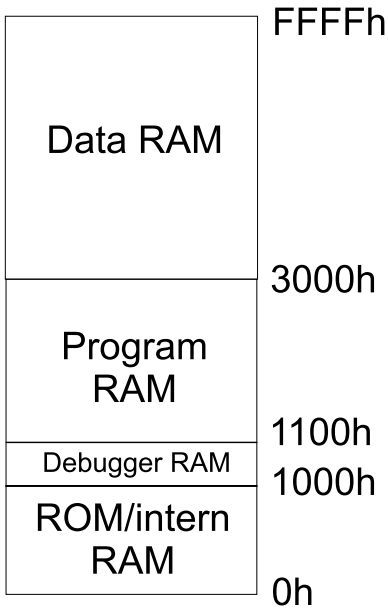
Hver effekt henter deres input i inputregistret og lægger deres output i inputregistret efter endt signalbehandling. Dette input register skrives så til DA konverter, når næste interrupt service rutine går igang.

Da den benyttede DSP har 10 millioner instruktioner pr. sekund (MIPS) skal interrupt service rutinen afvikles indenfor:

$$IRQ = \frac{MIPS}{f_S} = \frac{10000000}{22050} = 453,5 \approx 450 instruktioner \tag{7.1}$$

7.5 Opdeling af adresserum

For at udnytte processoren mest muligt, og for at sikre mod overlap i adresseringen, opdeles adresseområdet, således at der er fastelagte grænser mellem de forskellige effekter og mellem program og data. Adresseområdet er opdelt i et program område og et dataområde. Disse er vist på figur 7.5.



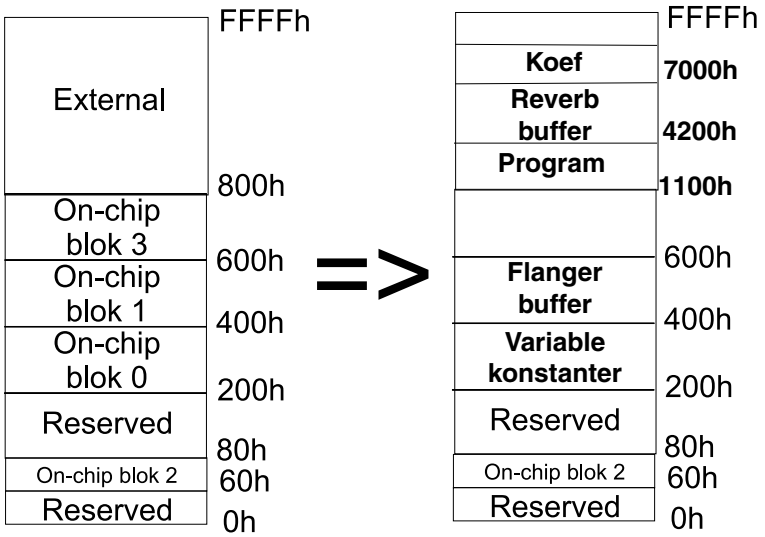
Figur 7.5: *Adresseområdet for TMS320C26.*

Som det ses på figur 7.5 og som det kan ses udfra adressedekodningen (afsnit 6.5.2 på side 48) skelnes der ikke mellem program og data RAM. Selve RAM'en ligger fra 1000_h og til $FFFF_h$. Debuggeren skal bruge de første 256words fra 1000_h, hvorefter RAM'en er fri til program og data (fra 1100_h). Det er valgt, at lade programmet starte fra denne adresse (1100_h). Programmet vurderes til ikke at fylde mere end 1900_h (6,4kword). Derfor lægges data RAM fra adresse 3000_h (1100_h+1900_h).

7.5.1 Intern RAM

DSP-processoren har 1,5kword intern RAM (se evt. afsnit 6.1 på side 41) fordelt på 3 blokke ($B0 - B2$). Denne RAM kan fungere både som data- og program RAM. Det er valgt at benytte CONF0, hvilket gør at alt intern RAM bruges til data RAM. Ved CONF0 befinder den interne RAM sig fra 200_h-$07FF_h$. For at udnytte dette RAM, vælges det at lægge alle konstanter i intern RAM. Disse lægges på side 4 (fra adresse 200_h).

Ud over dette vælges det at lægge buffere og variable fra kompressoren og flangeren i intern RAM. Dette gøres, da der i kapitlerne omhandlende disse effekter, kan ses, at der ikke bruges mere RAM end, der kan ligge i intern RAM. Selve opdelingen af den interne RAM ses på figur 7.6.



Figur 7.6: *Den interne RAM opdeling.*

Det betyder, at effekten rumklang lægges fra adresse 3000_h og opefter.

KAPITEL 8

Kompressor

Dette kapitel indeholder en beskrivelse af kompressoren, der er implementeret i systemet. Under dokumentationen er det matematiske og teoretiske princip beskrevet, og den implementerede algoritme er forklaret.

8.1 Krav til kompressoren

I kravsspecifikationen blev der opstillet følgende krav til kompressoren:

- Mulighed for indstilling af threshold-værdier mellem 0 og -30dB.
- Forskellige attack-/releasetid for høje/lave frekvenser.
- Mulighed for valg af forskellige kompressionsrater.
- Generelle krav fra kravspecifikationen.

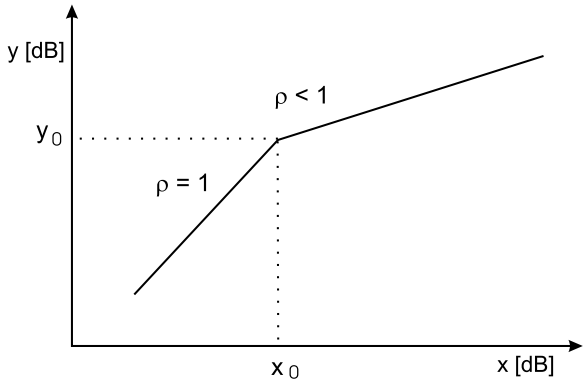
8.2 Princippet i kompressoren

I det følgende vil det matematiske princip bag kompressionen blive beskrevet på baggrund af kilden [10].

Princippet i en kompressor er, at dynamikområdet skal mindskes. Dette gøres ved at detektere, om et signals værdi er over eller under en bestemt værdi, kaldet threshold. Der findes i hovedtræk to former for kompressorer. Enten bliver signalet komprimeret, hvis det er *under* threshold eller også komprimeres det, hvis det er *over*. Det betyder hhv. at

svage lydtryk forstærkes eller at kraftige lydtryk dæmpes. Den kompressor, som ønskes realiseret i dette projekt, skal dæmpe lydtrykniveauer *over* threshold. Fordelen ved denne kompressor er, at eventuel støj ikke forstærkes op, og kraftige pulser dæmpes.

Generelt gælder om den slags kompressorer, at deres input-/outputrelation er som vist på figur 8.1.



Figur 8.1: *Relation mellem output og input for en kompressor.*

De på figuren viste symboler har følgende sammenhæng:

$$y = y_0 + 20 \cdot \rho \cdot log\left(\frac{x}{x_0}\right),\ 0 < x_0 < x \tag{8.1}$$

Hvor ρ er kompressionsraten, x er inputsignalet og y outputsignalet. Det ses udfra den ovenstående ligning, at kompressionsraten har indflydelse på, hvor kraftigt inputsignalet skal dæmpes. Hvis der forekommer en værdi af ρ på mere end én, ses det udfra formel 8.1, at der i stedet foretages en ekspansion af outputværdien. For kompressorer af denne type gælder, at $\frac{1}{4} < \rho < \frac{1}{2}$ [10]. Det skal understreges, at formlen for input-/outputrelationen kun gælder, når x_0 er mindre end inputtet, x.

Det er derfor nødvendigt, at have en algoritme til at bestemme om indgangssignalet er over threshold. Dette kan enten gøres med en peak-detektor, en envelope-detektor eller en RMS-detektor. En sådan detector beregner et såkaldt styresignal, og dette anvendes til at bestemme om indgangssignalet skal komprimeres.

8.3 Matlab simulering

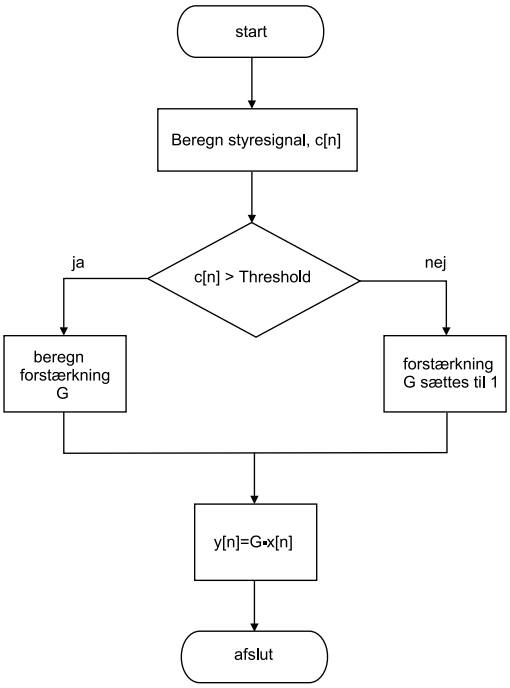
For at eftervise det valgte princip for kompressoren, er der lavet et program i MATLAB, efter dette princip. Inputsignalet er en cosinusfunktion:

$$x[n] = A \cdot \cos(\omega_0 \cdot n) \tag{8.2}$$

Signalet løber således over 600 samples, inddelt i 3 lige store udsnit med varierende amplituder; $A = 2$, $A = 4$ og $A = 0{,}5$. På denne måde har det været muligt at foretage tests med forskellige værdier for threshold og kompressionsrate. I appendiks C på side 111 ses resultatet af dette eksempel, samt for et langt mere varierende signal. I det følgende er algoritmen til kompressoren beskrevet.

8.4 Design af kompressor

På baggrund af det i foregående afsnit beskrevne princip for kompressoren, kan der udarbejdes et flowdiagram for en kompressor-algoritme. Dette ses på figur 8.2.



Figur 8.2: *Flowdiagram for et kompressor program.*

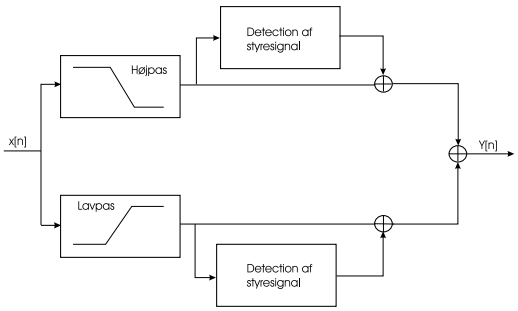
Det er ønskeligt at have forskellige parametre for attack-/releasetid for dybe/høje toner i kompressoren. Dette skyldes, at der derved opstår forskellige lyttemæssige effekter. Det er derfor ofte set ved kompressorer, at frekvensområdet er opdelt i bånd, hvor releasetid og attacktid kan indstilles forskelligt for båndene [8]. Det er på dette grundlag valgt, at opdele kompressoren i flere bånd.

Da det matematiske princip for kompressoren omfatter negative eksponenter mellem 0 og -1, kan kompressoren ikke implementeres direkte og eksakt. Derfor vælges det at arbejde med to forskellige kompressorer med hhv. tabellagte værdier for forstærkningen og stepping af forstærkningskoefficienter. Algoritmen til kompressoren omfatter i hovedtræk følgende faser:

Der foretages følgende valg mht. implementering af programmet for kompressoren:

- Frekvensopdeling af kompressionen i to bånd.
- Detection af styresignal vha. middelværdi-detektor.
- Tabellægning af forstærkningskoefficienter eller beregning vha. stepping.

Det betyder, at blokdiagrammet for kompressoren kommer til at se ud som vist på figur 8.3 på den følgende side:



Figur 8.3: *Blokdiagram for den valgte kompressor.*

8.4.1 Opdeling i frekvensbånd

Frekvensområdet, som kompressoren skal arbejde med, opdeles i et lavpas- og et højpasbånd. Dette gøres for at kunne give disse frekvensbånd to forskellige attack-/releasetider. Der ønskes en flankestejlhed på 40dB/dekade, og indenfor pasbåndet skal filtret have en flad karakteristik. På denne baggrund vælges at bruge 2. ordens filtre. Delefrekvensen sættes til 500Hz, da frekvenser under denne opfattes som bas, og frekvenser over som presence/diskant [6]. For et 2. ordens lavpasfilter er overføringsfunktionen:

$$H_{LP}(s) = \frac{(\Omega_0)^2}{(s+\Omega_0)^2} = \frac{\Omega_0^2}{s^2 + 2 \cdot \Omega_0 \cdot s + \Omega_0^2} \tag{8.3}$$

For et 2. ordens højpasfilter er overføringsfunktionen iflg. [9]:

$$H_{HP}(s) = \frac{s^2}{(s+\Omega_0)^2} = \frac{s^2}{s^2 + 2 \cdot \Omega_0 \cdot s + \Omega_0^2} \tag{8.4}$$

For at overføre den analoge frekvens, Ω_0, til en digital frekvens, ω_0, benyttes bilineær transformation, da den giver digitale filtre den samme orden som de analoge. Når bilineær transformation anvendes, bliver den analoge frekvens afbildet indenfor enhedscirklen. Transformationen udføres ved at substituere s med z således:

$$H(z) = H(s)\big|_{s=\frac{2}{T_d}\cdot\frac{1-z^{-1}}{1+z^{-1}}} \tag{8.5}$$

hvor T_d er samplingsintervallet, altså $\frac{1}{f_s}$. I det følgende udledes den bilineære transformation for lavpas- og højpasfiltret.

$$H_{LP}(z) = \frac{\Omega_0^2}{(\Omega_0+s)^2}\big|_{s=\frac{2}{T_d}\cdot\frac{1-z^{-1}}{1+z^{-1}}} \tag{8.6}$$

$\Downarrow$

$$H_{LP}(z) = \frac{(T_d^2\cdot\Omega_0^2)+2\cdot(T_d^2\cdot\Omega_0^2)\cdot z^{-1}+(T_d^2\cdot\Omega_0^2)\cdot z^{-2}}{(T_d^2\cdot\Omega_0^2+4\cdot T_d\cdot\Omega_0+4)+(T_d^2\cdot\Omega_0^2\cdot 2-8)\cdot z^{-1}+(T_d^2\cdot\Omega_0^2-4\cdot T_d\cdot\Omega_0+4)\cdot z^{-2}} \tag{8.7}$$

$\Downarrow$

$$H_{LP}(z) = \frac{(z^{-1}+1)^2\cdot T_d^2\cdot\Omega_0^2}{(z^{-1}\cdot(T_d\cdot\Omega_0-2)+T_d\cdot\Omega_0+2)^2} \tag{8.8}$$

For højpasfiltret foretages samme transformation.

$$H_{HP}(z) = \frac{s^2}{(\Omega_0+s)^2}\Big|_{s=\frac{2}{T_d}\cdot\frac{1-z^{-1}}{1+z^{-1}}} \tag{8.9}$$

$$\Downarrow$$

$$H_{HP}(z) = \frac{4\cdot(z^{-1}-1)^2}{(T_d^2\cdot\Omega_0^2+4\cdot T_d\cdot\Omega_0+4)+(T_d^2\cdot\Omega_0^2\cdot 2-8)\cdot z^{-1}+(T_d^2\cdot\Omega_0^2-4\cdot T_d\cdot\Omega_0+4)\cdot z^{-2}} \tag{8.10}$$

$$\Downarrow$$

$$H_{HP}(z) = \frac{4\cdot(z^{-1}-1)^2}{(z^{-1}\cdot(T_d\cdot\Omega_0-2)+T_d\cdot\Omega_0+2)^2} \tag{8.11}$$

For et 2. ordens system er overføringsfunktionen:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1\cdot z^{-1} + b_2\cdot z^{-2}}{a_0 - a_1\cdot z^{-1} + a_2\cdot z^{-2}} \tag{8.12}$$

Udfra denne sammenhæng kan filtrets koefficienter beregnes og indsættes i differensligningen. Differensligningen findes ved at foretage en invers z-transformation af $H(z)$:

$$y[n] = b_0\cdot x[n] + b_1\cdot x[n-1] + b_2\cdot x[n-2] + a_1\cdot y[n-1] - a_2\cdot y[n-2] \tag{8.13}$$

Som tidligere beskrevet afbildes den analoge frekvensakse indenfor enhedscirklen, hvilket i teorien vil sige, at en uendelig stor frekvens i tidsdomænet svarer til π på enhedscirklen. I praksis er frekvensen, som svarer til π, givet ved: $\frac{f_s}{2}$. Det resulterer i, at amplituderne for store og små frekvenser afbildes ulineært. Derfor prewarpes de analoge frekvenser. Det er en måde at kompensere for den frekvensforvrængning der forekommer, under udføring af bilineær transformation. Dette gøres ved at substituere Ω_0 således:

$$\Omega_{ny} = \frac{2}{T_d}\cdot tan\left(\frac{\omega_0}{2}\right) \tag{8.14}$$

hvor den digitale frekvens, ω_0 er givet ved:

$$\omega_0 = \frac{\Omega_0}{f_s} = \frac{2\cdot\pi\cdot 500Hz}{22050Hz} = 0,1425rad \tag{8.15}$$

Derved fås:

$$\Omega_{ny} = 2\cdot 22050Hz\cdot tan\left(\frac{0,1425rad}{2}\right) = 3147,25rad = 500,85Hz \tag{8.16}$$

Det vil sige, at de 500Hz bliver til 500,85Hz på grund af forvrængning. Differensligningens koefficienter findes ved at indsætte de transfomerede og prewarpede koefficienter for hhv. højpas- og lavpasfiltret, normeret mht. $a_0 = 1$.

Derved findes følgende udtryk for værdierne til differensligningen:

For lavpas:

$$b_0 = (T_d^2 \cdot \Omega_0^2) = 0,020$$
$$b_1 = (2 \cdot T_d^2 \cdot \Omega_0^2) = 0,041$$
$$b_2 = (T_d^2 \cdot \Omega_0^2) = 0,020$$
$$a_0 = (T_d^2 \cdot \Omega_0^2 + 4 \cdot T_d \cdot \Omega_0 + 4) = 4,591$$
$$a_1 = (2 \cdot T_d^2 \cdot \Omega_0^2 - 8) = -7,959$$
$$a_2 = (T_d^2 \cdot \Omega_0^2 + 4 - 4 \cdot T_d \cdot \Omega_0) = 3,449$$

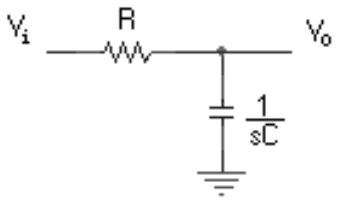
For højpas:

$$b_0 = 4$$
$$b_1 = -8$$
$$b_2 = 4$$
$$a_0 = (T_d^2 \cdot \Omega_0^2 + 4 \cdot T_d \cdot \Omega_0 + 4) = 4,591$$
$$a_1 = (2 \cdot T_d^2 \cdot \Omega_0^2 - 8) = -7,959$$
$$a_2 = (T_d^2 \cdot \Omega_0^2 + 4 - 4 \cdot T_d \cdot \Omega_0) = 3,449$$

8.4.2 Detektion af styresignal

I denne kompressor vælges at bruge en middelværdi-detektor, for vha af denne at kunne bestemme attack-/releasetiden.

Den valgte middelværdi-detector fungerer ved, at styresignalet beregnes vha. en differensligning, der virker som ensretter, efterfulgt af et lavpasfilter. Dette filter midler indgangssignalet og det foregående styresignal. Det vil sige, at en ændring i amplituden ikke udmunder i et tilsvarende ændret signal på udgangen. Dette skyldes, at styresignalet forholdsmæssigt, ikke ændrer sig så meget/hurtigt som indgangssignalet. Udledningen af differensligningen tager udgangspunkt i et 1. ordens lavpasfilter:



Figur 8.4: *1. ordens lavpasfilter.*

Det viste filter har en overføringsfunktion som vist nedenfor:

$$\frac{v_o}{v_i} = \frac{\frac{1}{s \cdot C}}{R + \frac{1}{s \cdot C}} = \frac{1}{1 + s \cdot C \cdot R} = \frac{1}{1 + \frac{s}{\Omega_0}} = \frac{\Omega_0}{s + \Omega_0} \tag{8.17}$$

En fremstilling af den digitale overføringsfunktion kan beregnes direkte ved at lave en "Backward Euler" transformation, hvor s substitueres med udtrykket i formel 8.18 :

$$s = \frac{1 - z^{-1}}{T_s} \tag{8.18}$$

Heraf følger overføringsfunktionen:

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)}\Big|_{s=\frac{1-z^{-1}}{T_s}} &= \frac{\omega_0}{\frac{1-z^{-1}}{T_s}+\omega_0} \\ = \frac{T_s \cdot \omega_0}{T_s \cdot \omega_0 + (1-z^{-1})} &= \frac{T_s \cdot \omega_0}{1+T_s \cdot \omega_0 - z^{-1}} \\ = \frac{\frac{T_s \cdot \omega_0}{1+T_s \cdot \omega_0}}{1-\frac{1}{1+T_s \cdot \omega_0} \cdot z^{-1}} & \end{aligned} \tag{8.19}$$

For at gøre beregningerne mere overskuelige, erstattes tælleren med a og koefficienten til z i nævneren erstattes med b.

$$\frac{Y(z)}{X(z)} = \frac{b}{1-a \cdot z^{-1}} \tag{8.20}$$

$$\Updownarrow$$

$$Y(z) \cdot (1 - a \cdot z^{-1}) = b \cdot X(z) \tag{8.21}$$

$$\Updownarrow$$

$$Y(z) = b \cdot X(z) + a \cdot Y(z) \cdot z^{-1} \tag{8.22}$$

$$\Updownarrow$$

$$y[n] = b \cdot x[n] + a \cdot y[n-1] \tag{8.23}$$

For differensligningen gælder tillige:

$$b + a = \frac{T_s \cdot \omega_0}{1 + T_s \cdot \omega_0} + \frac{1}{1 + T_s \cdot \omega_0} = 1 \tag{8.24}$$

Eftersom $a + b$ giver 1 kan differensligningen sammenlignes direkte med den i kilden [10] angivne differensligning. Det ses umiddelbart, at de to differensligninger er identiske:

$$c[n] = \lambda \cdot c[n-1] + (1 - \lambda) \cdot |x[n]| \tag{8.25}$$

Ensretningen af signalet består i at bruge absolut-værdier for inputtet. Ovenstående ligning anvendes til at beregne styresignalet.

Det ønskes at kunne vælge mellem to specifikke attack-/releasetider for styresignalet til hhv. lavpas- og højpasfiltret. Derfor drages en sammenhæng mellem det analoge filters opladningstid og differensligningens koefficienter. I det analoge filter er opladningstiden bestemt af RC-leddet (τ). Der gælder:

$$V(t) = V_\infty + (V_0 - V_\infty) \cdot e^{\frac{-t}{RC}} \tag{8.26}$$

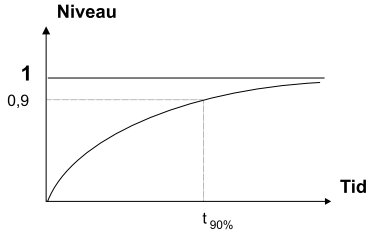
$$\Updownarrow$$

$$V(t) = V_\infty(1 - e^{\frac{-t}{\tau}}) \tag{8.27}$$

$$\Updownarrow$$

$$\tau = \frac{-t}{ln(1-\frac{V(t)}{V_\infty})} \tag{8.28}$$

hvor V_0 repræsenterer et indgangssignal, V(t) er spændingen til tiden t og V_∞ er den maksimale spænding der opnås over kondensatoren. Et typisk forhold mellem V_t og V_∞ er 90% [6], hvilket svarer til det procentvise niveau af indgangssignalet i forhold til udgangssignalet til tiden t (se figur 8.5. Det vil derfor blive praktiseret i designet af denne kompressor.



Figur 8.5: *Opladningskurve for kondensator.*

Der er af projektgruppen fastlagt en attack-/releasetid på 8ms for højpas- og 500ms for lavpasfiltret. På baggrund heraf kan filtrenes opladningskonstanter (τ) findes.

$$\tau_{højpas} = -\frac{t}{ln(1-\frac{V(t)}{V_\infty})} = -\frac{8ms}{ln(1-0{,}9)} \tag{8.29}$$

$$\Updownarrow$$

$$\tau_{højpas} = 3{,}47ms \tag{8.30}$$

På samme måde er det gældende for lavpasfiltret:

$$\tau_{lavpas} = -\frac{t}{ln(1-\frac{V(t)}{V_\infty})} = -\frac{500ms}{ln(1-0{,}9)} \tag{8.31}$$

$$\Updownarrow$$

$$\tau_{lavpas} = 217{,}15ms \tag{8.32}$$

Det betyder, at filterkoefficienten λ for hhv. højpas- og lavpasfilter kan findes:

$$\lambda_{højpas} = a_{højpas} = \frac{1}{1+T_s \cdot \omega_0} \tag{8.33}$$

$$\Updownarrow$$

$$\lambda_{højpas} = \frac{\tau_{højpas}}{T_s+\tau_{højpas}} = 0{,}9871 \quad (7E5A_h) \tag{8.34}$$

$$(1 - \lambda_{højpas}) = b_{højpas} = \frac{T_s \cdot \omega_0}{1 + T_s \cdot \omega_0} \tag{8.35}$$

$$\Updownarrow$$

$$(1 - \lambda_{højpas}) = \frac{\frac{1}{\tau_{højpas}}}{\frac{1}{T_s} + \frac{1}{\tau_{højpas}}} = 0,0129 \quad (01A5_h) \tag{8.36}$$

$$\lambda_{lavpas} = a_{lavpas} = \frac{1}{1 + T_s \cdot \omega_0} \tag{8.37}$$

$$\Updownarrow$$

$$\lambda_{lavpas} = \frac{\tau_{lavpas}}{T_s + \tau_{lavpas}} = 0,9998 \quad (7FF9_h) \tag{8.38}$$

$$(1 - \lambda_{lavpas}) = b_{lavpas} = \frac{T_s \cdot \omega_0}{1 + T_s \cdot \omega_0} \tag{8.39}$$

$$\Updownarrow$$

$$(1 - \lambda_{lavpas}) = \frac{\frac{1}{\tau_{lavpas}}}{\frac{1}{T_s} + \frac{1}{\tau_{lavpas}}} = 0,0002 \quad (0006_h) \tag{8.40}$$

Med disse oplysninger kan styresignalet, c[n], rekursivt beregnes, hvorved det kan afgøres om indgangssignalets størrelse skal komprimeres. Følgende sammenhæng gælder for forstærkningskoefficienten, Gain:

$$\text{Gain} = \begin{cases} 1 & for\ threshold \leq c[n] \\ (\frac{c[n]}{threshold})^{\rho - 1} & for\ c[n] > threshold \end{cases}$$

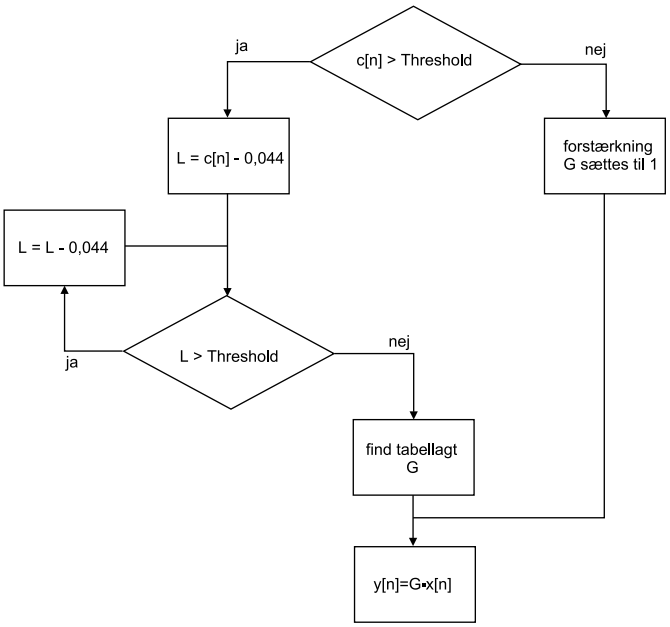
8.4.3 Beregning af forstærkning

Eftersom der ikke kan implementeres en eksakt algoritme til at bestemme forstærkningen, er der lavet forsøg med to forskellige metoder til beregning af forstærkningen.

Metode 1: Tabellægning af forstærkningskoefficienter

Threshold fastlægges for 0dB til 90% af maksimal. Det betyder at -30dB svarer til en forstærkningskoefficient på 0,028. Threshold og styresignalet $c[n]$, sammenlignes ved at subtrahere threshold fra $c[n]$. Intervallet for differencen mellem styresignal og threshold ligger i intervallet 0 til 0,872 (0,9-0,028), hvilket svarer til 0dB til -30dB. Dette er inddelt i 20 steps af størrelsen 0,044. Hvis differencen er mindre end eller lig med nul, er forstærkningen fastsat til én. Hvis differencen er større end 0, mindskes dette tal med en fastsat værdi på 0,044. Hvis denne værdi er lig med eller mindre end nul, ganges inputsignalet

på en tabellagt gain-værdi. Hvis ikke dette er tilfældet, trækkes 0,044 fra tallet igen, og inputsignalet multipliceres på en mindre gain-værdi. Således fortsætter det indtil differencen mellem styresignalet og threshold, med en vís tilnærmelse, er fundet. På figur 8.6 ses, hvorledes dette program afvikles.



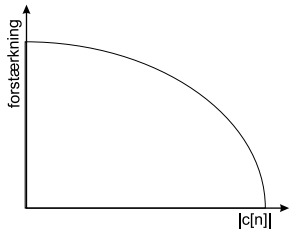
Figur 8.6: *Flowdiagram for metode 1 til beregning af forstærkningskoefficient.*

Forstærkningskoefficienterne er beregnet udfra et ønske om en eksponentiel aftagelse, således at signaler, hvis styresignal er tæt på threshold, forstærkes med en faktor lige under én. Hvis styresignalet er 30dB større end threshold, er forstærkningen meget lille. Som følge af opdelingen i intervaller, er det vigtigt, at forstærkningen ikke mindskes mere pr step, end at outputsignalet fortsætter med at vokse for tilsvarende større styresignaler. Udfra de valgte step på 0,044 kan det udregnes, at forstærkningen ikke må aftage med mere end $5,26\%$ i forhold til det forrige. Dette tal er fundet udfra formlen:

$$\frac{0,044 \cdot 20 - 0,044 \cdot 19}{0,044 \cdot 19} \cdot 100\% < x \tag{8.41}$$

hvor x er den maksimale procentvise ændring. Der ønskes derfor en forstærkning, hvor det procentvise fald pr. step, til at starte med, er meget lille, men ved sidste step bliver 5,26%. Altså skal grafen omtrent have udseende som på figur 8.7 på modstående side. Steppets størrelse (hen ad x-aksen) er 5% af værdien i punkt 20. Der findes 20 lige store steps fra origo hen til denne x-værdi i punkt nr 20.

Disse egenskaber findes ved et ellipseudsnit. For forskellige værdier af kompressionsraten ρ, vælges ellipser med forskellige radier på bredden. Stor radius medfører en lille kompression og lille radius medfører en stor kompressionsrate. Princippet til beregning af forstærkningskoefficienter er som følger: Der udvælges to steder på x-aksen, som ligger 5% fra hinanden i forhold til origo (inddeling i 20 intervaller). Herefter gælder, at det sted på ellipsen (x_0, y_0), hvor



Figur 8.7: *Funktion til at bestemme de tabellagte gain-værdier.*

$$\frac{y_0 - y_1}{y_0} = 5,26\% \tag{8.42}$$

forefindes den maksimale x-værdi. Intervallet på x-aksen fra origo til yderpunktet (x-max) inddeles i 20 punkter med lige store mellemrum, og hver af disse punkters *y-værdi* bruges til at skalere gain-faktoren med.

Princippet med at vælge ellipser med forskellige radier på bredden, kræver megen matematisk bearbejdning, hvis kompressionsraterne skal være fastsat til forudbestemte værdier. Derfor er det valgt, blot at bruge ellipser, hvor forholdet mellem højden og bredden af ellipsen varierer mellem 1 og 9.

Eftersom den anvendte algoritme for kompressoren beregner differencen mellem threshold og styresignalet og ikke forholdet, er der en vis afvigelse i den implementerede effekt og det matematiske princip. Der er dog kompenseret for dette ved at tabellægge forstærkningsværdien efter eksponentiel aftagelse.

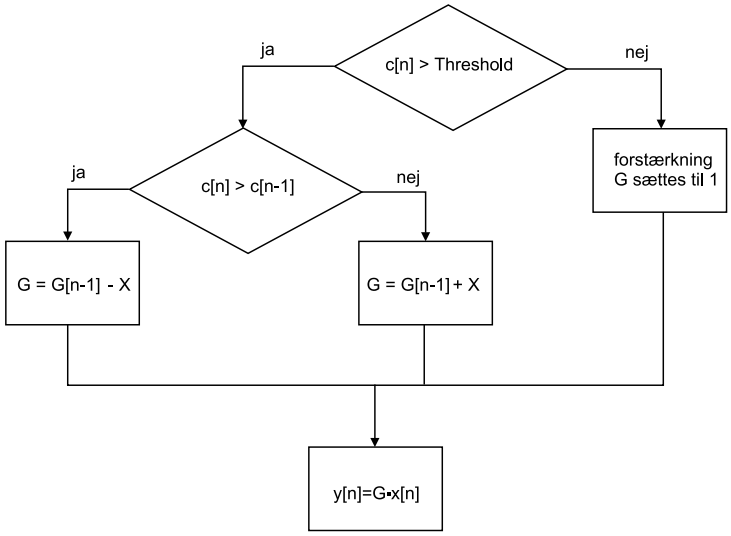
Metode 2: Forstærkningskoefficient steppes

Da metode 1 har 20 fastlagte forstærkningskoefficienter, resulterer det i pludselige spring for forstærkningen. Endvidere skal der subtraheres en værdi på 0,044, uanset hvilket indgangssignal der er tale om, bare det er over threshold. Det betyder, at der skal laves flere beregninger end nødvendigt, fordi styresignalet aldrig ændrer sig væsentligt fra sample til sample. Derfor er det, mht. effektivitet af programmet, mere hensigtsmæssigt at sammenholde det nuværende styresignals størrelse med det forrige, og derudfra vurdere om forstærkningskoefficienten skal mindskes eller forøges med et meget lille step.

Fordelen ved dette princip til beregning af forstærkningskoefficienter er, at der ikke skal subtraheres unødvendigt mange step og at forstærkningskoefficienterne bliver inddelt i finere intervaller.

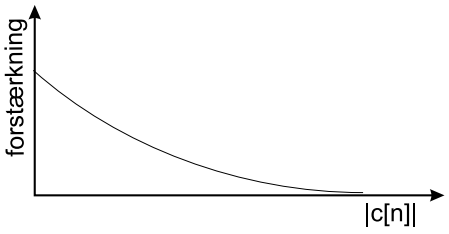
Princippet i denne algoritme til forstærkningskoefficientberegning er vist på figur 8.8 på næste side.

I programmet er det step (x), som forstærkningen enten forøges eller mindskes med, sat til forskellige værdier mellem 0,1% og 1%. Disse er valgt udfra MATLAB simuleringen, som forefindes i appendiks C på side 111.



Figur 8.8: *Flowdiagram for metode 2 til beregning af forstærkningskoefficient.*

Forstærkningen som funktion af inputtet, ser derfor ud som vist på figur 8.9 på næste side.



Figur 8.9: *Forstærkning som funktion af c[n], når c[n] er større end threshold.*

Som det fremgår af figuren, er der ifølge det matematiske princip, ikke tale om en "rigtig kompressor".

Valg af metode

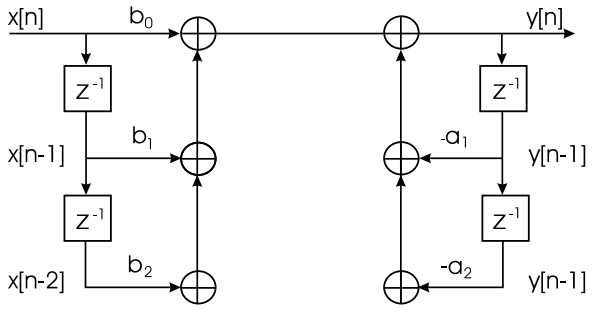
Det blev fra projektgruppens side valgt at bruge metode 2 til kompressoren. Dette valg blev truffet på baggrund af flere faktorer: For det første gav den det bedste lydmæssige resultat i MATLAB-simuleringen, og for det andet medfører det en hurtigere programafvikling. Det, at kompressionen ikke foregår som foreskrevet i det matematiske princip, er negligibelt, da resultatet har en tilfredsstillende lyd og karakteristik

8.5 Implementation af kompressor

Kompressoren implementeres som en algoritme, der er opbygget som i flowchartet på figur 8.2 på side 59. Forstærkningskoefficient beregnes som anvist på flowchartet på fi-

gur 8.8.

De tre filtre: Højpasfilteret, lavpasfilteret og detektoren implementeres efter Direkte Form 1 strukturen. Derved er der, hvis koefficienterne, a_1, a_2, b_0, b_1, b_2, (1-λ) og λ er skaleret rigtigt ikke risiko for overflow. De skaleres udfra at a_0 skal være 1 og derefter må de øvrige koefficienter ikke være større end 1, når de deles med 2^N. Strukturen for filteret ses på figur 8.10.



Figur 8.10: *Struktur i Direkte Form 1 realisationen.*

For 1. ordens filteret til detektoren fjernes den nederste del af strukturen.

Der er indsat en foranstaltning som gør, at forstærkningskoefficienten ikke kan komme under 0,5 og ligeledes er der sørget for, at forstærkningskoefficienten ikke kan blive sat til mere end én. Dette er gjort for, at forhindre overflow og sikre at signalet ikke "dør ud".

Implementering af brugerinterface

Opdateringen af kompressorværdierne sker ved at bruge inputtet fra brugerinterfaceregistret, som indirekte adresseringsoffset. Den valgte adresse har således en koefficient, der skal opdateres.

Det er valgt, at de første 10 adresser har værdier, der ændrer threshold fra 0dB til -30dB, og de sidste 6 adresser ændrer kompressionsfaktoren fra A_h til F_h. Den ændrede værdi lægges det sted i RAM, hvor funktionen kompressor henter sine threshold og kompressionsfaktor værdier fra.

8.6 Delkonklusion

Under udarbejdelsen af kompressoren, er der foretaget simuleringer i MATLAB af de to metoder til beregning af forstærkningskoefficienter (se appendiks C på side 111. Blandt andet på grund af en hurtigere programafvikling ved metode 2, er der arbejdet videre med denne metode.

Ved realiseringen af kompressoren har det været nødvendigt at foretage en modificering af koefficienterne til hhv. højpas- og lavpasfiltret. Denne ændring er blevet foretaget på baggrund af lyttetests og simuleringer i MATLAB. Tillige blev der afviget fra princippet

om at fastlægge threshold-værdier udfra et niveau på 0dB ved 90% af maksimal udstyring. Dette blev gjort, fordi der viste sig at være en for lille kompression til at det kunne høres. Istedet blev threshold-værdierne valgt til hhv. 73%, 31%, 9% og 1,5% af maksimal udstyring. Steppene kan indstilles til størrelser på: 6,1 $^{0}/_{00}$, 3,05 $^{0}/_{00}$ eller 1,53 $^{0}/_{00}$, 0,61 $^{0}/_{00}$. Der kan således, fra brugerens side, vælges mellem de forskellige fastlagte værdier for threshold og stepstørrelser.

Det er udfra lyttetesten, beskrevet i C på side 111, vurderet at kompressoren virker som forventet.

På baggrund heraf konkluderes det, at kompressoren er funktionel og vellykket.

KAPITEL 9

Flanger

Dette kapitel indeholder dokumentation af den designede flanger-algoritme. Først gennemgåes princippet i flangeren, derefter opstilles nogle løsningsmuligheder. Den mest hensigtsmæssige løsning vælges, og derefter realiseres denne.

9.1 Krav til flanger

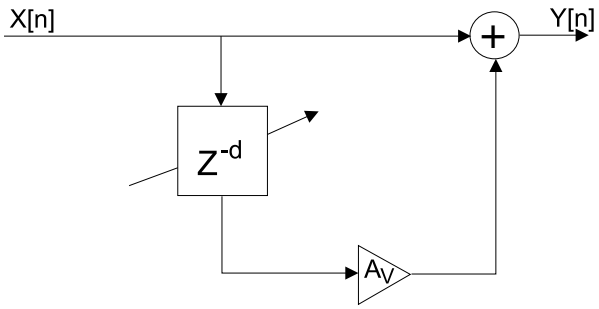
Der opstilles følgende krav til flangeren:

- Variabel flangerfrekvens.
- Realiseres digitalt.
- Generelle krav fra kravspecifikationen.

9.2 Generelt om flangeren

Et signal med flangereffekt består af det "originale" inputsignal, hvortil et tidligere inputsignal adderes. Det foregående inputs delay i forhold til det nuværende input, bestemmes ud fra en sinussvingning. Dette princip ses på figur 9.1, hvor systemet betragtes i Z-domænet.
Flangeren skal have et delay, d[n], der varierer sinusformet imellem 1 og 10ms [10], hvorfor d[n] kan skrives som:

$$d[n] = sin(\omega_{flanger} \cdot n) \tag{9.1}$$



Figur 9.1: *Blokdiagram for flangeren.*

Med den valgte samplingsfrekvens, f_s, på 22050Hz, kan d[n]'s maksimum og minimumsværdier findes:

$$T_s = \frac{1}{f_s} = \frac{1}{22050} = 45,3\mu s. \tag{9.2}$$

$$d[n]_{min} = \frac{t_{min}}{T_s} = \frac{1ms}{45,3\mu s} = 22samples \tag{9.3}$$

$$d[n]_{max} = \frac{t_{max}}{T_s} = \frac{10ms}{45,3\mu s} = 220samples \tag{9.4}$$

Dermed er det fundet, at sinusfunktionen kontinuert skal svinge mellem sample n-22 og n-220 i forhold til input[n]. Sinusfunktionens offset beregnes til:

$$flangeroffset = \frac{d[n]_{max} + d[n]_{min}}{2} = \frac{220 + 22}{2} = 121 \tag{9.5}$$

Funktionen for d[n] kan dermed skrives som:

$$d[n] = offset + (d[n]_{max} - offset) \cdot sin(\omega_{flanger} \cdot n) \tag{9.6}$$

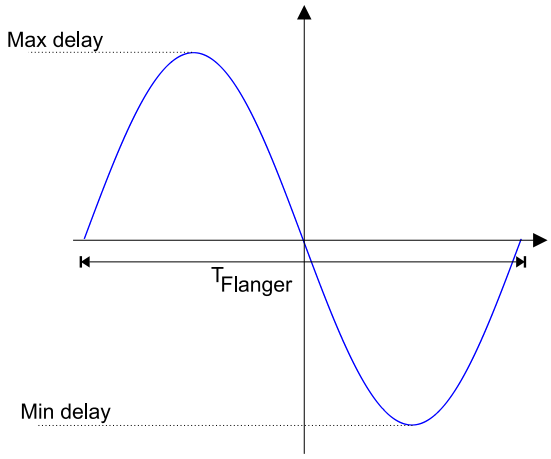
$$\Downarrow$$

$$d[n] = 121 + 99 \cdot sin(\omega_{flanger} \cdot n) \tag{9.7}$$

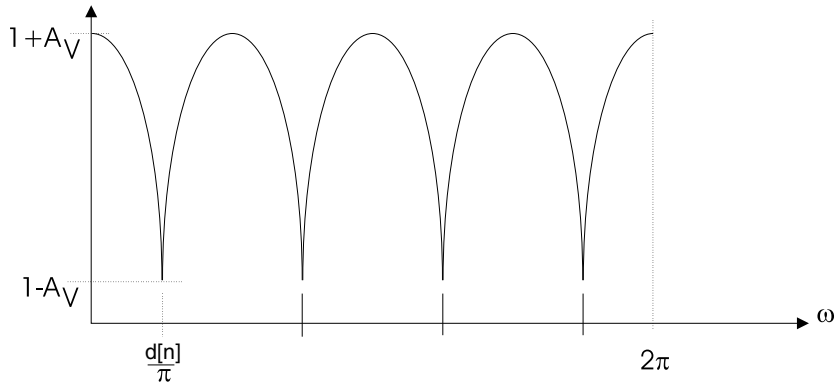
Denne funktion kan plottes som vist på figur 9.2 på næste side, hvor det maksimale delay er 220 samples og mindste delay er 22 samples.
Flangerens effekt på frekvensspektret ses på figur 9.3 på modstående side, hvor der sker en udfasning af signalet ved bestemte frekvenser. Disse frekvenser varierer med den sinus, hvormed flangeren svinger.

9.2.1 Interpolation

Da flangeren skal implementeres i DSP-systemet, bliver inputsignalet diskretiseret, idet det samples før det behandles. Flangerens output består som før nævnt af det aktuelle inputsignal, input[n], adderet med et tidligere input, input[n-delay], hvor delay ligger mellem 22 og 220 samples.
Jo mere præcist delay er defineret, desto "pænere" vil outputsignalet lyde. Det vil sige, at hvis delay er et heltal mellem 22 og 220, vil der være 199 samples at finde det delayede sample i. Ved at forøge opløsningen med 10, således at delay kan variere mellem 22 og



Figur 9.2: *Flanger effektens sinussvingning.*

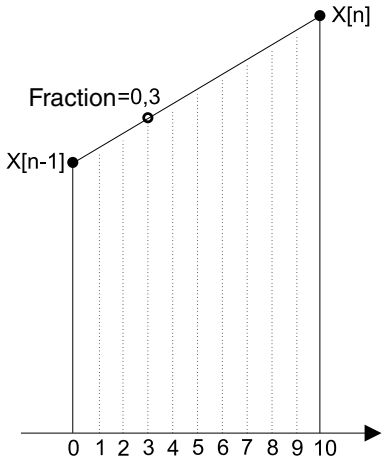


Figur 9.3: *Flangerens effekt på frekvensresponset.*

220 i intervaller med een decimal, vil der være 1980 samples at finde det delayede signal i. Problemet er, at det vil kræve, at der samples 10 gange så hurtigt, for at der efterfølgende er 10 gange så mange samples at vælge delayet i. En anden løsning er at anvende interpolation.
Interpolation er en matematisk måde at forbedre opløsningen af signalet på. Dette kan gøres på flere forskellige måder, hvoraf den simpleste, er lineær interpolation. Denne metode går ud på, at mellemrummet mellem to samples, anses for at være udfyldt af en lineær forbindelse mellem de to samples, i stedet for at være tomt. Ønskes en interpolation på 10 gange, opdeles den lineære forbindelse i 10 samples, og der kan dermed findes samples med op til een decimal (fraction). Dette ses på figur 9.4 på næste side.
Den matematiske metode til beregning af de lineært interpolerede samples er:

$$x[n - delay] = (x[n - a] - x[n - (a + 1)]) \cdot fraction + x[n - a] \tag{9.8}$$

På figur 9.4 på den følgende side vil ligning 9.8 svare til a=0 og fraction=0,3.
Ved anvendelse af denne metode, vil delay være repræsenteret ved 1980 samples ved den oprindelige samplingsfrekvens.



Figur 9.4: *Beregning af interpoleret signal.*

9.3 Metoder til realisering af flanger

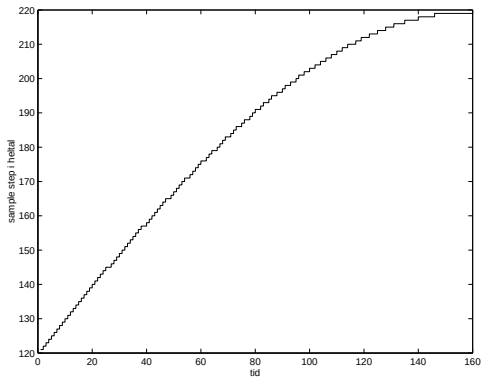
Flangeren kan realiseres på flere forskellige måder. Det vælges at undersøge 2 metoder: Den tabellagte sinus, og den koefficientlagte sinus.

9.3.1 Metode 1: Tabellagt sinus

Een måde at foretage sinusberegningerne på, er ved at tabellægge sinusfunktionen for en periode, og ved beregning foretage opslag i denne tabel. Kvantiseres sinusen i steps af heltal, opløses sinusen i 199 steps, svarende til antallet af samples i delay. Dette gøres i MATLAB med følgende kode:

```
t=1:1:160;
fl99=fix(99*sin((2*pi*t)/640))+121;
stairs(fl99)
```

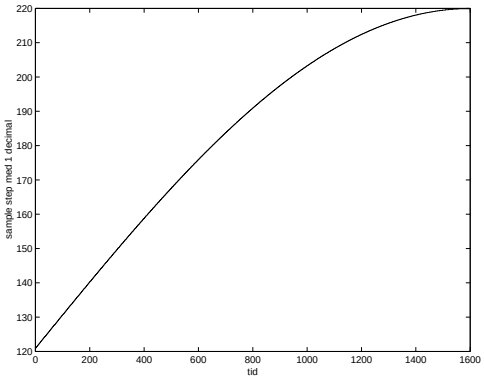
På figur 9.5 ses det, at sinusen bliver meget kantet, og derved giver grove spring imellem de forskellige samples. Flanger-sinusen kan dog forbedres ved at anvende interpolation.



Figur 9.5: *1/4 periode af tabellagt sinus, med en opløsning på 199 steps.*

Interpoleres med en faktor 10, vil sinussvingningen blive opløst med een decimal. Ulempen bliver, at tabellen derved også forøges med 10 gange, og derved vil fylde mere i hukommelsen ved implementation i DSP-systemet. Sinusen plottes ud fra følgende MATLAB kode:

```
t=1:1:1600;
fl990=fix(990*sin((2*pi*t)/6400))+1210;
stairs(fl990/10)
```

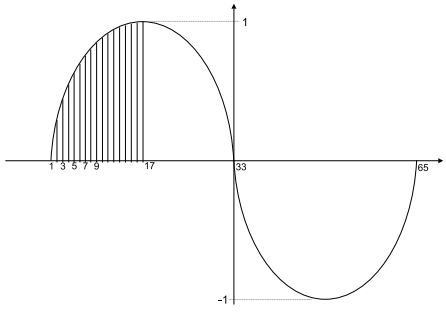


Figur 9.6: *1/4 periode af tabellagt sinus med 10 gange bedre opløsning.*

Det ses af den interpolerede sinusfunktion på figur 9.6, at den tabellagte sinus har meget bedre opløsning, og at der dermed ikke forekommer nær så grove steps, som i den ikke-interpolerede version.
Denne metode simuleres i MATLAB (se kode i appendiks, og hør resultatet på CDROM'en). Efter at have lyttet til resultatet af simuleringen, vurderes det, at denne metode ikke giver det ønskede resultat. MATLAB koden til simuleringen ses i appendiks C.5 på side 128.

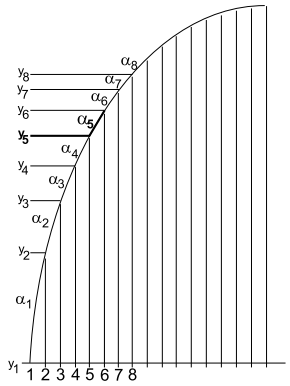
9.3.2 Metode 2: Tabellagte koefficienter

Metode 2 til beregning af flangerens sinus tager udgangspunkt i, at en sinussvingning deles op i n lige store dele, og hver del tilnærmes med en hældningskoefficient (α) og en offsetværdi (y) (se figur 9.7).
Da en sinus stort set er stykkevis lineær i små intervaller, vil denne metodes tilnærmelse



Figur 9.7: *Inddeling af sinus kurve i n=64 dele.*

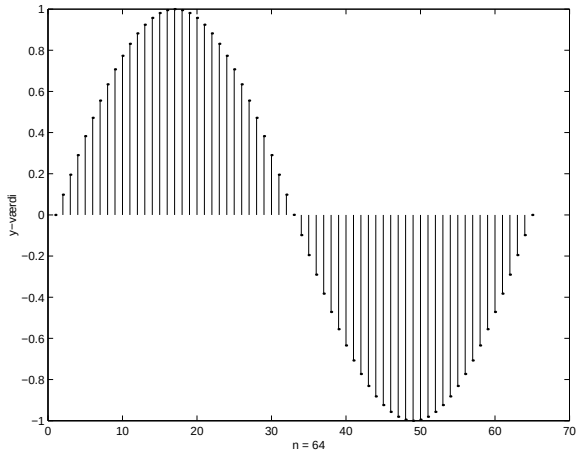
ikke afvige væsentlig fra en kontinuert sinusfunktion.
Disse koefficienter tabellægges med en y-værdi og en α-værdi til hver n (se figur 9.8).
For n=64 kan y-værdierne og α-værdierne beregnes, med følgende MATLAB kode:



Figur 9.8: *1/4 periode af tabellagte koefficienter.*

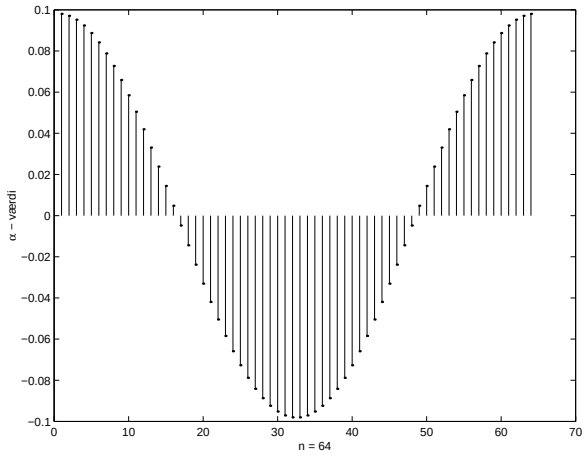
```
t=0:1:65;
y=sin((2*pi*t)/64);
stem(y,'.')
for l=1:64
   a(l)=y(l+1)-y(l);
end
figure
stem(a,'.')
```

På figur 9.9 ses værdierne for y, og på figur 9.10 på næste side ses værdierne for α.



Figur 9.9: *Graf for y koordinater til tabellagte koefficienter.*

Hver af de 64 dele, opdeles yderligere i 2^{15} lige store stykker, som skal udgøre fractiondelen ved interpolation. Dermed kan der beregnes en output-værdi for det pågældende step i en opløsning på 2^{15}. Sinusen er hermed opløst i $n_{total} = 64 \cdot 2^{15} = 2.097.152$ steps (= thetaværdi). Det aktuelle output gives ved placeringen på x-aksen. Frekvensen, hvormed sinusen gennemløbes, gives ved det antal steps der forøges med per sample.

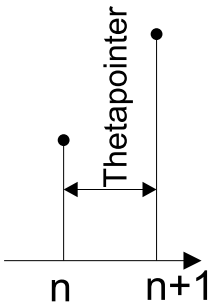


Figur 9.10: *Graf for hældningen (α) af tabellagte koefficienter.*

Tabelopslaget foretages ud fra kendskabet til n-værdien. Ud fra n-værdien findes den aktuelle α og y-værdi. Dermed kan det øjeblikkelige delay beregnes som:

$$delay = (y + \alpha \cdot thetapointer) \cdot amplitude + offset$$

hvor y er y-værdien, amplituden er 99 og offset er 121. Thetapointeren styres af fractiondelen, og ses på figur 9.11.
Metode 2 simuleres ligeledes i MATLAB, og vurderes med henblik på valg af løsningsme-



Figur 9.11: *Thetapointeren kan antage 2^{15} steps for hver α-værdi.*

tode (resultatet kan høres på CDROM'en).
Det konkluderes, at MATLAB-simuleringen af metode 2 giver en flangerlyd tættere på det forventede, i forhold til simuleringen af metode 1.

9.4 Valg af løsningsmetode

Der er i de foregående afsnit beskrevet 2 forskellige metoder til implementering af flangeren. Begge metoder arbejder med tabelopslag. Metode 1 bygger på en tabel med en kortlagt sinus, og dermed kræver mest tabelplads, bygger metode 2 på tabellægning af hældningskoefficienter og y-værdier af en sinus i 64 steps. Dermed kræver metode 2 kun 2

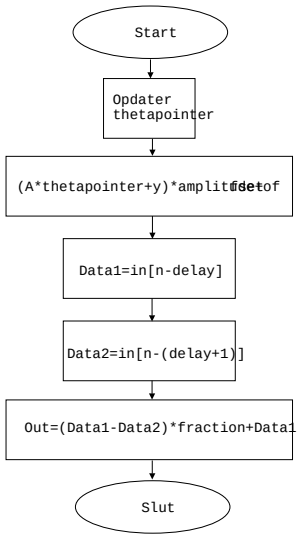
· 64 words tabel, hvor metode 1, med den beskrevne sinusopløsning, vil kræve 1600 words. Endvidere benyttes der 16-bit interpolation i metode 2, hvor metode 1 kun benytter en interpolation på 10 gange. Derved bliver outputsignalet mere præcist beregnet, og det giver mindre støj.
Det vælges at arbejde videre med metode 2, da den, udover den forholdsvis korte tabel, også lød bedre ved MATLAB-simuleringen.

9.5 Design af flanger

Når det delayede signal er fundet, tages det efterfølgende datasample, og differencen mellem de 2 datasamples ganges den beregnede brøkdel. Herved findes det interpolerede flangeroutput, som til sidst adderes til det oprindelige inputsignal.
Funktionen er, at der beregnes et decimaltals output ud fra sinusen. Dette output består af delay (heltal) og fraction (brøk). Delayet er det antal samples, der skal gåes tilbage og hentes input fra, og fractionen er et udtryk for vægtningen af data'et fra det fundne sample. Outputtet fra flangeren bliver således det oprindelige inputsignal adderet til flangerdatasignalet. Dette kan skrives som:

$$Output[n] = in[n] + ((in[n-delay] - in[n-(delay+1)]) \cdot fraction + in[n-delay]) \quad (9.9)$$

Der opstilles et flowdiagram for flangerens funktion, og senere vil dette diagram danne grundlag for implementeringen af flangeren i DSP'en. Flowdiagrammet ses på figur 9.12



Figur 9.12: *Flowdiagram for flangeren.*

9.5.1 Implementation af flanger i DSP

I afsnit 9.2 på side 71 blev det fundet, at flangersinusen skal svinge mellem sample 22 og 220. Det betyder, at der skal implementeres en inputbuffer til flangeren, hvor der er plads

til mindst 220 samples.
Frekvensen, hvormed flangersinusen skal svinge, bestemmes ud fra størrelsen af de steps thetapointeren foretager. Stepstørrelsen implementeres brugervalgt, således at den styres via brugerinterfacets thumbleswitch. Hermed er der 16 indstillinger for stepstørrelsen der beregnes ud fra:

$$step = stepoffset + konstant \cdot brugerstepvalg \tag{9.10}$$

hvor stepoffset sættes til 6, og konstanten sættes til 6. Dette gøres ud fra et lyttemæssigt skøn. Frekvensmæssigt vil det give 16 frekvenser, som ses i tabellen nedenfor.

IO	0	1	2	3	4	5	6	7
step	6	12	18	24	30	36	42	48
frekvens [Hz]	0,063	0,126	0,189	0,252	0,315	0,379	0,442	0,505
IO	8	9	A	B	C	D	E	F
step	54	60	66	72	78	84	90	96
frekvens [Hz]	0,568	0,631	0,694	0,757	0,820	0,883	0,946	1,009

9.5.2 Brugerinterface til flanger

Opdateringen af flangerværdien sker ved at brugerinterfaceregistret læses og dekodes. Værdien fra thumbleswitchen (mellem 0-F_h) bruges til at opdatere frekvensen, hvormed flangeren svinger. Der laves ændringer således, at flangerens frekvens kan varieres mellem 0,06Hz og 1Hz. Dette gøres ved at bruge inputtet til at styre de steps som $STEP$ springer med (se kapitel 9 afsnit 9.5.1 på forrige side).

$$STEP = 6 + 6 \cdot brugerinterfaceregister \tag{9.11}$$

9.5.3 Revurdering af flangerkonstanter

Flangeren er opbygget ud fra metoden beskrevet i [10], hvor det beskrives, at flangerdelayet bør ligge mellem 1 - 10ms (mellem 22 og 220 samples). Flangeren blev derfor implementeret ud fra disse værdier. Efter at have implementeret flangeren i DSP'en, er der foretaget en revurdering af disse værdier. Ved at have et delay mellem 22 og 220 samples, kommer flangerlyden til at lyde "forkert". Ved at nedsætte amplituden på sinusen og offsettet opnås den forventede flangereffekt. Amplituden sættes til 10 samples (før 99 samples), og offsettet sættes til 11 samples (før 121 samples). Herefter vurderes det, at flangeren lyder som forventet, og at der sker en udfasning.

9.6 Delkonklusion

Der blev opstillet 2 metoder til designs af flangereffekt. Den valgte metode byggede på stykkevis linearisering af sinus i 6 bit (64 intervaller), hvor hældningskoefficienter og amplituder tabellægges i en opløsning på 16 bit. Da kvantiseringen af sinusen foretages ud

fra den stykkevis lineariserede sinus, opnås en større sinusnøjagtighed i forhold til den forkastede metode 1.
Den valgte metode er blevet implementeret i DSP'en, og ud fra lyttetest er det vurderet, at de i kildematerialet angivne flangerdata, behøvede tilretning. Efter denne korrektion af flangerkonstanter, er flangerens output tilfredsstillende, og vurderes som en korrekt flangereffekt.

KAPITEL 10

Rumklang

Dette kapitel omhandler den del af det digitale software, der varetager rumklangeffekten. Denne effekt skal få guitaren til at lyde, som om den spiller i et rum med forskellig efterklang.

Afsnittet er baseret på [[10], s. 255-388] og [16].

10.1 Krav til rumklang

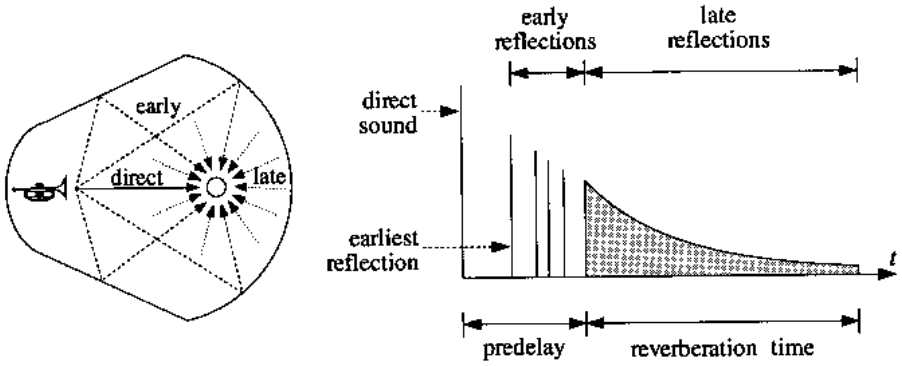
Følgende krav til rumklangen er opstillet:

- Variabel efterklangstid på mellem 0,8-3,0s.
- De generelle krav til effekterne opstillet i kravspecifikationen.

10.2 Beskrivelse af rumklang

Rumklang forekommer, når lyd udbredes i et rum. Når lyden rammer væggene, reflekteres lyden, og den oprindelige lyd blandes med reflektioner og reflektionernes reflektioner. Rumklangen kan deles op i 3 forskellige perioder: Direkte lyd, de tidlige reflektioner og de sene reflektioner. Dette er illustreret på figur 10.1, hvor impulsresponsen af et rum, og måden lyden reflekteres på, er illustreret (figuren er taget fra [[10], s. 368]).

Når det drejer sig om refleksioner i et rum, anvendes der et målestoksforhold, der hedder efterklangstid. Denne er defineret til det tidspunkt hvor impulsresponsen er faldet med 60dB. Længden af efterklangstiden er proportional med størrelsen af rummet, hvorfor



Figur 10.1: *Impulsresponsen for et rum.*

store koncertsale ofte har efterklangstider på 1,5-3s, mens mindre koncertsale har fra 0,8-1,5s (fra [16]).

10.3 Løsningsmuligheder

Konstruktionen af rumklangeffekten kan foretages på flere måder, men impulsresponsen for effekten skal ligne et rums. Derfor kunne det være nærliggende at benytte et FIR filter, hvis koefficienter er impulsresponsen for et specifikt rum. Derved kunne impulsresponsen for forskellige rum lægges i ROM (lager), og det ville være muligt at skifte mellem forskellige specifikke rum. Ulempen ved denne realisation er, at for at rumklangen i et FIR filter kan være op til 3 sekunder lang, skal længden af FIR filtret ved f_s=22050Hz være:

$$FIR_{length} = 3s \cdot f_s = 3 \cdot 22050 = 66150 samples \tag{10.1}$$

Dette filter skal så gennemløbes ved f_s=22050Hz, hvilket kræver mindst:

$$INSTRUKTIONER = 66150 \cdot 22050 = 14586 MIPS \tag{10.2}$$

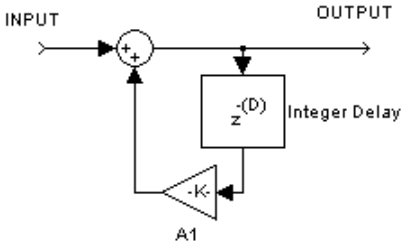
Dette vil ikke være muligt at realisere i den benyttede DSP.

Derfor ses der på anvendelsen af IIR filtre. En af de simpleste rumklang-effekter består af et kamfilter, hvis blokdiagram og impulsrespons er vist på figur 10.2 og 10.3. Som det ses falder impulsresponsen med en faktor A hver gang der er gået et delay D (dvs. amplituden for hver delay, D, skrives som: A,A^2,A^3,A^4,....). Differensligningen for kamfiltret er:

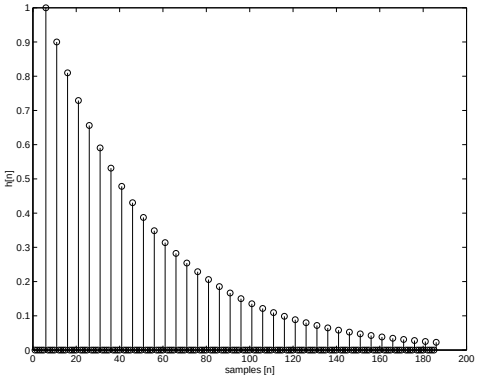
$$y[n] = x[n] + A \cdot y[n - D] \tag{10.3}$$

Som det ses af impulsresponsen (figur 10.3) ligner responsen for kamfilteret responsen for et rum (se evt. figur 10.1). Ved at bruge et kamfilter opnås rumklangeffekten. Dog lyder denne metallisk ("digital"), da der mellem stavene på filtret ikke findes nogle informationer. For at forhindre denne metalliske lyd kan der benyttes flere kamfiltre samt allpassfiltre. Dette kan realiseres ved at bruge "Schroeders reverberator". Denne er illustreret på figur 10.4.

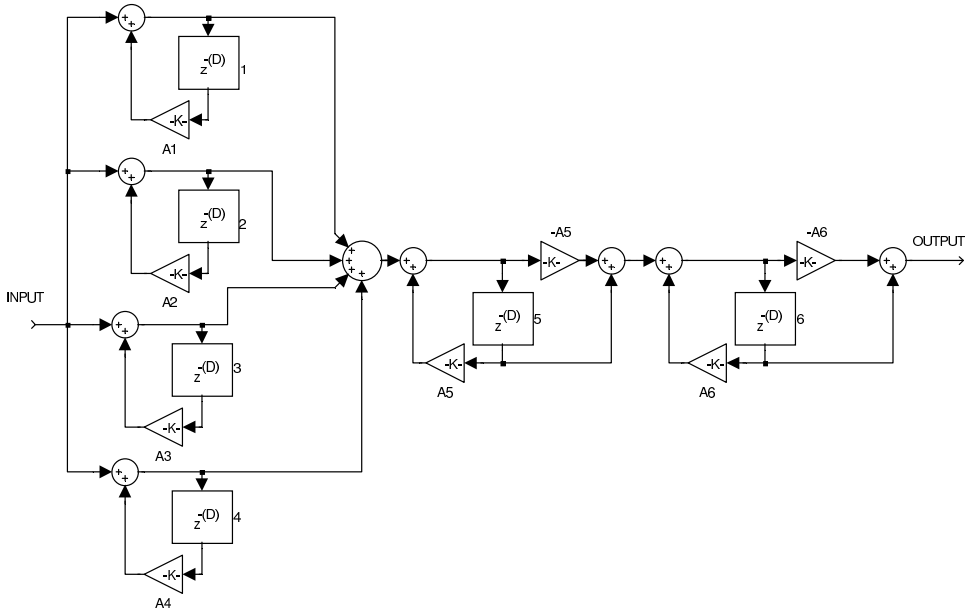
Det vælges at implementere Schroeders reverberator som den benyttede rumklang.



Figur 10.2: *Blokdiagram for et kamfilter.*



Figur 10.3: *Impulsrespons for et kamfilter med D=5 og a=0,9.*



Figur 10.4: *Blokdiagram for "Schroeder's reverberator".*

10.4 Design af Schroeders reverberator

I designet af Schroeders reverberator tages der udgangspunkt i figur 10.4. Her ses det, at Schroeder's reverberator består af 4 kamfiltre og 2 allpassfiltre. Impulsresponsen for dette system skal ligne impulsresponsen for et rum, og efterklangstiden på dette rum skal kunne varieres mellem 0,8 - 3 sekunder.

Det, der giver den nedadgående kurve på impulsresponsen, er kamfiltrene. Disse sørger for, at impulsen langsomt uddør, hvorved der opstår stave. Disse stave sendes så igennem 2 allpassfiltre, hvor der i det første filter bliver lagt et "slør" efter hver stav. Det sidste allpassfilter lægger et "slør" over sløret, således at impulsresponsen bliver mere kontinuert.

10.4.1 Kamfiltrene

Delay'et i disse filtre ligger typisk mellem 0,03-0,05 sekunder [16]. Delay'et vælges til et "ulige" tal (f.eks. primtal) for at undgå for mange sammenfaldende stave. De 4 delay's skal ligge mellem 0,03-0,05 sekunder, hvilket giver:

$$D_1 = 0,03 \cdot f_s = 0,03 \cdot 22050Hz = 661,25samples \quad (10.4)$$

$$D_2 = 0,05 \cdot f_s = 0,05 \cdot 22050Hz = 1102,5samples \quad (10.5)$$

Det er valgt at bruge følgende delay's:

- D_1=661 samples
- D_2=787 samples
- D_3=887 samples
- D_4=1021 samples

For at få impulsresponsen til at ligne et rums, skal forstærkningen i tilbagekoblingen på kamfilter 1 (A_1) være større end A_2, som igen skal være større end A_3 osv. Dette skal opfyldes for at få en aftagende kurve (amplitude vs. tid). For at finde amplituden af A_1 til A_4 ses der på efterklangstiden. Efterklangstiden er bestemt af det største delay samt amplituden i pågældende filter (kamfilter 4). I dette filter er delay D i sekunder givet ved:

$$COMB_{delay} = D \cdot \frac{1}{f_s} = \frac{1021}{22050} = 0,046s \quad (10.6)$$

For at finde forstærkningen A_4 i de to situationer hvor efterklangstiden er 0,8-3,0 sekunder, ses der på $COMB_{delay}$ samt amplituden . Denne er givet ved [16]:

$$-60dB = A_4^{\frac{reverb-time}{COMB-delay}} \quad (10.7)$$

For efterklangstid lig 3,0 sekunder fås A_4:

$$\frac{1}{1000} = A_{4_{3sek}}^{\frac{3,0s}{0,046s}} \quad (10.8)$$

$$\Downarrow$$

$$0,001^{\frac{1}{65,217}} = A_{4_{max}} = 0,899 \quad (10.9)$$

For efterklangstid lig 0,8 sekunder fås A_4:

$$\frac{1}{1000} = A_{4_{0.8sek}}^{\frac{0,8s}{0,046s}} \quad (10.10)$$

$$\Downarrow$$

$$0,001^{\frac{1}{17,391}} = A_{4_{max}} = 0,672 \quad (10.11)$$

Da A_1 til A_3 skal være større end A_4, for at sikre den aftagende kurve og da A<1, er der valgt følgende amplituder til den største efterklangstid (efterklangstiden på 3s):

- A_1=0,983
- A_2=0,945
- A_3=0,923
- A_4=0,899

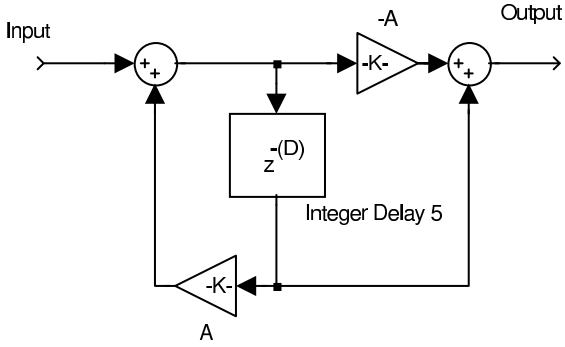
Det skal være muligt at skifte mellem forskellige rumklangsformer, og det er valgt at beholde det samme forhold mellem $A_1 - A_4$, således at ændringen fra 3,0s til 0,8s foretages ved at gange en konstant (K) på alle A'er. K er 1 for $A_{4_{3sek}}$. Konstanten varierer så i 16 trin ned til $A_{4_{0,8sek}}$. Konstanten K for $A_{4_{0,8sek}}$ er:

$$K = \frac{K_{3,0sek} \cdot A_{4_{0,8sek}}}{A_{4_{3,0sek}}} = \frac{1 \cdot 0,672}{0,899} = 0,748 \tag{10.12}$$

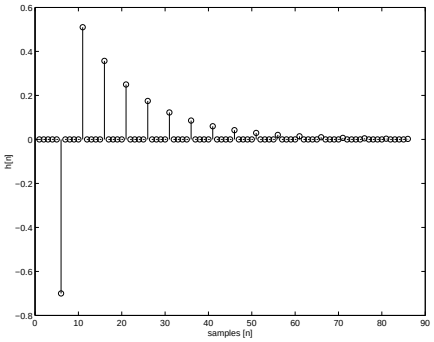
Derved skal 0,748 < K < 1,0.

10.4.2 Allpassfiltrene

Allpassfiltret benyttes i "Schroeders reverberator" for at fylde informationer ind mellem de stave som kamfiltrene laver. Blokdiagrammet for allpassfiltret ses på figur 10.5.



Figur 10.5: *Blokdiagram for et allpassfilter.*



Figur 10.6: *Impulsrespons for et allpassfilter med D=5 og A=0,7.*

Allpassfiltrene har en lineær frekvensrespons. Ændringen af signalet sker på fasen. Differensligningen for et allpassfilter er:

$$y[n] = A \cdot y[n - D] - A \cdot x[n] + x[n - D] \tag{10.13}$$

Ses der på impulsresponsen for et allpassfilter ser den ud som på figur 10.6, hvor delay'et er 5 og amplituden er 0,7. Ved at studere blokdiagrammet kan amplituden for hver delay, D, skrives som: $(-A)$, $1 - A^2$, $A \cdot (1 - A^2)$, $A^2 \cdot (1 - A^2)$, $A^3 \cdot (1 - A^2)$, $A^4 \cdot (1 - A^2)$, ... Delayet i allpassfiltrene skal være kortere end delay'et til kamfiltrene, fordi de skal lægge

et "slør" over hver af stavene fra kamfiltret. Ifølge [16] skal allpass delay'et ligge mellem 0,005-0,02s svarende til 110-441 samples. Det vælges at lægge delay'ene D_5 og D_6 på:

$$D_5 = 111$$

$$D_6 = 239$$

Amplituden A_5 og A_6 sættes til henholdsvis:

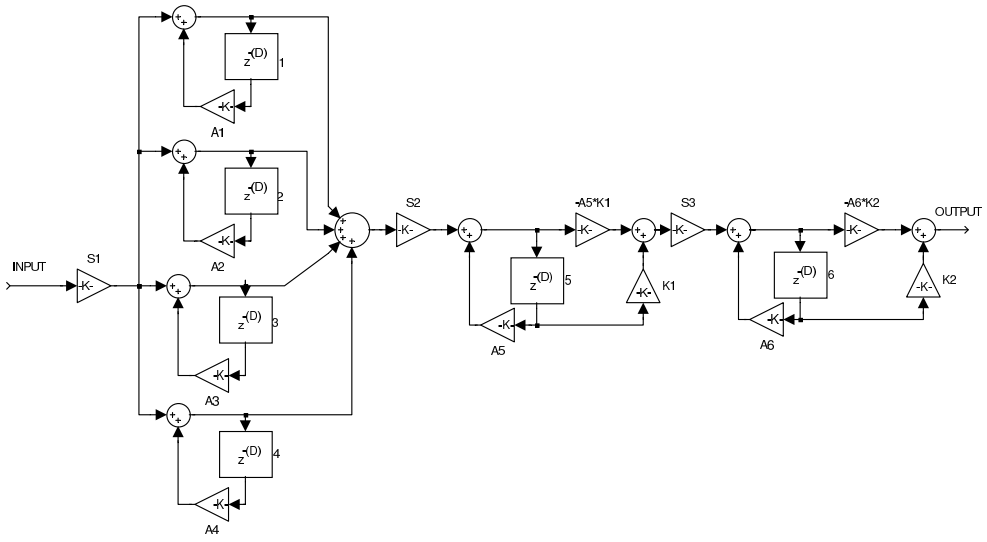
$$A_5 = 0,901$$

$$A_6 = 0,713$$

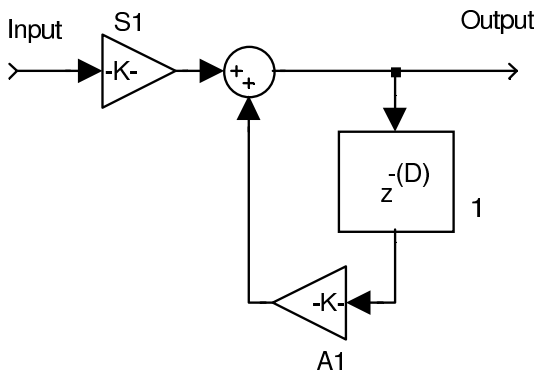
Dette valg er truffet, da impulsresponsen for allpassfiltret er længere i tid, men mindre i amplitude jo større A er (0<A<1). Ved $A_6 = 0,7$ fåes en større amplitude, men en kortere efterklangstid end $A_5 = 0,9$. Da A_5 har det korteste delay betyder det, at få høje stave fra allpassfilter 1 (A_6) bliver efterfulgt af et allpassfilter med længere delay og mindre amplitude (allpassfilter 2).

10.4.3 Skalering

For at undgå overflow i processeren skal "Schroeders reverberator" skaleres. Der benyttes variansskalering fremfor "worst case" skalering, da det ønskes, at signal/støj forholdet skal være højst muligt, samtidig med at overflow ikke må forekomme. Skaleringsfaktorerne, der bruges forskellige steder i systemet, er illustreret på figur 10.7.



Figur 10.7: *Skaleringsfaktorer i "Schroeders reverberator".*



Figur 10.8: *Skalering af kamfiltre.*

Skalering af kamfiltre

For at sikre at kamfiltrene ikke får overflow skaleres disse. Dette er vist på figur 10.8, hvor S_1 er skaleringsfaktoren.

S_1 er ved variansskalering givet ved (fra [[9], s. 401]):

$$S_1^2 = \frac{1}{\sum_{n=0}^{\infty} |h[n]|^2} \tag{10.14}$$

Hvor $h[n]$ er impulsresponsen på udgangen af kamfiltret. Den summerede impulsrespons er:

$$\sum_{n=0}^{\infty} |h[n]|^2 = 1^2 + A^2 + (A^2)^2 + (A^3)^2 + \cdots \tag{10.15}$$

$$\Updownarrow$$

$$\sum_{n=0}^{\infty} |h[n]|^2 = 1 + A^2 + A^4 + A^6 + A^8 + A^{10} + \cdots = \sum_{n=0}^{\infty} A^{2n} \tag{10.16}$$

Summen findes ved brug af MATLAB, hvor summen udregnes op til A^{200000}. A'et der benyttes, er "worst case", hvilket er det filter med størst A (A_1=0,983). Indsættes A_1 i ligningen bliver S_1:

$$S_1^2 = \frac{1}{\sqrt{29,6639}} \tag{10.17}$$

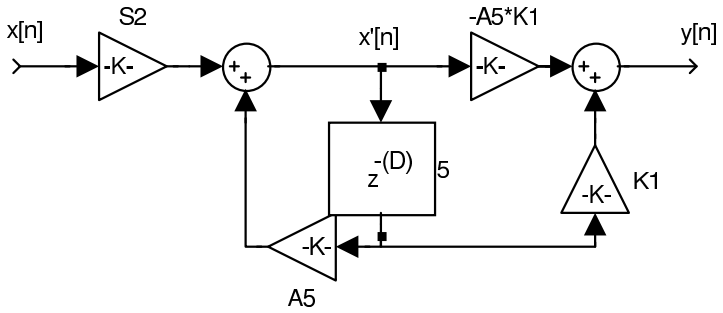
$$\Downarrow \tag{10.18}$$

$$S_1^2 = 0,1836 \Rightarrow S_1 = 0,428 \tag{10.19}$$

Det betyder, at x[n] skaleres med en faktor 0,428 inden den indsættes i filtrene. Derved kan der i "worst case" komme en værdi ud fra hvert kamfilter, der er i nærheden af 1. Dette betyder, at summen af disse 4 kamfiltre kan blive en værdi der nærmer sig 4. Derfor skal skaleringsfaktor S_2 ikke kun skaleres for allpassfiltret, men også for outputtet fra kamfiltrene. Dette gør, at S_2 som beregnes nedenfor, skal skaleres yderligere med en faktor $\frac{1}{4}$.

Skalering af allpassfiltrene

For at undgå overflow i allpassfiltrene skaleres disse efter variansmetoden og som direkte form II. Skaleringsfaktorerne ses på figur 10.9.



Figur 10.9: *Skalering af allpassfiltre.*

Ved brug af direkte form II er der 2 knudepunkter, der skal undersøges for overflow: Et internt knudepunkt ($x'[n]$ se figur 10.9) og outputtet ($y[n]$). Først undersøges impulsresponsen fra inputtet (til allpassfilter 1) og til det interne knudepunkt i allpassfilter 1. Denne regnes på samme måde som impulsresponsen for kamfiltret. Derfor er S_2 (ved A5=0,901):

$$S_2 = 0,659 \tag{10.20}$$

Så ses der på faktoren K_1. Denne er givet ved ([[9], s. 397-412]):

$$K_1 = S_2^{-1} \cdot g_1 \tag{10.21}$$

g_1 er givet ved:

$$g_1 = \frac{1}{\sqrt{\sum_{n=0}^{\infty} |h[n]|^2}} \tag{10.22}$$

$$\Downarrow$$

$$g_1 = \frac{1}{\sqrt{1^2 + A^2 + (1-A^2)^2 + (A \cdot (1-A^2))^2 + (A^2 \cdot (1-A^2))^2 + ..}} \tag{10.23}$$

$$\Downarrow$$

$$g_1 = \frac{1}{\sqrt{2}} = 0,707 \tag{10.24}$$

Derved bliver K_1:

$$K_1 = S_2^{-1} \cdot g_1 = 0,659^{-1} \cdot \frac{1}{\sqrt{2}} = 1,074 \tag{10.25}$$

Dernæst findes K_2 på samme måde, blot ved allpassfilter 2 og A=0,713:

$$K_2 = S_3^{-1} \cdot g_2 = 0,837^{-1} \cdot \frac{1}{\sqrt{2}} = 0,8444 \tag{10.26}$$

10.4.4 Software

Kamfiltrene designes udfra formlen:

$$y[n] = S \cdot x[n] + A \cdot y[n - D] \tag{10.27}$$

hvor S_1 er den fundne skaleringsfaktor. Der laves en ringbuffer til outputtet på hvert kamfilter. Dennes længde er bestemt af delay'et i kamfiltret.

Som det ses af figur 10.9 er allpassfiltret delt op i 2 dele med et internt knudepunkt. Dette er gjort, da der er behov for en ringbuffer, der gemmer $x'[n]$. Ved denne opdeling bruges følgende ligninger:

$$x'[n] = S \cdot x[n] + A \cdot x'[n] \tag{10.28}$$

$$y[n] = -A \cdot K \cdot x'[n] + K \cdot x'[n - D] \tag{10.29}$$

Hvor S og K er skaleringsfaktorerne.

Implementation af brugerinterfacet

Opdateringen af rumklangens koefficicenter sker ved at hæve eller sænke kamfiltrenes tilbagekoblingskonstanter ($A_1 - A_4$). Disse konstanter multipliceres med en faktor K som skal være mellem 0,748 og 1 og ændres logaritmisk.

De maksimale værdier for $A_1 - A_4$ er lagt ind som defaultværdier og som startværdier i RAM. Da brugerinterfaceregistret har en værdi, der kan variere mellem 0-F_h, og da 0,748 $<$ K $<$ 1, kræves der en tilpasning af brugerinterfaceregistret til K, hvor K samtidig skifter logaritmisk. Dette gøres ved at følgende formel (her for A_1):

$$A_1 = \left(K_{max} - \left((K_{max} - K_{min}) \cdot 2^{-B}\right)\right) \cdot A_{1_{max}} \tag{10.30}$$

$$A_1 = \left(1 - (0,252)^{-B}\right) \cdot A_{1_{max}} \tag{10.31}$$

Hvor B er brugerinterfaceregistrets binære værdi mellem 0 og 15. Dette giver den ønskede ændring af $A_1 - A_4$. Disse værdier gemmes så i RAM, der hvor funktionen rumklang bruger dem.

10.4.5 Realisering af rumklang

Efter simulering af rumklang samt implementation af software på DSP blev det ene af allpassfiltrene fjernet fra rumklangen (det sidste). Dette blev gjort, da det lyttemæssigt lød bedst med kun et allpassfilter. Ydermere ændres størrelsen af A_5 i allpassfiltret. Denne er nu 0,496 mod før 0,901.

Det viste sig også, at det var nødvendig at ændre både på delay's og forstærkningskonstanterne i kamfiltrene. Disse er ændret således at følgende værdier nu bruges til den maksimale rumklangseffekt:

$$A_1 = 0,944(78D5_h) \qquad D_1 = 801$$

$$A_2 = 0,929(76F5_h) \qquad D_2 = 850$$

$$A_3 = 0,907(7424_h) \qquad D_3 = 920$$

$$A_4 = 0,899(7311_h) \qquad D_4 = 1021$$

Indstillingskonstanten K er stadig mellem 0,748 og 1 (se evt. afsnit 10.4.1 på side 84 formel 10.12). Ved ændring af kamfiltrenes forstærkning ændres også skaleringsfaktor S_1. Denne blev initielt ændret til 0,5744 (vha. variansmetoden), der dog viste sig at give overflow. Derfor ændres skalering til $S_1 = \frac{1}{4}$, hvilket ikke giver overflow.

10.5 Delkonklusion

Dette kapitel har omhandlet design og konstruktion af effekten rumklang. I dette kapitel er designet en rumklang udfra "Schroeders Reverberator" princip. Efter simuleringer og implementation på DSP fjernes et allpassfilter fra designet på baggrund af lyttetests. Derved realiseres rumklangseffekten som 4 kamfiltre og 1 allpassfilter.

Den følgende lyttetest og beregninger på støj i systemet (se appendiks D på side 132) konkluderer, at rumklangseffekten virker efter hensigten.

KAPITEL 11

Accepttest

Dette kapitel indeholder testresultater og en konklusion af accepttesten, som er blevet udført på systemet. Kravene til denne accepttest er opstillet i kravsspecifikationen 2.4 på side 13.

11.1 Udførelse af accepttesten

11.1.1 Test 1: Indstillingstest

Denne test er udført med formål at verificere, om effekterne kan vælges og reguleres i henhold til brugerparametrene.

Testen blev udført ved at uploade programmet til DSP-systemet og derefter køre programmet. De enkelte moduler blev indkoblet et efter et, og det blev vurderet, om systemet opførte sig som forventet. Under indkobling af hvert enkelt modul blev indstillingsmulighederne afprøvet.

Det blev konkluderet, at hver af effekterne kunne indstilles vha. brugerfladen, og fungerede selvstændigt.

11.1.2 Test 2: Pålidelighedstest

Formålet med denne test var, at verificere systemets pålidelighed under indkobling af samtlige effekter.

Testen blev udført ved at uploade programmet til DSP-systemet og derefter køre programmet. Alle effekterne blev aktiveret, og det blev på subjektivt niveau vurderet, hvorvidt

systemet reagerede hensigtsmæssigt. Den subjektive vurdering blev foretaget på grundlag af lyttetest.

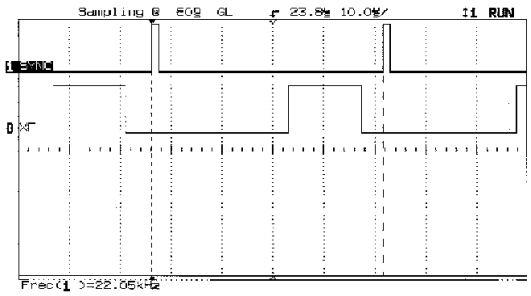
Resultatet var, at det kun var muligt at aktivere to digitale effekter samtidigt. De analoge effekter kunne uden problemer afvikles samtidigt med de øvrige effekter. Hvis der blev aktiveret mere end 2 digitale effekter, opstår der problemer med periodisk støj i systemet.

11.1.3 Test 3: Programmets afviklingshastighed

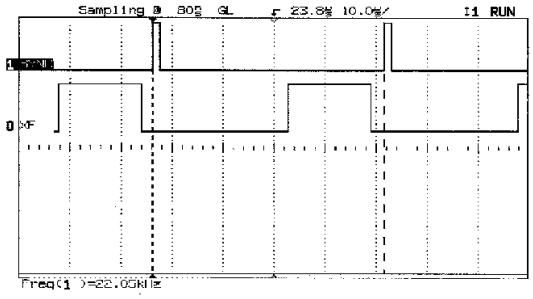
Denne test har til formål at bestemme, hvorvidt programmellet til de digitale effekter kan afvikles mellem to samples og dermed om det konstruerede system kan arbejde i realtid, når alle digitale effekter er aktiveret.

Testen blev udført ved at anvende DSP'ens XF-ben, som software-mæssigt kan enten sættes eller nulstilles. Ved at sætte benet, når en subroutine til en effekt/interruptroutine kaldes og nulstille det, når der returneres fra subroutinen, kan der ved brug af "HP Logic Analyzer" aflæses, hvor lang tid en subroutine tager. Derefter kan denne tid sammenlignes med tiden, der er til rådighed mellem to samples. Denne tid er målt ved at aflæse "Frame Synch" pulsen fra codec.

På figur 11.1, 11.2, 11.3 på modstående side og 11.4 på næste side ses, hvorledes subroutinerne blev afviklet tidsmæssigt.

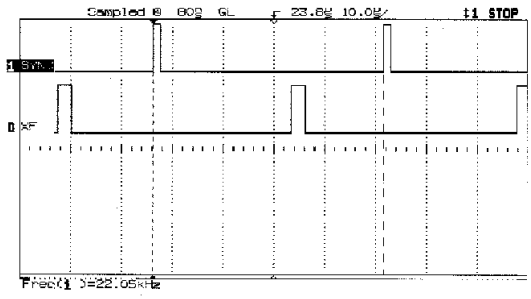


Figur 11.1: *Afviklingstid for flangereffekt.*

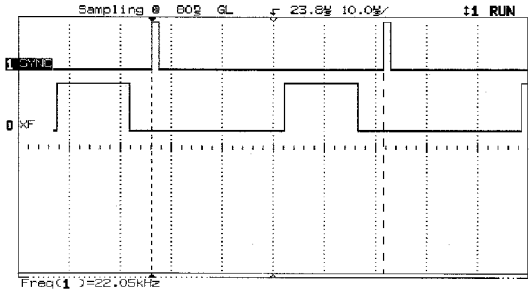


Figur 11.2: *Afviklingstid for rumklangeffekt.*

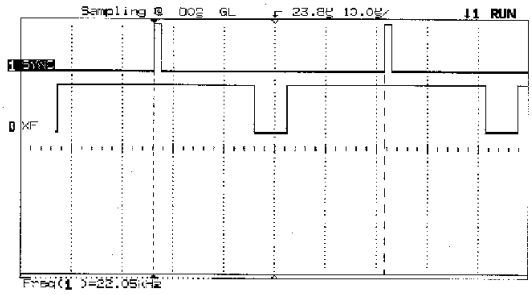
Udfra disse figurer ses det, at det lige netop kan nåes at afvikle det indenfor en samplingsperiode. Dermed er kravet om, at signalbehandlingen skal foregå i realtid opfyldt, også



Figur 11.3: *Afviklingstid for interrupt serviceroutine.*



Figur 11.4: *Afviklingstid for kompressoreffekt.*



Figur 11.5: *Afviklingstid for alle effekter indkoblet på samme tid.*

når alle effekter er koblet ind.

11.1.4 Test 4: SNR & THD

Denne test har til formål at bestemme SNR og THD for det samlede system.

Denne test udføres ved brug af THD/SNR-måleapparatet i laboratoriet. Der blev målt for det samlede system med alle effekter tilsluttet. Det blev dog valgt at udelade indkobling af forvrængeren, da den er designet til at skulle levere et støjfyldt signal. SNR blev fundet til ca. 61dB og THD blev fundet til omkring 0,35%.

Det konkluderes dermed at systemet lever op til kravet om et SNR på minimum 60dB og et THD på 0,7%.

11.2 Konklusion på accepttesten

Udfra ovenstående test af systemet, kan det konkluderes, at systemet levede op til de fire krav, og dermed er accepttesten bestået.

KAPITEL 12

Konklusion

Dette kapitel indeholder en opsummering af det konstruerede system, en samlet vurdering af dets funktionalitet og en konklusion på accepttesten. Endvidere vurderes det om semestrets PE-kurser er involveret tilstrækkeligt i projektet. Derudover er der skrevet et afsnit om, hvilke forbedringer der eventuelt kunne laves på systemet.

12.1 Det konstruerede system

I dette projekt er der blev designet og konstrueret et rack med guitareffekter. Disse effekter er: Kompressor, flanger, rumklang, equalizer og forvrænger. Disse effekter er gennem projektforløbet blevet testet hver for sig i modultests, og til sidst er det samlede system blevet testet i en accepttest.

Systemet indeholder analoge, såvel som digitale signalbehandlingssystemer, for at overholde studieordningen for 6. semester, Signalbehandling.

12.2 Forbedringer af systemet

Der er naturligvis mange elementer i det konstruerede system der kan forbedres på den ene eller anden måde.

12.2.1 Forbedringer af hardware

Dette afsnit omhandler nogle forslag til ændringer af den anvendte hardware. Herunder hvordan systemet er opbygget med hensyn til EMC, SNR samt THD.

EMC-forhold

Alt det designede hardware er implementeret på "europakort" og de forskellige moduler og processer er forbundet via den bus, der findes på racket. Det betyder, at der er potentiale for støj på signalvejene.

Hvert af modulerne på "europakortene" i racket burde have en afkoblingskondensator til forsyningsspændingen, så der sikres en stabil DC-forsyning.

Stort set alle moduler og processer er wrappet op, hvilket er skyld i, at afkoblingskondensatorerne ikke er forbundet så tæt på stel som muligt. Forbedring af afkoblingskondensatorernes placering vil medføre en essentiel forbedring af hardwarens støjniveau.

Alle digitale kredsløb udsender magnetisk støj og følsomme overfor udefrakommende støj, så alle moduler med IC'ere burde skærmes med metalskærm. Det samme gælder kredsløbene med store modstande på 1MΩ.

Signal-forhold

Af hensyn til støj ville det være en god idé at bruge forholdsvis små modstande. Ydermere bør der bruges modstande med lille afvigelse.

Endvidere ville det være hensigtsmæssigt at anvende transistorkoblinger, istedet for operationsforstærkere på de steder, hvor der ikke er påkrævet meget store forstærkninger. Dette ville sandsynligvis udmunde i et større SNR og lavere THD.

Indgangstrin

Forstærkningen i indgangstrinnet burde sidde i det første led, altså spændingsfølgeren, for at minimere støjen i kredsløbet.

Equalizer

I målingerne og simuleringerne af equalizeren blev der registreret nogle peaks/dyk i forstærkningen omkring knækfrekvenserne. Dette skyldes det fasedrej der forekommer, når andenordens filtre bruges. For at afhjælpe dette problem kan der istedet anvendes førsteordens filtre. Ellers er det nødvendigt at bruge en anden slags filtre end Butterworth-filtre, f.eks. Thompson-filtre som har en meget lineær fase.

Forvrænger

Den designede forvrænger blev implementeret med to forskellige slags dioder til klipningen. Det ville naturligvis være bedre med flere forskellige dioder, som hver især har forskellige karakteristikker.

12.2.2 Optimering af programmel

I dette afsnit beskrives, hvilke forbedringer der kan foretages af det designede software.

Kompressor

Kompressorens forstærkningskoefficienter kunne, for forskellige forhold mellem styresignal og threshold, tabellægges. Derved ville fåes en effekt som er i bedre overensstemmelse med det matematiske princip.

For at få en bedre kompressor-lyd er det i henhold til kilden [6] en god idé, at opdele kompressoren i bånd. Det vil medføre forskellig attack/release-tid for forskellige frekvensbånd, hvilket giver en bedre dynamik. Derfor vil en opdeling i flere bånd, med stejlere filtre være en mulig forbedring.

Desuden vil det være godt at kunne regulere koefficienten,λ, som er detectorens op-/afladningstid.

Rumklang

Den designede rumklang har en anelse metallisk lyd. Dette kan afhjælpes ved at indsætte et FIR-filter på udgangen. Dette ville også svare mere til impulsresponsen for et rum, der dæmper høje frekvenser mere end lave. Den designede "Schroeder reverberator" dæmper ikke de høje frekvenser.

Flanger

Flangerens sinus kunne eventuelt tabellægges med en større opløsning. Derved vil signalet komme til at lyde mere "rent". Det ville også være anvendeligt at kunne justere amplituden af den sinus, som flangereffekten moduleres med.

12.3 Overholdelse af projektenhedbeskrivelsen

I henhold til projektenhedsbeskrivelsen for 6. semester i signalbehandling, er det ene delmål for semesteret:

- *At sætte den studerende i stand til ud fra applikationsmæssige specifikationer at foretage syntese af et realtids analogt/digitalt signalbehandlingssystem.*

Efter endt projektenhed mener projektgruppen, at have indlært evner til at foretage en analyse på syntheseniveau af et realtids analogt/digitalt signalbehandlingssystem. Denne kundskab er opnået gennem design og konstruktionen af et analogt/digitalt realtidsystem. Gennem designfasen har projektgruppen fået erfaring med assembler-programmering af en DSP (TMS320C26) ved brug af en debugger/monitor. Det andet delmål for semesteret er:

- *At sætte den studerende i stand til at anvende teorier og metoder, som benyttes ved design, realisation og test af digitale signalbehandlingsalgoritmer samt software-implementation af disse på programmerbare digitale signalprocessorer.*

Disse evner har projektgruppen opnået gennem arbejde med SPU-modellen, som beskæftiger sig med designmetoder for programmer.

Programmel er optimeret ved at placere tidsmæssigt ukritiske procedurer i main-programmet, som kun afvikles, når det ikke afvikles interrupt-rutiner. Tidskritiske procedurer afvikles i interrupt-rutiner, således at det er muligt at overholde kravet om realtidsbehandling.

12.3.1 Anvendelse af PE-kurser

I projektet er følgende kurser anvendt til design/dokumentation:

- Analoge filtre.
- Realisation af signalbehandlingssystemer.
- Analog og digital regneteknik.

Kurset, Analoge filtre er blevet brugt til design af equalizeren, hvor der er foretaget beregninger af delefrekvenser, afskæringsstejlhed, nødvendig filterorden osv. Kurset i realisation af signalbehandlingssystemer er anvendt i flere dele af systemet, eksempelvis under opkobling af DSP, design af digitale filtre, skalering af koefficienter osv. Kurset om analog og digital regneteknik er blevet brugt i projektet, bl.a. til beregning af støj i softwareimplementeringerne og støj i indgangstrinnet.

Det kan dermed konkluderes at alle PE-kurser for 6.semester på signalbehandlingslinien, pånær kurset om signalers oprindelse og registrering er anvendt i dette projekt.

12.4 Opsummering

Der er i dette projekt blevet designet og implementeret effektenheder til en el-guitar. Der er lavet accepttest af det samlede system og det er blevt vurderet, hvorvidt de opstillede krav i kravspecifikationen er blevet overholdt. Det blev vurderet at accepttestens udfald var tilfredsstillende. Om de forskellige effekter gælder følgende:

- Equalizeren er blevet realiseret som en 3 bånds grafisk equalizer og er testet til at virke efter hensigten.
- Forvrængeren er ligesom equalizer realiseret analogt. Denne effekt virker som forventet.
- Kompressoren er realiseret digitalt som software på det designede DSP-system. Denne effekt overholder kravspecifikationen.
- Flangereffekten er ligeledes realiseret med software. Denne effektenhed lever ligeledes op til det forventede.
- Rumklangeffekten er implementeres software-mæssigt og den virker også efter hensigten.

Konklusionen på projektet er derfor, at der gennem projektforløbet er opnået det ønskede resultat: PE-kurserne er involveret i projektet, kravspecifikationen er overholdt og accepttesten er blevet udført med tilfredsstillende succes.

Litteratur

[1] Institut for elektroniske systemer AAU. *Noter til SB6/2000: Realisation af Signalbehandlings systemer.* AAU, 2000. ISBN NAN.

[2] Finn Haugen. *Regulering av dynamiske systemer.* Tapir forlag, 1994. ISBN 82-519-1433-7.

[3] Josef Holecek. *Lær at spille guitar.* AB Carl Gehrmans Musikforlag, 1975. ISBN Ingen.

[4] Lawrence P. Huelsman. *Active and passive analog filter design.* McGraw-Hill, 1993. ISBN 0-07-112519-1.

[5] Stephen Biering-Sørensen. Finn Overgaard Hansen. Susanne Klim. Preben Thalund Madsen. *Håndbog i Struktureret Programudvikling.* Teknisk Forlag A/S, 1998. ISBN 87-571-1046-8.

[6] D. Rosenberg & G.W. Hoffman M.G. Duncan. *Design Criteria of a Universal Compander for the Elimination of Audible Noise in Tape, Disc, and Broadcast Systems.* Audio Engineering Society, New York, 1974. ISBN NAN.

[7] Sofus Birkedal Nielsen. *Debugger/Monitor Systems, For the Digital Signal Processor families TMS320C/2X//2XX/5X.* Department of Acoustics, 2000. ISSN 09081224.

[8] Eddy Bøgh Brixen og Jan Voetman. *Praktisk Elektroakustik.* Teknisk Forlag A/S, 1989. ISBN 87-571-0985-0.

[9] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing 2nd Edition.* Prentice Hall, 1999. ISBN 0-13-083443-2.

[10] Sophocles J. Orfanidis. *Introduktion To Signal Processing.* Prentice Hall International Edition, 1996. ISBN 0-13-240334-X.

[11] Adel S. Sedra/Kenneth C. Smith. *Microelectronic Circuits - Fourth Edition.* Oxford, 1998. ISBN 0-19-511690-9.

[12] Keith Spence. *Bogen om musik*. Munksgaard, 1995. ISBN 87-16-11590-2.

[13] Microprocessor Development Systems. *TMS320C2x Fixed-Point DSP User's Guide*. Texas Instruments, 1993. ISBN SPRU014C.

[14] *WWW link*. http://www.eden.com, 27. Marts 2001.

[15] *WWW link*. http://www.chariot.net.au/ gmarts/ampovdrv.htm, 10. Maj 2001.

[16] *WWW link*. http://www.cs.ust.hk/faculty/layers/comp342/effects/comb_rev.html, 22. April 2001.

APPENDIKS A

Målerapport for guitar

Formålet med dette appendiks er at fastlægge den benyttede guitars frekvensområde, samt at finde udgangsimpedansen og udgangsspændingen. Dette gøres, for at kunne lave den rette signaltilpasning af signalet fra guitaren. Endvidere vil guitarens signal/støj forhold (SNR) blive beregnet.

A.1 Øvre og Nedre grænsefrekvens samt SNR

Formålet med denne måling er at finde guitarens øvre og nedre grænsefrekvens samt signal/støj forholdet.

A.1.1 Forsøgsbeskrivelse

For at finde den laveste og højeste frekvens foretages der optagelser af den dybeste og højeste tone som guitaren kan spille. Disse to toner optages via lydkortet i 16 bit og f_s=44100Hz. Derefter foretages en FFT-analyse via MATLAB og frekvensspektret plottes. Udfra dette er det muligt at se guitarens frekvensområde.

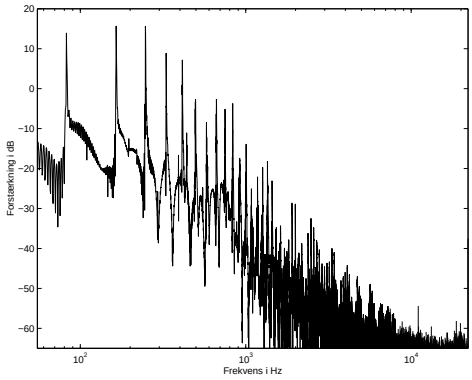
Følgende udstyr blev benyttet under målingerne:

- Guitar: Axtech Vintage Plus VS-571.
- Lydkort: 16-bit ESS Audiodrive ES1938/1946 Version 4.05.22.
- Oscilloskop: Agilent 54621A.
- Power supply: B&O SN16.

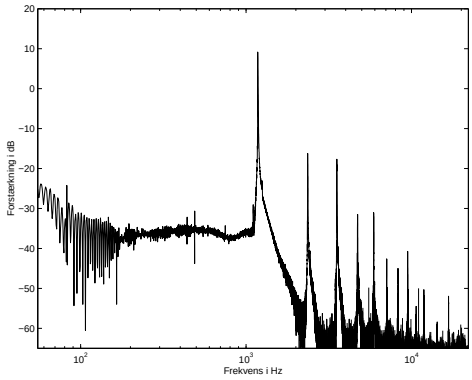
- Ohmmeter: Fluke 37 Multimeter.
- Sinusgenerator: Philips PM 5192.

A.1.2 Måleresultater

Målingerne blev foretages som nævnt i forsøgsbeskrivelsen og FFT-analyserne af frekvensspektrene er vist på figur A.1 og A.2.



Figur A.1: *FFT af den laveste frekvens på guitaren.*



Figur A.2: *FFT af den højeste frekvens på guitaren.*

Af figurene ses, at den dybeste tone ligger ved ca. 82Hz. Endvidere ses, at der sjældent forekommer frekvenser over ca. 11kHz, og at de er dæmpet med 50 dB.

Signal/støj forhold
Signal/støj forholdet er aflæst som "best case" (ud fra figur A.2), da det designede system skal have et signal/støj forhold, der altid er bedre end guitarens. Den er aflæst ved den højeste tone, hvor SNR er aflæst:
$A_{db_{max}} = 10dB$ Aflæst på figur A.2
$A_{db_{min}} = -50dB$ Aflæst på figur A.2

$$SNR = 20 \cdot log\left(\frac{A_{max}}{A_{min}}\right) = 20 \cdot log\left(\frac{3,16}{0,00316}\right) = 60dB \tag{A.1}$$

A.2 Måling af udgangsspænding

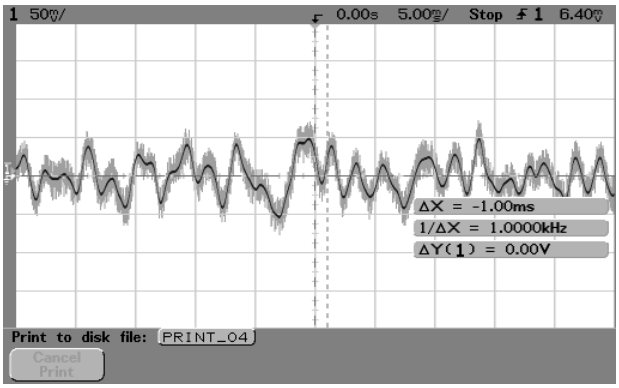
A.2.1 Formål

Formålet med målingen er, at fastlægge guitarens udgangsspænding. Denne undersøges både for et normalt anslag og et kraftigt anslag.

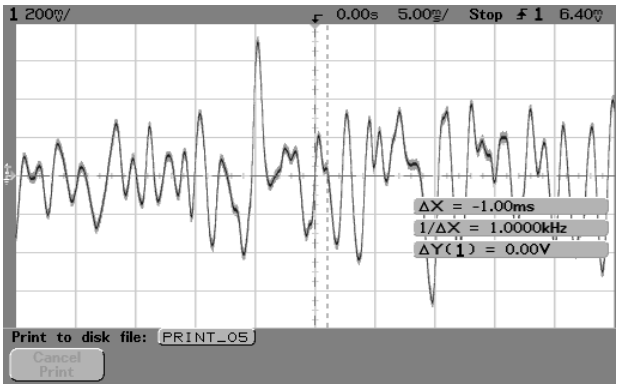
A.2.2 Forsøgsbeskrivelse

For at måle spændingen på guitarens udgang, kobles guitarens udgang til et oscilloskop. Der foretages et normalt og et kraftigt anslag. Efter hvert anslag måles amplituden af signalet.

A.2.3 Måleresultater



Figur A.3: *Udgangsspænding ved normalt anslag.*



Figur A.4: *Udgangsspænding ved kraftigt anslag.*

Udfra figur A.3 ses, at udgangsspændingen ved normalt anslag ligger på ca. 0,1V_{pp} og figur A.4 viser, at udgangsspændingen ved kraftigt anslag ligger på ca. 1,4V_{pp}

A.3 Måling af udgangsimpedans

A.3.1 Formål

Formålet med denne måling er, at finde udgangsimpedansen for guitaren. Dette er for at kunne tilpasse impedansforhold i henhold til næste trin.

A.3.2 Forsøgsbeskrivelse

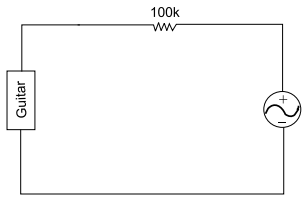
For at finde guitarens udgangsimpedans foretages målinger fra DC til ca. 15kHz. Målingerne foretages ved at sætte DC og forskellige frekvenser på guitarens udgang i serie med en modstand på 100kΩ. Vha. loven om spændingsdeling kan guitarens indgangsmodstand beregnes:

$$V_{guitar} = V_{in} \cdot \frac{R_{100k\Omega} + R_{guitar}}{R_{guitar}} \tag{A.2}$$

$$\Updownarrow$$

$$R_{guitar} = \frac{R_{100k\Omega} \cdot V_{in}}{V_{guitar} - V_{in}} \tag{A.3}$$

Af figur A.5 ses forsøgsopstillingen for måling af guitarens udgangsimpedans.

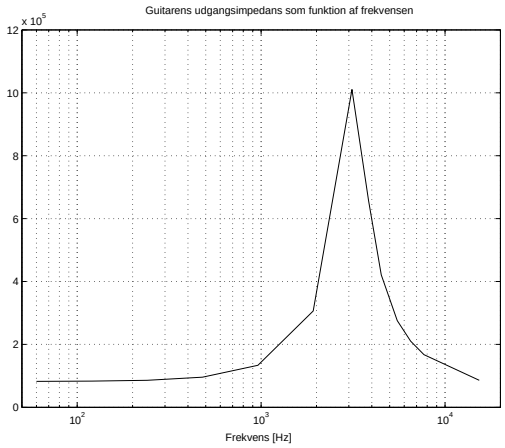


Figur A.5: *Forsøgsopstilling for måling af impedans for forskellige frekvenser.*

A.3.3 Måleresultater

Ved DC blev indgangsimpedansen målt til 6,8kΩ og ved frekvenser op til 15kHz blev impedansen som på figur A.6 på den følgende side

Det ses at der er en impedans-peak omkring 3,12kHz. Dette skyldes formentlig en RLC-resonator, bestående af pickup'ens selvinduktans, guitarens indgangsmodstand og kablets kapacitans.



Figur A.6: *Udgangsimpedansen på guitaren for forskellige frekvenser.*

APPENDIKS B

Måling af diodekarakteristikker

Dette appendiks er et tillæg lavet til afsnittet om forvrængeren. Dette undersøger forskellige diodetyper med henblik på anvendelse i forvrængermodulet. De undersøgte dioder er: silicium-, schottky-, lys- og germaniumdioder.

B.1 Diode

B.1.1 Ulineariteten i en diode

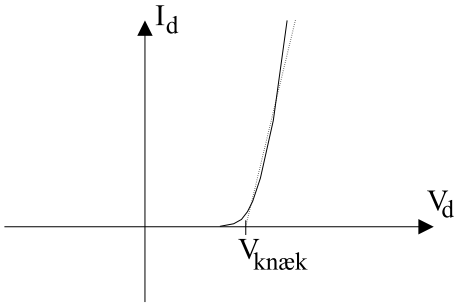
En diodes karakteristik for diodestrømmen som funktion af diodespændingen, ses som en eksponentialfunktion på figur B.1 på næste side med nedenstående diodeligning:

$$I_D = I_s \cdot e^{\frac{V_D}{n \cdot V_T}} \tag{B.1}$$

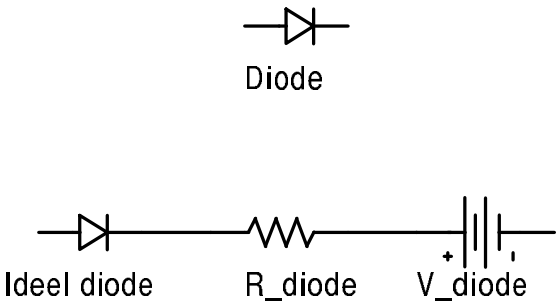
På figuren ses også den tilnærmede funktion som kan ækvivaleres med en ideel diode, en diodespænding og en modstand, se figur B.2 på den følgende side.
Den ideelle diode-ensretter, og diodespændingen er det spændingsfald der er over dioden når den leder. Modstanden svarer til resistansen i bl.a. tillederne på dioden, samt urenhed i doteringen.

B.1.2 Målinger af forskellige diodetyper

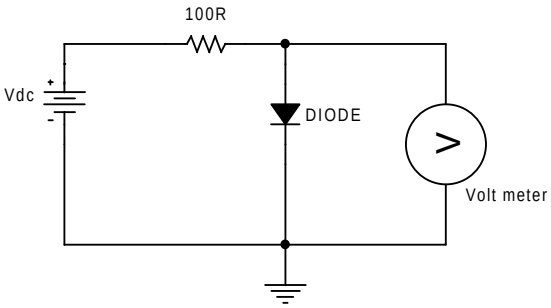
Der vælges fire typer dioder: en silicium, en schottky, en lysdiode og en germanium. Der ønskes at måle karakteristikkerne for dioderne for at finde den spænding og diodemodstand dioden kan ækvivaleres med. Til at måle karakteristikken benyttes måleopstilling som ses på figur B.3 på næste side.



Figur B.1: *Diodekarakteristik med tilnærmet funktion.*



Figur B.2: *Ækvivalentdiagram for diode*



Figur B.3: *Opstilling til måling af diodekarakteristik*

Spændingen (V_{dc}) varieres fra 0 til 5V. Spændingen over modstanden måles, og derved kan strømmen i dioden og spændingen over dioden findes.

$$I_D = \frac{V_{100\Omega}}{R_{100\Omega}} \tag{B.2}$$

$$V_D = V_{dc} - V_{100\Omega} \tag{B.3}$$

I laboratoriet er karakteristikkerne blevet målt, og resultaterne ses på figurerne B.4 på side 110 til B.6 på side 110.

Ud fra simuleringerne kan diodespændingen og diodemodstanden aflæses.

$$\frac{1}{R_{diode}} = \frac{\Delta I_d}{\Delta V_d} \Leftrightarrow R_{diode} = \frac{\Delta V_d}{\Delta I_d} \tag{B.4}$$

Derved fåes følgende modstandsværdier:

$$R_{1N4148} = \frac{0,86 - 0,751}{0,0518 - 0,01252} = 2,77\Omega \tag{B.5}$$

$$R_{BAT85} = \frac{0,4 - 0,35}{0,033 - 0,020} = 3,8\Omega \tag{B.6}$$

$$R_{LED} = \frac{2,5 - 2,0}{0,050 - 0,008} = 11,9\Omega \tag{B.7}$$

$$R_{AA119} = \frac{0.56 - 0,41}{0,0024 - 0,0012} = 129.79\Omega \tag{B.8}$$

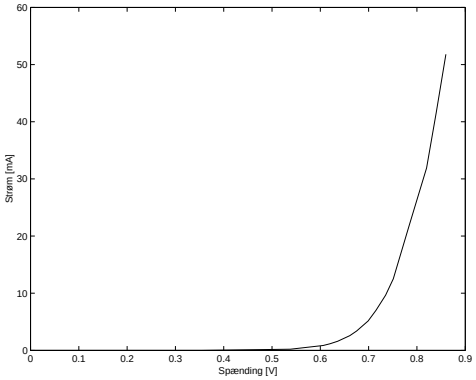
Endvidere kan diode spændingerne beregnes til:

$$V_{1N4148} = 0,643V$$

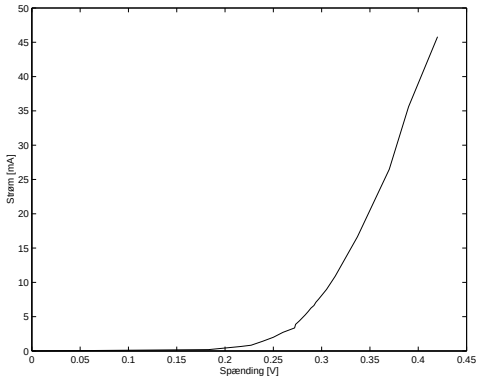
$$V_{BAT85} = 0,273V$$

$$V_{LED} = 1,9V$$

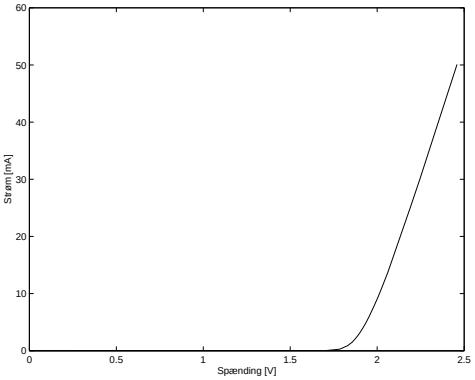
$$V_{AA119} = 0.25V$$



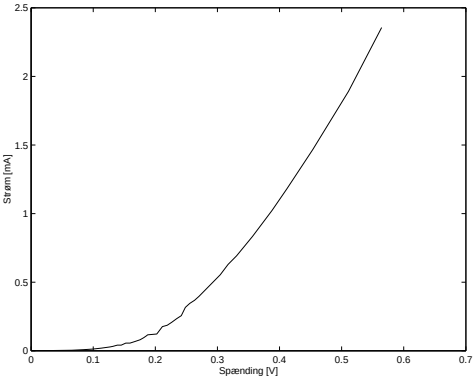
Figur B.4: *Målt diodekarakteristik for 1N4148*



Figur B.5: *Målt diodekarakteristik for BAT85*



Figur B.6: *Målt diodekarakteristik for LED*



Figur B.7: *Målt diodekarakteristik for AA119*

APPENDIKS C

Testrapport

Dette appendiks omhandler de simuleringer og målinger, der foretages på systemet for at eftervise at det enkelte modul virker som forventet. Først simuleres og testes de analoge moduler, hvorefter de digitale moduler simuleres og testes.

Appendikset er opbygget på følgende måde:

- Modul 1: Equalizer.
- Modul 2: Forvrænger.
- Modul 3: Codec/indgangstrin.
- Modul 4: Kompressor.
- Modul 5: Flanger.
- Modul 6: Rumklang.

C.1 Simulering/test af modul 1: Equalizer

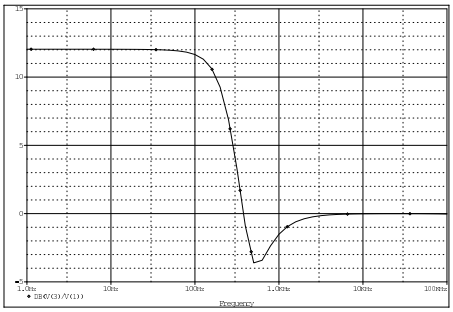
Formålet med dette afsnit er, at eftervise at equalizeren virker efter hensigten. For at gøre dette benyttes simuleringer samt målinger på equalizeren i laboratioriet.

C.1.1 Simulering af equalizer

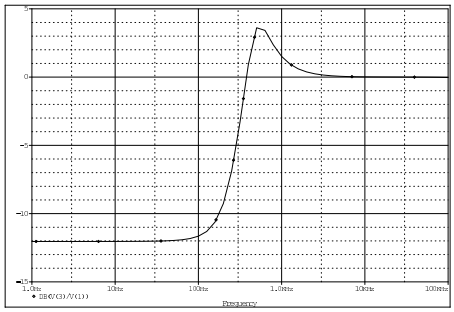
Simuleringerne er foretaget i Pspice, og det er valgt at vise følgende simuleringer:

- Baskontrol m. 12dB hævning.
- Baskontrol m. 12dB sænkning.
- Mellemtonekontrol m. 12dB hævning.
- Mellemtonekontrol m. 12dB hævning.
- Diskantkontrol m. 12dB hævning.
- Diskantkontrol m. 12dB sænkning.

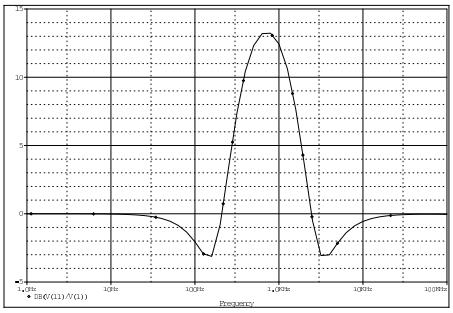
cir-filerne, som simuleringerne er lavet udfra, findes på den vedlagte CD-ROM under equalizer. Herunder følger resultatet af simuleringerne.



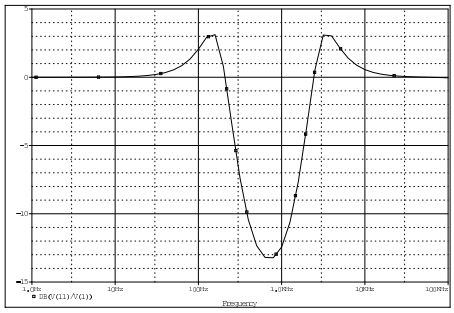
Figur C.1: *Diskantkontrol m. 12dB's hævning.*



Figur C.2: *Diskantkontrol m. 12dB's sænkning.*

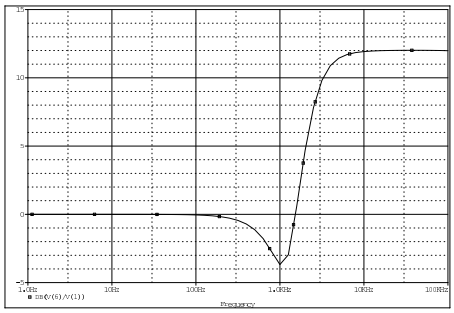


Figur C.3: *Mellemtonekontrol m. 12dB's hævning.*

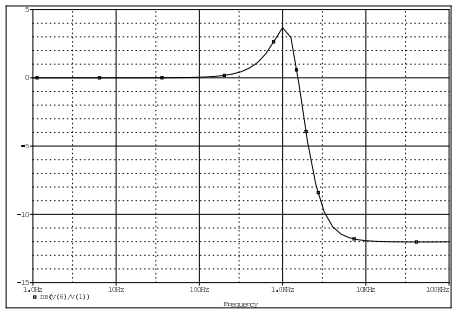


Figur C.4: *Mellemtonekontrol m. 12dB's sænkning.*

Som det fremgår af figurerne, er der inden for pasbåndet den ønskede dæmpning. Dog er der for alle tre bånd nogle over- og undersving pga. problemer med fasedrej omkring afskæringsfrekvenserne. Alt i alt konkluderes at simuleringen viser et tilfredsstillende resultat.



Figur C.5: *Baskontrol m. 12dB's hævning.*



Figur C.6: *Baskontrol m. 12dB's sænkning.*

C.1.2 Målinger på equalizer

Formålet med laboratoriemålingerne er, at eftervise at equalizeren virker i praksis, samt at måle om de opstillede krav er blevet opfyldt.

Testbeskrivelse

De målinger der foretages, er målinger, der viser frekvensresponsen, for de benyttede filtre. Dette er både hævning og sænkning af bas-, mellemtone- og diskantkontrol. Ud over dette skal signal/støj forholdet (SNR) og total harmonisk forvrængning (THD) måles. For at måle frekvensresponsen bruges en funktionsgenerator og amplituden aflæses på et oscilloskop.

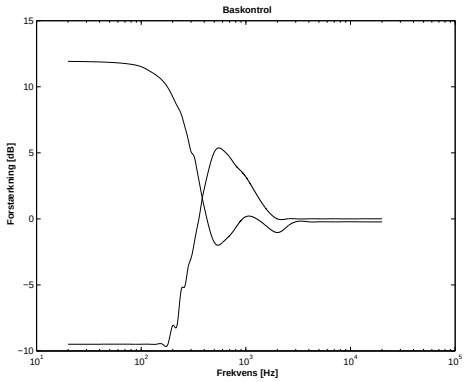
Det benyttede apparatur

- Funktionsgenerator: Philips PM5131 (LBNR 08088)
- Strømforsyning: Hameg HM7042 (LBNR 33877)
- Oscilloskop: Agilent 54621A (LBNR 33846)

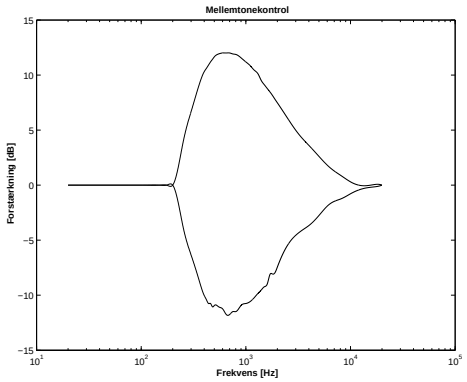
Testresultater

Efter endt test blev frekvensresponsen for filtrene plottet. Disse ses på figurerne C.7 på den følgende side,C.8 og C.9. Målepunkterne kan ses på CD-ROM under equalizer.

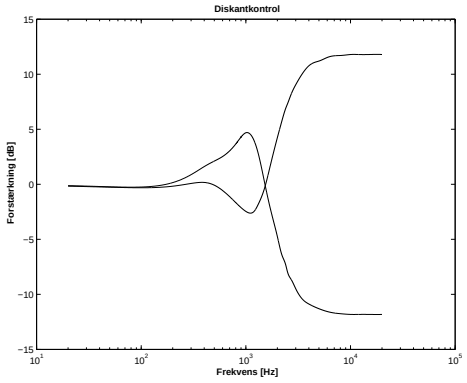
Som det ses af figurerne stemmer simuleringerne og målingerne overens, der skal dog bemærkes at der er målefejl i sænkningen af baskontrollen, hvorfor der ikke sænkes helt med 12dB.



Figur C.7: *Målinger for baskontrol*



Figur C.8: *Målinger for mellemtonekontrol*



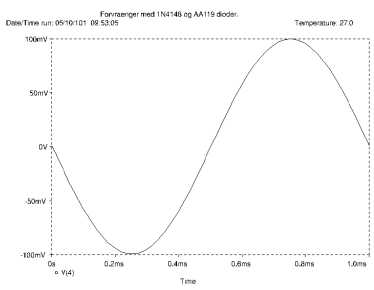
Figur C.9: *Målinger for diskantkontrol*

C.2 Simuleringer/målinger modul 2: Forvrængeren

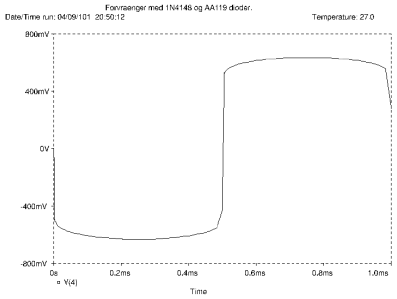
Der ønskes at få verificeret om forvrængeren virker, og overholder de i kravspecifikationen opstillede krav. Dette gøres vha. simuleringer og laboratoriemålinger.

C.2.1 Simulering af forvrænger

Det samlede forvrængermodul, uden udgangstrin, er simuleret i Spice. Der påtrykkes en 1kHz sinustone med en amplitude på henholdsvis $0{,}1V_p$ og $2{,}5V_p$. Indgangsmodstanden varieres mellem minimum (499Ω) og maksimum (220kΩ), og der simuleres med 1N4148 som forvrængerdiode. Outputtet aflæses efter forvrængerenheden.
Resultatet af simuleringerne ses på figurerne C.10 og C.11.



Figur C.10: *1N4148 med $0{,}1V_p$ ind og 1 ganges forstærkning.*



Figur C.11: *1N4148 med $2{,}5V_p$ ind og 440 ganges forstærkning.*

Den anvendte CIR-fil til simuleringen findes på CD-ROM'en under forvrænger.
Konklusionen på simuleringerne er, at forvrængeren virker som forventet.

C.2.2 Målinger på forvrænger

Formålet med laboratoriemålingerne er, at eftervise at forvrængeren virker i praksis, samt at måle om de opstillede krav er blevet opfyldt.

Testbeskrivelse

Der foretages en række målinger, som skal vise forvrængerens signalbehandling, ved de forskellige indstillinger. Først måles udgangssignalet ved de forskellige forvrængningstyper, med en sinus med en amplitude på henholdsvis $0{,}1V_p$ og $2{,}5V_p$ og en frekvens på 1kHz. Målingerne foretages med maksimal forvrængningsgrad, og ses på figurene C.12 til C.17.
Målingerne optages til en wav-fil (findes på CD-ROM'en). Der foretages en FFT-analyse på wav-filen, for at se frekvensindholdet i målingerne. Disse analyser ses på figurene C.18 på side 118 til C.21 på side 119. Desuden bruges FFT-analyserne også til at finde signal/støj forholdet.

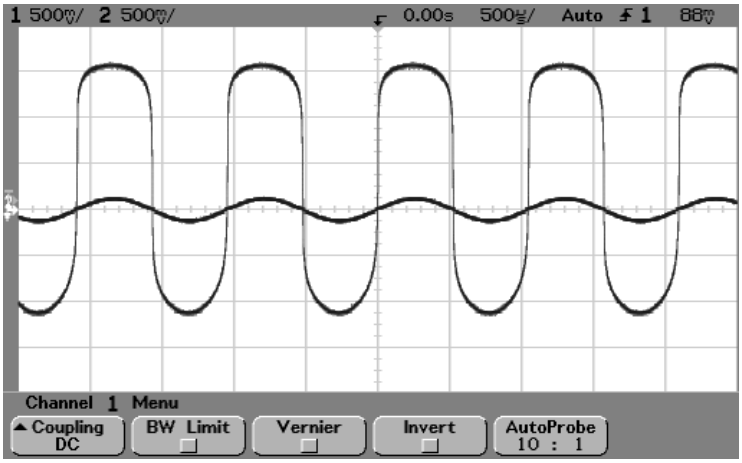
Måleudstyr

Til målingerne er anvendt følgende måleudstyr:

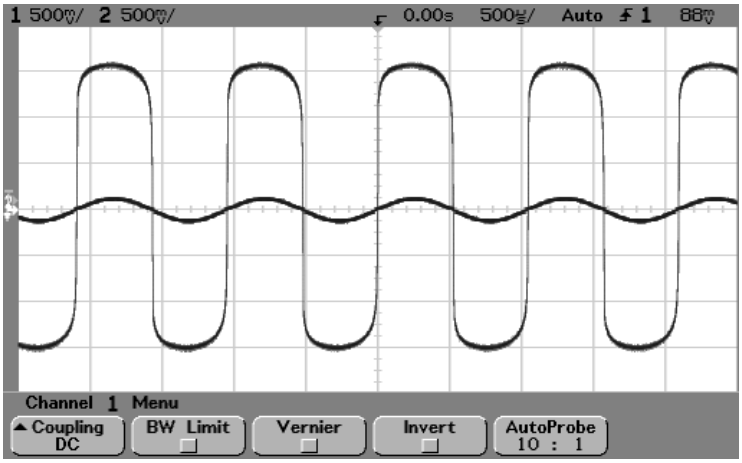
- Funktionsgenerator: Philips PM5131 (LBNR 08088)
- Strømforsyning: Hameg HM7042 (LBNR 33877)
- Oscilloskop: Agilent 54621A (LBNR 33846)
- Lydkort: Siemens-Fujitsu sound MAX

Testresultater

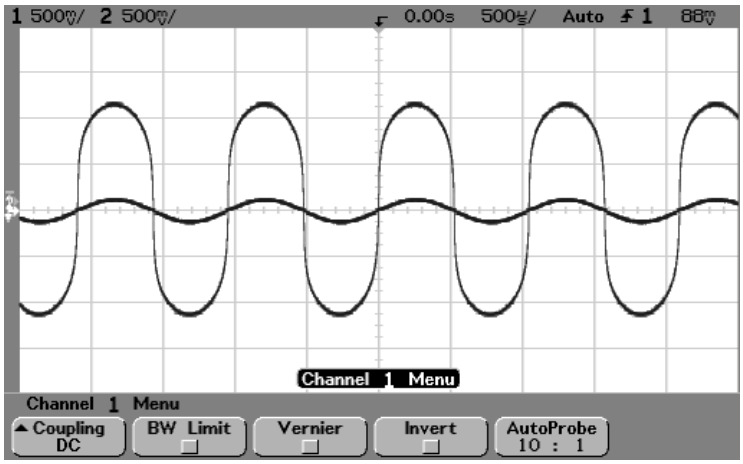
Resultatet af målingerne ses på følgende figurer(figur C.12 til C.17 på side 118).



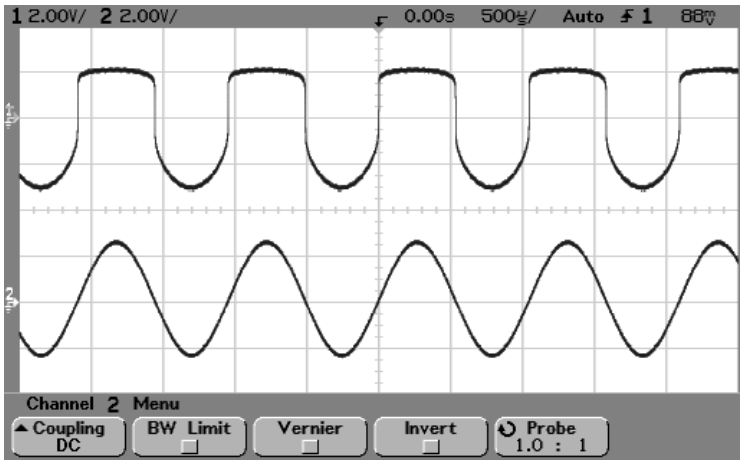
Figur C.12: *Asymmetrisk forvrængning med 0,1V_p ind og 440 ganges forstærkning.*



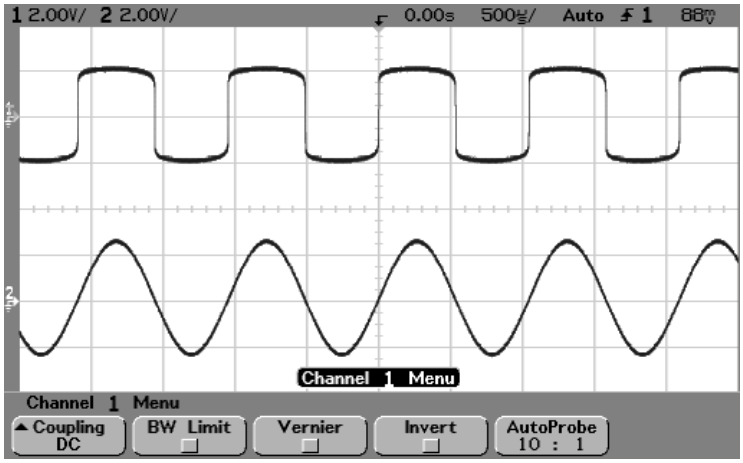
Figur C.13: *Symmetrisk forvrængning med 0,1V_p ind og 440 ganges forstærkning (diodetype: 1N4148).*



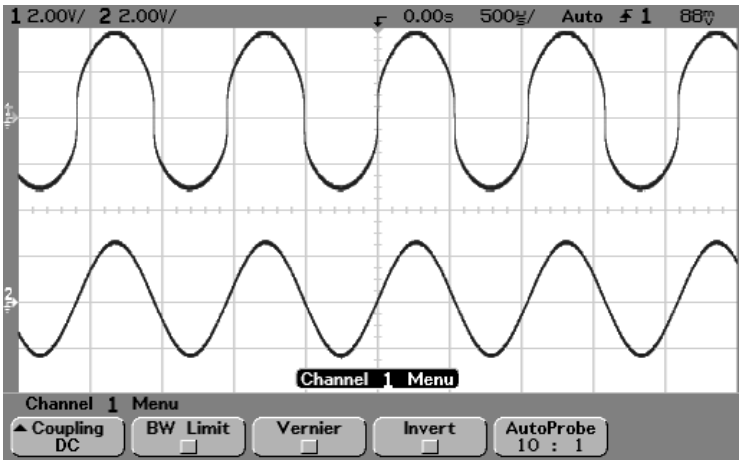
Figur C.14: *Symmetrisk forvrængning med 0,1V_p ind og 440 ganges forstærkning (diodetype: AA119).*



Figur C.15: *Asymmetrisk forvrængning med 2,5V_p ind og 440 ganges forstærkning.*

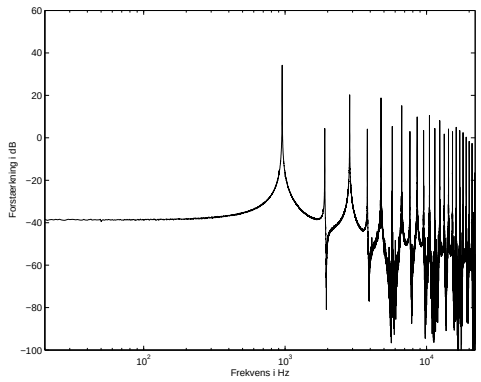


Figur C.16: *Symmetrisk forvrængning med 2,5V_p ind og 440 ganges forstærkning (diodetype: 1N4148).*

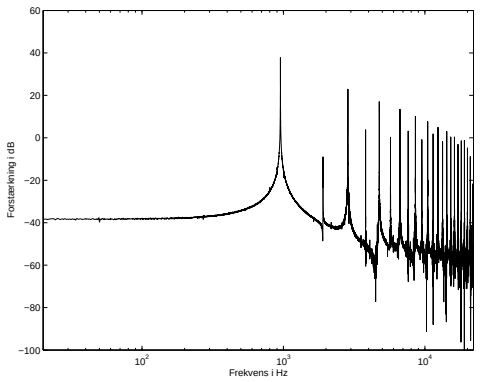


Figur C.17: *Symmetrisk forvrængning med 2,5V_p ind og 440 ganges forstærkning (diodetype: AA119).*

På figurene C.18 til C.21 på modstående side ses resultatet af FFT-analyserne for de 3 forvrængningstyper, med 0,1V_p sinus og 1kHz ind, med maksimal forvrængning.



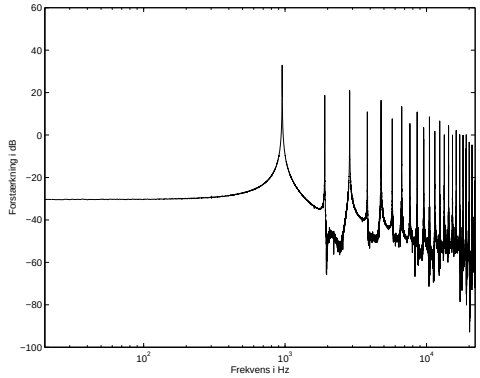
Figur C.18: *FFT, symmetrisk med 2,5 V_p ind og 440 ganges forstærkning (diodetype: 1N4148).*



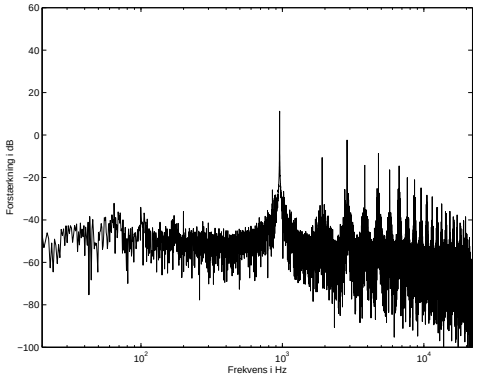
Figur C.19: *FFT, symmetrisk med 2,5 V_p ind og 440 ganges forstærkning (diodetype: AA119).*

Som det ses på figurerne C.18 til C.21 på modstående side forstærkes overtonerne forskelligt, alt afhængig af diodetypen. Signal/støjforholdet kan aflæses på FFT-analysen på figur C.18 til C.21 på modstående side, hvor det sammenlignes mellem de 3 klipningstyper.
Signal/støjforholdet aflæses ud fra FFT-analyserne, og ses i følgende tabel:

Type	Amplitude	Grad	SNR
AA119	2,5 V_p	hård	70 dB
1n4148	2,5 V_p	hård	70 dB
asym	2,5 V_p	hård	60 dB
asym	0,1 V_p	blød	60 dB



Figur C.20: *FFT, Asymmetrisk med 2,5 V_p ind og 440 ganges forstærkning.*



Figur C.21: *FFT, Asymmetrisk med 0,1 V_p ind og 5 ganges forstærkning.*

Opsamling

Målingerne viser, at forvrængeren også fungerer i praksis. Den blødeste forvrængning opnås ved AA119 germaniumsdioderne, hvor 1N4148 giver den hårdeste forvrængning. Signal/støjforholdet er målt til mindst 60dB, hvilket overholder kravet fra kravspecifikationen.

C.3 Simuleringer/målinger på modul 3: Codec/indgangstrin

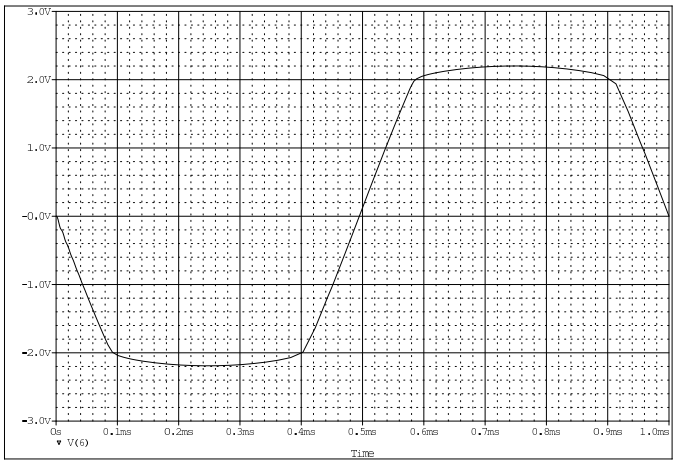
Formålet med denne test er at verificere at indgangstrinnet virker efter hensigten, samt at den overholder de opstillede krav, foretages der målinger og simuleringer af trinnet. Testen er delt op i 2 dele: En del, der tester den analoge hardware og en del, der tester selve AD/DA konverteringen.

C.3.1 Simulering af indgangstrin

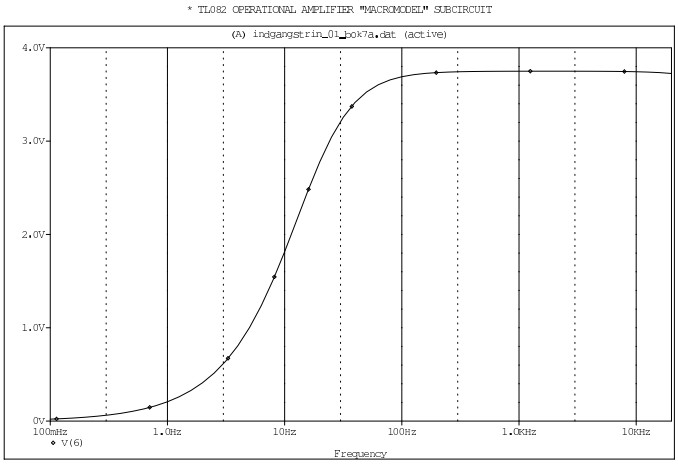
Simuleringen af systemet foretages på det analoge system (indgangstrinnet). Resultatet af simuleringerne ses på figur C.22 på næste side og C.23 på den følgende side, hvor henholdsvis transientresponsen og frekvensresponsen er simuleret.

Som det kan ses af figur C.23 på næste side er der en højpasfiltrering omkring 16Hz, hvor amplituden er ca 0,7·3,57V. Derudover er der en forstærkning på ca 3,57V/V i resten af frekvensspektret, bortset fra ca. 500kHz og opefter, hvor der er indskudt en kondensator, eftersom simuleringen viste en grim peak i forstærkningen omkring dette punkt. I laboratoriet viste denne peak sig også men ikke i nær så høj grad. Det formodes at den skyldes en form for resonans pga. diodernes spole/kapacitive egenskaber.

Den anvendte CIR-fil til simuleringen findes på CD-ROM'en under codec. Som det ses af simuleringen virker systemet efter hensigten.



Figur C.22: *Udgangssignalet for en frekvens på 1 kHz med 1,4V_{pp} på indgangen.*



Figur C.23: *Frekvensresponsen for indgangstrinnet.*

C.3.2 Målinger på analog hardware i indgangstrinet

Formålet med laboratoriemålingerne er, at dokumentere, at den benyttede hardware fungerer efter hensigten, samt at det opfylder de opstillede krav.

Testbeskrivelse

Først testes om indgangsforstærkeren klipper som ønsket. Ved at sætte et sinussignal på indgangen kontrolleres både at forstærkningen på 3,57 passer samt at signalet klippes ved spændinger over $\pm$2,5V.

Dernæst måles signal/støj forhold (SNR) og total harmonisk forvrængning (THD).

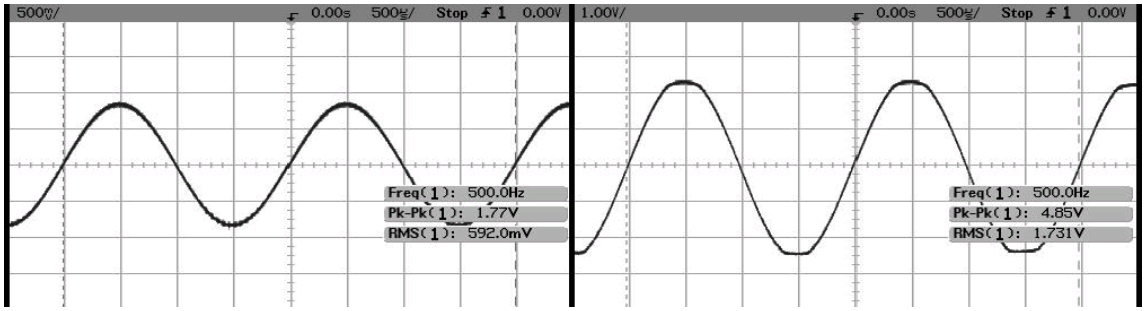
Måleudstyr

Følgende udstyr er benyttet under målingerne:

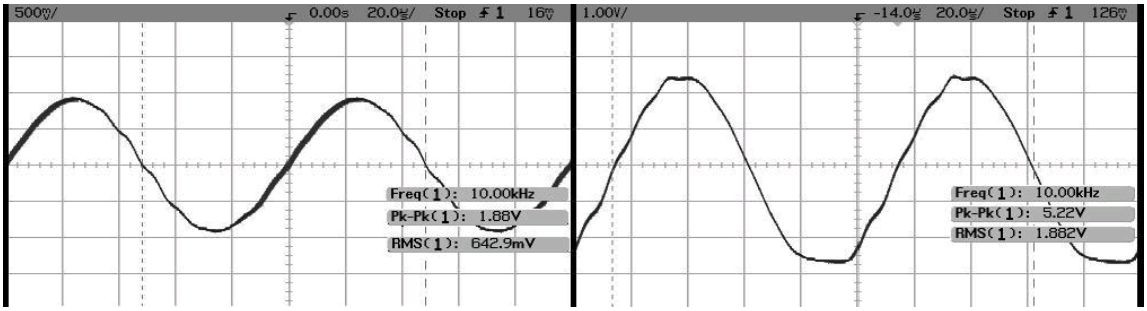
- Philips PM5138 Function Generator.
- Agilent 54621A Oscilloscope.
- Hameg Triple Power Supply HM7042.

Testresultater af analog hardware

For at teste om indgangstrinnet klippede som ønsket, blev der sat et sinussignal på indgangen og testet om spændingssignalet på udgangen var forstærket med en faktor 3.57 og klippet, hvis nødvendigt. Der blev målt for to forskellige frekvenser, 500 Hz og 10 kHz. Indgangssignalernes spændingsniveauer blev sat til henholdsvis 0,5 og 1,5 V_{pp}. Figur C.24 og C.25 viser, hvorledes signalet klippes ved de forskellige indgangssignaler:



Figur C.24: *Udgangssignalet med frekvens på 500Hz og indgangsspænding på hhv. 0,5 og 1,5V_{pp}.*



Figur C.25: *Udgangssignalet med frekvens på 10kHz og indgangsspænding på hhv. 0,5 og 1,5V_{pp}.*

SNR og THD er målt til følgende:

- SNR = 66dB.
- THD = 0,4%.

Da SNR og THD overholder kravene og på baggrund af målingerne ses det, at systemet virker efter hensigten.

C.3.3 Målinger på codec

Denne test foretages for at sikre at AD/DA konverteren virker efter hensigten. Ydermere skal THD og SNR måles. For at foretage denne test udføres en "loop-test". Det betyder, at det ben, hvor det digitale output til DSP'en (pin 43: SDOUT) sendes på, forbindes til det ben hvor det digitale output fra DSP'en (pin 42: SDIN) modtages. Princippet er, at det signal, som sendes ind på pin 27 (LIN1: analog input), kan aflæses på pin 16, som er det analoge output. Codec'en testes under følgende forhold:

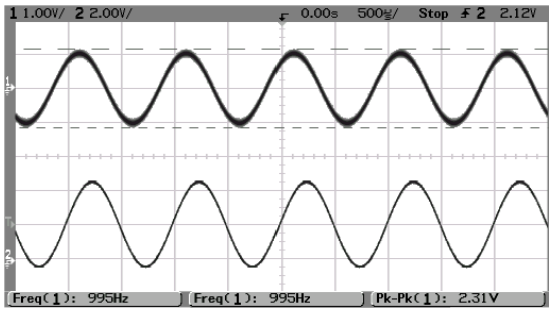
Test	Inputfrekvens [Hz]	Inputamplitude [V_{pp}]	Outputfrekvens	Outputamplitude [V_{pp}]
1	100Hz	1	100Hz	600 mV
2	10kHz	1	10kHz	490 mV
3	100Hz	5	100Hz	2.38 V
4	10kHz	5	10kHz	1.99 V

Tabel C.1: *måleresultater for test af codec.*

Testresultater af AD/DA konvertering

Der blev registreret 10MHz-støj på outputsignalet, hvilket sandsynligvis skyldes at codec'en er wrappet op/dårlige EMC-forhold. Det betyder, formentligt at afkoblingskondensatorerne ikke er placeret tæt nok på stel og codec. Endvidere ses det at amplituden ikke er som forventet på udgangen. Dette skyldes ikke opkoblingen af codec'en, men selve IC'en.

Signal/støj forhold blev målt til 71dB, og total harmonisk forvrængning til 0.3%, hvilke opfylder de opstillede krav. Det vurderes ud fra testen og de tilhørende måleresultater, som ses på figur C.26 at codec'en har en tilfredsstillende funktionalitet.



Figur C.26: *Måleresultater for test af AD/DA konverter. Øverst ses outputsignalet og nederst inputsignalet.*

Opsamling

Sammenlignes simuleringen af indgangstrinnet (figur C.22 på side 120) med målingerne ses det, at de målte kurver stemmer godt overens med det simulerede, hvorfor det konkluderes, at klipningen fungerer efter hensigten. AD/DA konverteringen foretages også

korrekt og opfylder de opstillede krav.

Det kan dermed konkluderes, at codec'en fungerer efter hensigten.

C.4 Simulering/test af modul 4: Kompressor

Som tidligere nævnt 8.3 på side 58 er princippet for en kompressor simuleret i Matlab. Det er gjort udfra [[10], s. 384-388] . Tillige er de to beregninger af forstærkningskoefficienter til kompressoren simuleret.

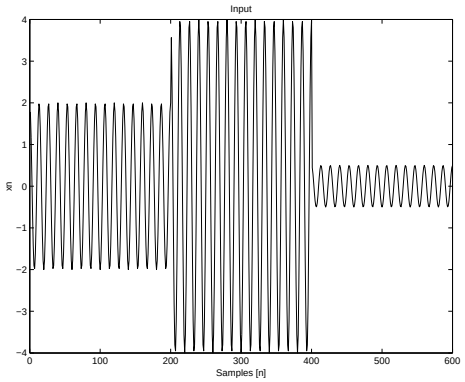
C.4.1 Simulering af kompresssor efter matematisk princip

Dette program er lavet, for at afprøve forskellige indstillinger af threshold, kompressionsrate og indstilling af koefficienten, λ. Desuden giver det er overblik over, hvordan en kompressor i teorien bør fungere.

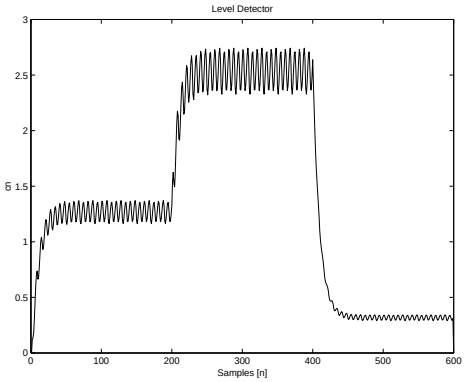
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialisering %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c0 = 0.50;                          % ønsket threshold
lambda = 0.b8;                      % filter koefficient
p = 0.50;                           % kompression rate
n = 1:600;                          % samples

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Input %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xn = cos(0.15 * pi * n);            % input funktion

xn(1:200) = xn(1:200) * 2;          % opdeling af samples
xn(201:400) = xn(201:400) * 4;
xn(401:600) = xn(401:600) / 2;

figure plot(xn)
title('Input')
xlabel('Samples [n]')
ylabel('xn')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Level Detector %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

cn = [zeros(1,length(xn))];

for i = 1:(length(xn) - 2)
    cn(i + 1) = (lambda * cn(i)) + ((1 - lambda) * abs(xn(i + 1)));
end

figure plot(cn)
title('Level Detector')
xlabel('Samples [n]')
ylabel('cn')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gain Processor %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc = [zeros(1,length(xn))];

for i = 1:(length(xn) - 2)
    if cn(i) >= c0

    fc(i) = (cn(i) / c0)^(p - 1);
```

```
else fc(i) = 1; end end

figure plot(fc)
title('Gain Processor')
xlabel('Samples [n]')
ylabel('fc')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Output %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

yn = [zeros(1,length(xn))];

yn = xn .* fc;

figure plot(yn)
title('Output')
xlabel('Samples [n]')
ylabel('yn')
```

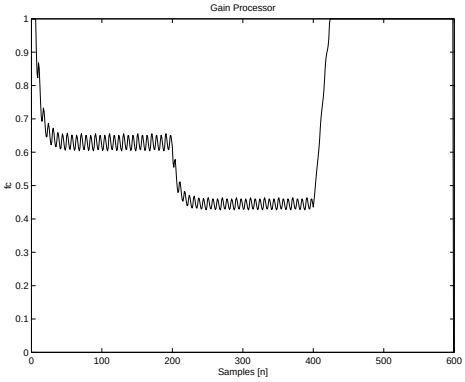
På figurerne: C.27, C.28, C.29 og C.30 ses, hvorledes signalet modificeres af kompressoren, hvordan forstærkningskoefficienten ændres og hvordan styresignalet ser ud.



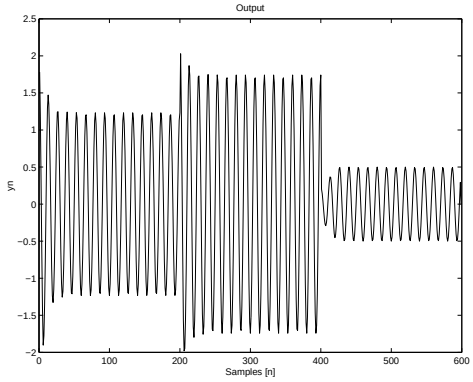
Figur C.27: *Indgangssignal*



Figur C.28: *Styresignal*



Figur C.29: *Forstærkningskoefficient*



Figur C.30: *Udgangssignal*

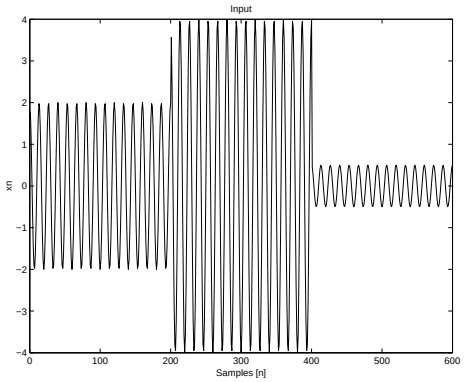
C.4.2 Simulering af 1. metode

Dette underafsnit indeholder kildekoden til simulering af metode 1 til beregning af forstærkning.

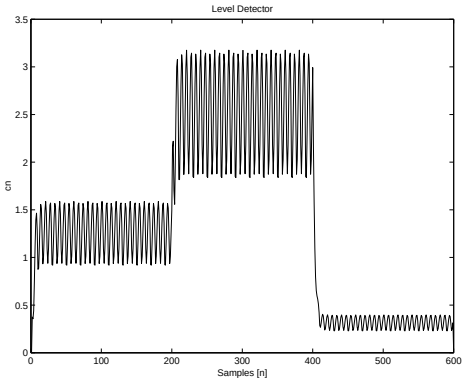
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialisering %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

c0 = 0.5;                           % ønsket threshold
lambda = 0.68;                      % filter koefficient
n = 1:600;                          % samples
k = 0; s = 0;

%%%%%%%%%%%%%%%%%%%%%%%%%%% Tabellægning af gain %%%%%%%%%%%%%%%%%%%%%%%%%%%

g1 = 0.97776; g2 = 0.95695; g3 = 0.93741; g4 = 0.91902; g5 =
0.90167; g6 = 0.88527; g7 = 0.86973; g8 = 0.85498; g9 = 0.84096;
g10 = 0.82761; g11 = 0.81487; g12 = 0.80270; g13 = 0.79106; g14 =
0.77992; g15 = 0.76923; g16 = 0.75897; g17 = 0.74911; g18 =
0.73963; g19 = 0.73049; g20 = 0.72169; g21 = 0.71889;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gain Processor %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc = [zeros(1,length(xn))];

for i = 1:(length(xn) - 2)
    if cn(i + 1) >= c0
        k = 0;
        s = cn(i + 1) - c0;
        while s > 0
            k = k + 1;
            s = s - 0.043;
        end

        if k == 1
            fc(i) = g1;
        elseif k == 2
            c(i) = g2;
        elseif k == 3
            fc(i) = g3;
        elseif k == 4
            fc(i) = g4;
        elseif k == 5
            fc(i) = g5;
        elseif k == 6
            fc(i) = g6;
        elseif k == 7
            fc(i) = g7;
        elseif k == 8
            fc(i) = g8;
        elseif k == 9
            fc(i) = g9;
        elseif k == 10
            fc(i) = g10;
        elseif k == 11
            fc(i) = g11;
        elseif k == 12
            fc(i) = g12;
        elseif k == 13
            fc(i) = g13;
        elseif k == 14
            fc(i) = g14;
        elseif k == 15
            fc(i) = g15;
        elseif k == 16
            fc(i) = g16;
        elseif k == 17
            fc(i) = g17;
        elseif k == 18
            fc(i) = g18;
        elseif k == 19
            fc(i) = g19;
        else
            fc(i) = g20;
```

```
        end
    else fc(i) = 1;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

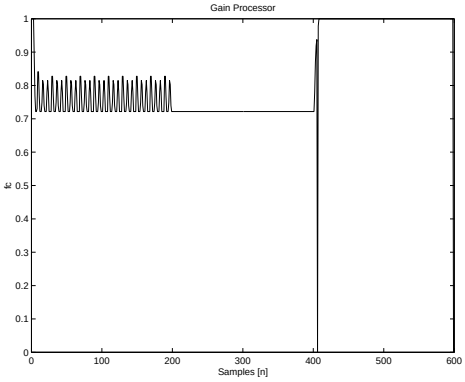
På figurerne: C.31, C.32, C.33 og C.34 ses, hvorledes signalet modificeres af kompressoren, hvordan forstærkningskoefficienten ændres og hvordan styresignalet ser ud.



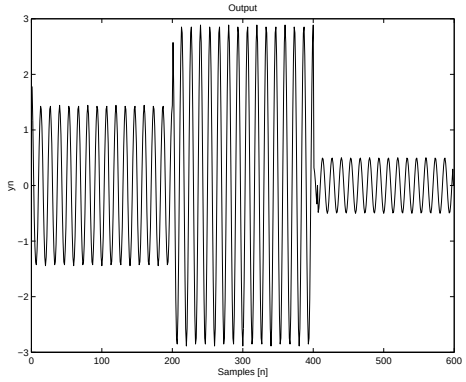
Figur C.31: *Indgangssignal*



Figur C.32: *Styresignal*



Figur C.33: *Forstærkningskoefficient*



Figur C.34: *Udgangssignal*

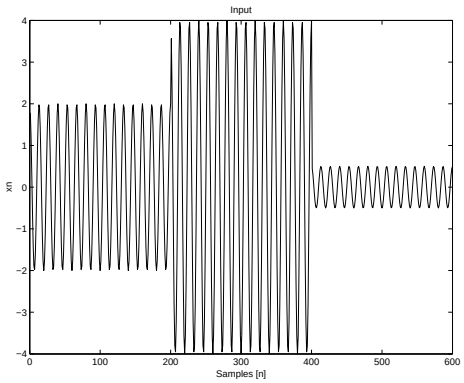
C.4.3 Simulering af 2. metode

Denne simulering er lavet for at teste den anden metode til design af forstærkningskoefficienter.

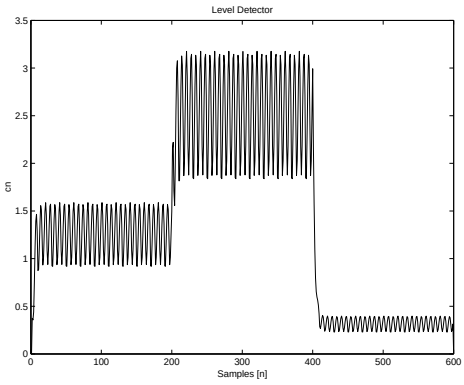
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Initialisering %%%%%%%%%%%%%%%%%%%%%%%%%%%%

c0 = 0.5;                           % ønsket threshold
lambda = 0.68;                      % filter koefficient
fc = 1;                             % initierende gain-værdi
diff = 0.005;                       % stepstørrelsen
n = 1:600;                          % samples

%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gain Processor %%%%%%%%%%%%%%%%%%%%%%%%%%%%

fc = [zeros(1,length(xn))];
```

```
for i = 1:(length(xn) - 2)
    if cn(i + 1) >= c0
        if cn(i + 1) >= cn(i)
            fc(i + 1) = fc(i) - diff;
        else
            fc(i + 1) = fc(i) + diff;
        end
    else fc(i + 1) = 1;
    end
end

if 0.5 > fc(i)
    fc(i + 1) = 0.5;
end

if fc(i) > 1
    fc(i + 1) = 1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

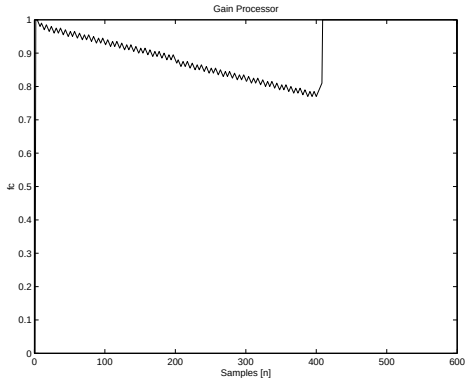
På figurerne: C.35, C.36, C.37 og C.38 ses, hvorledes signalet modificeres af kompresoren, hvordan forstærkningskoefficienten ændres og hvordan styresignalet ser ud



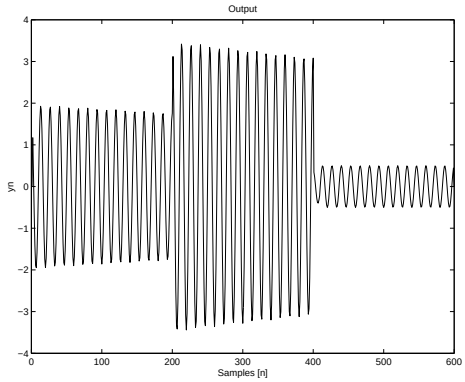
Figur C.35: *Indgangssignal*



Figur C.36: *Styresignal*



Figur C.37: *Forstærkningskoefficient*



Figur C.38: *Udgangssignal*

C.4.4 Opsummering af simuleringer

Som det kan ses på figurerne for output-signalet, resulterer metode 1 i et signal, som giver en god tilnærmelse af det matematiske princip. Metode 2 har til gengæld en kurve for udgangssignalet, som bærer præg af at være der steppes.

C.4.5 Lyttetest

For at bestemme, hvilken af de to metoder, der skal bruges til beregning af forstærkning, er der foretaget aflytninger af musiksignaler med de forskellige metoder. Under disse aflytninger blev der varieret forskellige parametre for hver af metoderne.

- For metode 1 blev der eksperimenteret med forskellige værdier af threshold og forskellige radier på ellipsen, som blev brugt til beregning af koefficienter.
- For metode 2 blev der foretaget variationer af værdier for threshold og størrelsen,x der blev steppet med.

Det blev vurderet udfra denne lyttetest, at metode 2 gav større variation af signalet. Derfor indebærer denne metode en mere interessant kompression. Oveni dette er metode 2 hurtigere i afvikling end metode 1.

C.5 Simulering/test af modul 5: Flanger

Flangeren er blevet simuleret i MATLAB, for at teste de 2 metoder, til tabellægningen af flangersinusen. Efter simuleringen beskrives simuleringsresultatet.

C.5.1 Simulering af metode 1

Der læses en wav-fil ind i MATLAB, og outputtet beregnes ved brug af en tabellagt sinus.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Guitareffekt: Flanger          %
% Dato:         30/4 2001        %
% Gruppe:       641              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fs=22050;                                   % samplefrekvensen sættes

% Læser inputfil ind i 'in'
[in fswavfil b]=wavread('../lyde/tiredman.wav');
input =[in(1:length(in))]';                 % læser in ind i input array
out=[zeros(1,length(in))];                  % lægger nuller i output array

N = 10; %Bestemmer den frekvens hvormed flangeren svinger N=4 => 1 Hz, N=30 => 0.1 Hz

% Kortlægning af sinus til flanger

t=0:1:(length(in)/N);                        % t defineres
freq_fl=0.000982;                            %
offset=121;                                  % offsettet for flangersinus
min_delay=22;                                % delay-minimum
amplitude=offset-min_delay;                  % flangersinus amplitude
startoffset=offset+amplitude;                %
kvant=((99)^-1);                             % kvantisering u/interpolation
kvant2=((990)^-1);                           % kvantisering m/interpolation

fl=sin(freq_fl*t);                           % sinusfunktionen
```

```
fl_afr=offset+amplitude*quant(fl,kvant);      % sinusfkt kvantiseret u/ipol
fl_afr_10x=offset+amplitude*quant(fl,kvant2); % sinusfkt kvantiseret m/ipol

% For-løkke der beregner flanger output

for(x=1:(length(in)/N-N-startoffset))         %
  delay=fl_afr(x);                            % heltals delay u/ipol
  d_n_1 = fix(fl_afr_10x(x))                  % delay afrundet til
                                              % heltal m/ipol
  d_n = d_n_1 + 1;                            %
  for(y=1:N)
    vaerdi(y) =input(N*x+startoffset-delay);  %
    out(N*x+y+startoffset) = input(N*x+y+startoffset)+vaerdi(y);
  end                                         %
end

% Skriver output til fil.
wavwrite(out,fswavfil,'../lyde/flanger_lyd.wav');
```

C.5.2 Simulering af metode 2

En sinussvingning inddeles i 2^6 bit, og der beregnes hældningskoefficient og y-værdi til hver af disse dele. Sinusen beregnes ud fra hældnings- og y-værdierne, og der interpoleres i 2^16 bits opløsning.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Metode 2 til Flanger guitareffekt       %%%%
%%% Gr 641 - 2001                           %%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;

start=1

bit=6;              % Antal bit sinusen skal repraesenteres ved
bitOploesning=2^16; % Opløsning af område mellem 2 hældninger
kvant=2^(-16);
step=80;            % Step der skal "steppes" frem pr gang
offset=22+99;% Offset,da systemet skal vaere kausalt (ingen fremtidige vaerdier)


fs=22050;            % Samplingsfrekvens sættes
                    % Laeser inputfil ind i 'in'
[in fswavfil b]=wavread('../lyde/guitar-akustisk2.wav');
input =[in(1:length(in))]';  % laver input array
out=[zeros(1,length(in))];      % laver outputarray



interval=360/(2^bit);    % Sinussvingning inddeles i 2^bit dele

t=0:interval:360;        % Sinussvingningen inddeles i t dele

sinuspkt=sin((2*pi*t)/360);     % Sinus opløses
% De bit-1 haeldningskoefficienter beregnes
for p=1:((2^bit));
a(p)=(bitOploesning/2)*((sinuspkt(p+1)-sinuspkt(p))/(t(p+1)-t(p)));
y(p)=(bitOploesning/2)*sinuspkt(p); % De bit-1 y-værdier beregnes
end

alphaVaelger = 1;           % Haeldningsnummeret

thetaPointer = 1;           % Pointer til placering på x-aksen (theta-aksen)
l=1;                        % Step i sinus: (2^16)*(2^bit) for 1 periode
```

```
                          % Step for bit=6 => 4.194.304

for i=1:(length(in)-offset)
  thetaPointer = (thetaPointer+(step));
                % Pointeren defineres som foregaaende pointervaerdi + update
  if thetaPointer>bitOploesning          % Når thetaPointer er kommet
                                         % til enden i hver inddeling,
                                         % tælles hældningsdelen op
    thetaPointer=thetaPointer-bitOploesning;
    l=l+1;
    if l==65
        l=1;
    end
  end

    alphaVaelger = l;
    punkt=y(l);                          % aktuel y-værdi
% Y-vaerdien = alpha*x-vaerdien
sinusVaerdi(i) = (a(alphaVaelger)*5.625*(thetaPointer/bitOploesning)+punkt)*99+offset;

  ttt=fix(sinusVaerdi(i));                       % Den delayede
                                                 % sample lokaliseres

  diff=input(i-ttt+offset+1)-input(i-ttt+offset); % differencen mellem
                                                 % de 2 signaler til
                                                 % interpolering
  vaerdi=diff*(sinusVaerdi(i)-ttt)+input(i-ttt+offset); % Interpolering

  out(i+offset) = 0.5*((input(i+offset))+vaerdi); % Outputtet beregnes

end

 wavwrite(out,fswavfil,'../lyde/nyflanger_lyd.wav'); % outputtet skrives
```

C.6 Simulering/test af modul 6: Rumklang

For at undersøge om "Schroeders reverberator" gav det ønskede resultat, blev der lavet en MATLAB fil, der foretog signalbehandlingen på en .wav fil. Denne fil blev brugt til at eftervise, at "Schroeders reverberator" giver det ønskede resultat. MATLAB simuleringen tager en lydfil, som input, og skriver en ny lydfil, hvor der er tilført rumklang. Det er vigtig, at denne fil er samplet ved 22050Hz og at den ikke er for lang. Længden af filen bestemmer hvor hurtigt signalbehandlingen er slut. MATLAB filen der benyttes er følgende:

```
% Denne fil beregner impulsresponset af rumklangseffekten ved fs=22kHz
clear all;
[in fswavfil b]=wavread('../lyde/tiredman.wav');
input =[in(1:length(in))]';      % læser in ind i input array

%Skaleringsfaktorer
S1=0.25;
S2=0.659;
K1=1.074;

% Konstant til sættelse af delay
K=1;

%Kam filter konstanter
a1=0.944*K;
a2=0.924*K;
a3=0.907*K;
a4=0.899*K;
```

```
%Allpass konstanter
a5=0.496;

%Impulsresponsmaaler
input=[zeros(1,1500) 1 zeros(1,30000)];

% Skaleringsfaktor
input=S1*input;

% Forskellige buffere til beregning af rumklang
reverb1=[zeros(1,length(input))];
reverb2=[zeros(1,length(input))];
reverb3=[zeros(1,length(input))];
reverb4=[zeros(1,length(input))];
reverb_out=[zeros(1,length(input))];
output1=[zeros(1,length(input))];
output=[zeros(1,length(input))];
taeller=[zeros(1,length(input))];

% Kamfilter beregninger
%y(n)=a*y(n-d)+x(n-d)
for i=1:(length(input)-1022);
taeller(i)=i+1;
reverb1(i+801)= input(i+801)+(a1*reverb1(i));
reverb2(i+850)= input(i+850)+(a2*reverb2(i));
reverb3(i+920)= input(i+920)+(a3*reverb3(i));
reverb4(i+1021)= input(i+1021)+(a4*reverb4(i));
end

% Summering af kamfiltre samt skalering til allpassfiltre
reverb_out=(reverb1+reverb2+reverb3+reverb4)*0.25;
reverb_out=S2*reverb_out;

% Allpassfilter beregninger
% X'[n]=A5*X'[n-d]+S2*reverb_out
% y[n]= A5*K1*X'[n]+K1*X'[n-D]
for i=1:(length(input)-111);
output1(i+111)= a5*output(i)+S2*reverb_out(i+111);
output(i+111)= (a5*K*output1(i+111))+(K1*output(i+111));
end

wavwrite(output,fswavfil,'../lyde/reverb.wav'); % outputtet skrives
```

På baggrund af denne fil vurderes det, at rumklangen kan implementeres som "Schroeders reverberator".

APPENDIKS D

Beregning af støj

Dette appendiks omhandler støjberegning på dele af systemet. Der er foretaget beregninger af støj på ét analogt modul samt ét digitalt modul, nemlig indgangstrinnet og rumklangeffekten. Det undersøges, hvorvidt kravet om et SNR på mindst 60dB er overholdt

D.1 Støj i indgangstrinnet

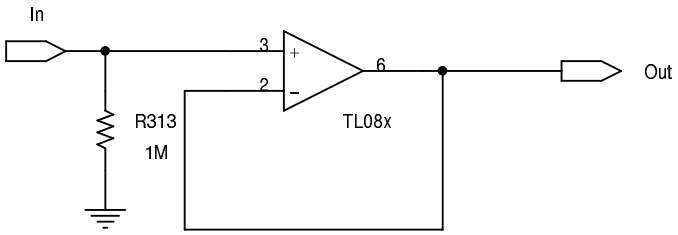
Der er flere forskellige former for støj i elektriske kredsløb med aktive og passive komponenter. De tre understående former for støj optræder i alle kredsløb med forstærkning vha. transistorer:

- Kontaktstøj: opstår hvor to metalflader møder hinanden.
- Termisk støj: findes i alle ledende materialer.
- Shot-noise: optræder i alle p-n overgange, altså dioder/transistorer.

Eftersom indgangstrinnet leverer det signal som resten af systemet foretager signalbehandling af, er dette et essentielt led i det samlede system. Det er derfor vigtigt at kredsløbet ikke opfanger/genererer uhensigtsmæssigt meget støj, hvorfor støjen i indgangstrinnet er beregnet. Afsnittet med beregningerne er opdelt i to underafsnit. Først beregnes der støj i spændingsfølgeren og derefter i limiter kredsløbet.

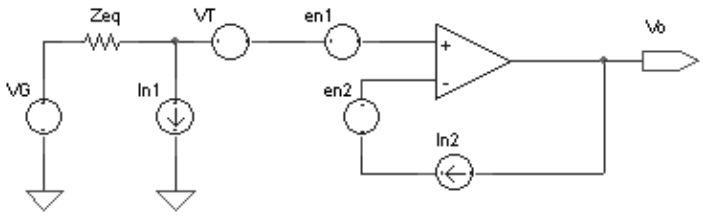
D.1.1 Støj i spændingsfølger

Spændingsfølgeren er koblet som vist på figur D.1 på næste side



Figur D.1: *Diagram for spændingsfølgeren.*

Hvis de forskellige støjkilder, repræsenteret som strøm og spændingskilder samt modstande indtegnes i diagrammet, kommer det til at se ud som på figur D.2



Figur D.2: *Det modificerede spændingsfølgerkredsløb med støjkilder.*

De på figuren viste generatorer, V_T, $en1$, $en2$, $In1$, $In2$ er støjkilder. V_T er den termiske støj, givet ved:

$$V_t = \sqrt{4 \cdot k \cdot T \cdot B \cdot Z_{eq1}} \tag{D.1}$$

- K = Boltzman's konstant.
- T = Absolut temperatur i grader Kelvin (298K).
- B = Båndbredde i Hz (11kHz).
- Z_{eq1} = Indgangsmodstand som operationsforstærkeren ser ind i.

Det ses at guitarens indgangsmodstand sidder i parallel med modstanden på 1MΩ, og derfor er indgangsmodstanden, Z_{eq1}, maksimalt $\frac{1}{2}$MΩ. Ved indsættelse af værdier beregnes den maksimale termiske støj til:

$$V_{t_{Z_{eq1}}} = \sqrt{4 \cdot 1.38 \cdot 10^{-23} J/K \cdot 298K \cdot 11kHz \cdot \frac{1}{2} M\Omega} = 9,51\mu V \tag{D.2}$$

e_n er operationsforstærkerens ækvivalente input-støjspænding og i_n er den ækvivalente input-støjstrøm. Disse værdier findes i databladet for operationsforstærkeren til maksimumværdier på hhv. 55 $\frac{nV}{\sqrt{Hz}}$ og 2.8 $\frac{fA}{\sqrt{Hz}}$ ved frekvensen 10 Hz og 25 ° C. Dette betyder, at støjspændingen, udfra kendskabet til båndbredden kan findes til at være:

$$V_n = \sqrt{B} \cdot e_n = \sqrt{11kHz} \cdot 55 \frac{nV}{\sqrt{Hz}} = 5,77\mu V \tag{D.3}$$

På samme måde beregnes støjstrømmen I_n:

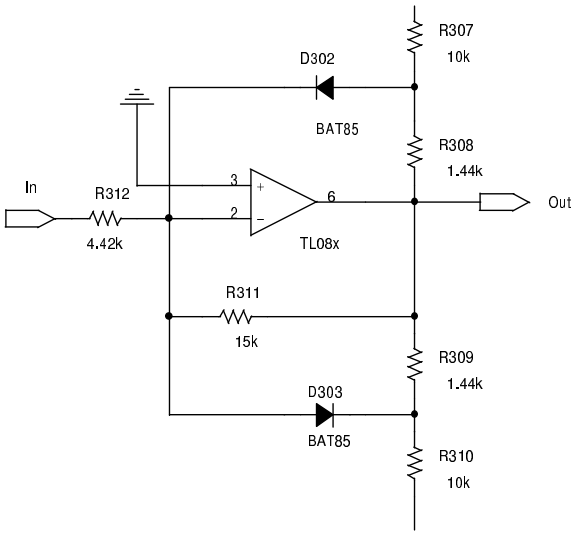
$$I_n = \sqrt{B} \cdot i_n = \sqrt{11kHz} \cdot 2.8\frac{fA}{\sqrt{Hz}} = 0,29pA \tag{D.4}$$

Det samlede støjbidrag fra koblingen kan findes udfra formlen:

$$V_{nt1} = \sqrt{2 \cdot V_n^2 + V_{t_{Z_{eq1}}}^2 + (I_n \cdot Z_{eq1})^2} = 12,53\mu V \tag{D.5}$$

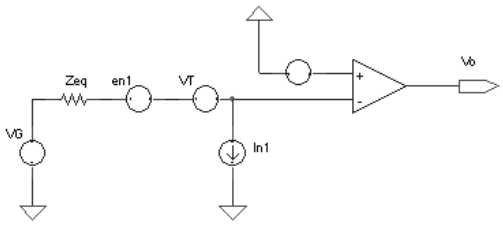
D.1.2 Støj i limiter kredsløb

Limiter kredsløbet er koblet som vist på figur D.3:



Figur D.3: *Diagram for limiter krdsløbet.*

Når støjkilderne indsættes i diagrammet kommer det til at se ud som på figur D.4



Figur D.4: *Diagram for limiter kredsløbet med støjkilder implementeret.*

Den termiske støjspænding beregnes på samme måde som for spændingsfølgeren, bortset fra en ændring af indgangsmodstanden. Den er givet ved:

$$Z_{eq2} = R312 \parallel R311 \approx 3,45k\Omega \tag{D.6}$$

Herved kan den termiske støj beregnes til:

$$V_{t_{Z_{eq2}}} = \sqrt{4 \cdot 298K \cdot 1,38 \cdot 10^{-23} \cdot 11kHz \cdot 3.45k\Omega} = 0,79\mu V \tag{D.7}$$

Det samlede støjbidrag fra limiter koblingen kan findes udfra formlen:

$$V_{nt2} = \sqrt{2 \cdot V_n^2 + V_{t_{Z_{eq2}}}^2 + (I_n \cdot Z_{eq2})^2} = 8.2\mu V \tag{D.8}$$

D.1.3 Støjen i hele indgangstrinnet

Det samlede støjbidrag fra de to kredse findes ved følgende formel:

$$V_{nt3} = \sqrt{V_{nt2}^2 + V_{nt1}^2} = \sqrt{(12,53\mu V)^2 + (8.2\mu V)^2} = 14.97\mu V \tag{D.9}$$

Signal-støj forholdet i kredsløbet er naturligvis meget afhængig af signalets amplitude. Hvis signalet ikke klippes ved en amplitude på $0,7V_p$, er SNR mindst:

$$SNR_{min} = 20 \cdot log(\frac{\frac{0,7V}{\sqrt{2}}}{14.97\mu V}) \approx 90,4dB \tag{D.10}$$

Derfor er SNR mindst denne værdi for signalniveauer på $1.4V_{pp}$. For et signal på $0,1V_{pp}$ er SNR mindst 67dB.

det konkluderes, på baggrund af de ovenstående beregninger, at SNR praktisk taget altid vil have en værdi på over 70dB. Eftersom beregningerne er foretaget på de data for operationsforstærkeren , som kun forekommer i absolut værste tilfælde er det en rimelig antagelse. Derfor er kravet om et SNR-forhold på over 60dB overholdt for indgangstrinnet.

D.2 Trunkeringsstøj

Den benyttede processor er en 16bit processor med 32bit produkt- og akkumulatorregistre. Når tal gemmes, gemmes de 16 mest betydende bit. Det betyder, at der opstår støj (trunkeringsstøj). Denne støj fremkommer hver gang et signal gemmes i RAM. I det følgende undersøges først den generelle teori for trunkeringsstøj, hvorefter de enkelte digitale effektenheders støj vil blive udregnet.

Afsnittet bygger på [[9], s. 391-418]

D.2.1 Trunkeringsstøj generelt

Når to tal multipliceres bliver det til en 32bit produktsum. Derved opstår ingen støj, men når dette tal gemmes i RAM, gemmes kun de 16 mest betydende bit. Derved fjernes 16bit fra signalet, og trunkeringsstøjen opstår. Der er i kravspecifikationen opstillet et krav om, at SNR skal være over 60dB. Derfor udregnes SNR i hvert af de digitale moduler. Der findes 2 forskellige måder at udregne SNR på: "Worst case" metoden og variansmetoden. Først beskrives "worst case" signal/støj forholdet.

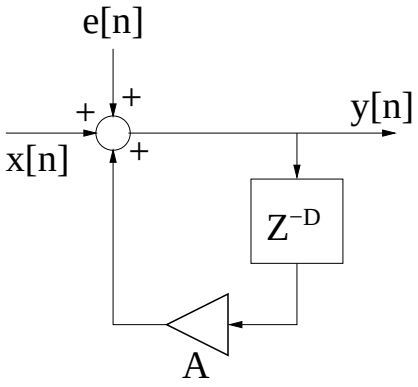
"worst case" SNR
Dette er givet ved:

$$SNR = \frac{y[n]}{e_y[n]} \tag{D.11}$$

$$\Updownarrow$$

$$SNR_{DB} = 20 \cdot log_{10}\frac{y[n]}{e_y[n]} \tag{D.12}$$

Hvor y[n] er det maksimale outputsignal og $e_y[n]$ er den fejl, der er på udgangen af systemet. Dette er illustreret på figur D.5



Figur D.5: *Støjkilde i kamfilter.*

Her er støjkilden e[n]. For at finde $e_y[n]$ bruges følgende formel [[9], s. 396] :

$$e_y[n] = e[n] \cdot \sum_{n=-\infty}^{\infty} |h_{ey}[n]| \tag{D.13}$$

Hvor $|h_{ey}[n]|$ er den numeriske summerede impulsrespons fra støjkilden (e[n]) og til outputtet (y[n]). Størrelsen e[n] er i "worst case" sat lig med det mindst betydende bit uden signbit (e[n]=2^{-15}).

Kamfiltrets "worst case" signal/støj forhold beregnes til (her med A=0,7):

$$SNR_{DB} = 20 \cdot log_{10}\frac{y[n]}{e_y[n]} \tag{D.14}$$

$$\Downarrow$$

$$SNR_{DB} = 20 \cdot log_{10}\frac{1}{2^{-15} \cdot \sum_{n=-\infty}^{\infty} |h[n]|} \tag{D.15}$$

$$\Downarrow$$

$$SNR_{DB} = 20 \cdot log_{10}\frac{1}{2^{-15} \cdot 3,333} = 79,85dB \tag{D.16}$$

SNR ved brug af variansmetoden
Ved variansmetoden bruges middeleffekten af støjen til at beregne SNR med. Ved denne metode antages følgende:

- Alle støjkilder (e[n]) er hvid støj og stationære i den svage betydning (WSS).
- Alle støjkilder er ukorrelerede.
- Støjen opstår som følge af afrunding, ikke trunkering.

Ved denne metode er SNR givet ved [[9], s. 403] :

$$SNR = \frac{\delta_x^2}{\delta_f^2} \tag{D.17}$$

$$\Updownarrow$$

$$SNR_{DB} = 10 \cdot log_{10}\frac{\delta_x^2}{\delta_f^2} \tag{D.18}$$

Hvor δ_f^2 er variansen af trunkeringsstøjen på outputtet, og δ_x^2 er variansen af trunkeringsstøjen på inputtet. Denne er ved et fuldt udstyret sinussignal lig $\frac{1}{2}$ [9]. Trunkeringsstøjen på outputtet beregnes ved at finde trunkeringsstøjen i systemet, og føre den ud til outputtet. Dette illustreres ved at se på kamfiltret fra før (figur D.5 på modstående side). Her findes variansen af e[n], δ_e, der ved 2's komplementære tal er givet ved [[9], s. 394] :

$$\delta_e^2 = \frac{2^{-2 \cdot B}}{12} \tag{D.19}$$

Hvor B er antallet af bit uden signbit (dvs. 15). For at finde støjkildens indvirkning på outputtet bruges følgende formel [[9], s. 396] :

$$\delta_y^2 = \delta_e^2 \cdot \sum_{n=0}^{\infty} |h_{ey}[n]|^2 \tag{D.20}$$

Hvor $|h_{ey}[n]|^2$ er den kvaderede impulsrespons fra støjkilden til output. For kamfiltret er denne ved A=0,7:

$$\sum_{n=0}^{\infty} |h_{ey}[n]|^2 = 1,9608 \tag{D.21}$$

Derved bliver δ_y^2:

$$\delta_y^2 = \delta_e^2 \cdot \sum_{n=0}^{\infty} |h_{ey}[n]|^2 = \frac{2^{-30}}{12} \cdot 1,9608 = 1,52 \cdot 10^{-10} \tag{D.22}$$

Dette giver et signal/støj forhold på:

$$SNR_{DB} = 10 \cdot log_{10}\frac{\delta_x^2}{\delta_y^2} = 95,17dB \tag{D.23}$$

Valg af metode

Som det ses er der stor forskel på SNR alt afhængig af den benyttede metode. Dette hænger sammen med, at SNR ændrer sig i takt med input signalet. Ved et system med mange

støjkilder vil signal/støj forholdet i "worst case" ikke være særlig stort, da der altid anvendes "worst case" udregninger. Systemet vil have et større SNR en udregningerne indikerer.

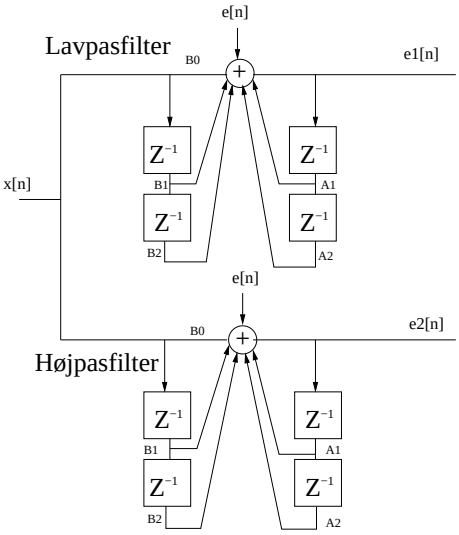
For at undersøge hvad SNR er ved et musiksignal kunne der laves et MATLAB program, der udregnede trunkeringsstøjen. Denne ville i teorien være:

$$e_y[n] = y_{trunkeret}[n] - y[n] \tag{D.24}$$

Udfra denne værdi kunne variansen (δ_f^2) og SNR for det pågældende musiksignal findes. Det er af tidsmæssige årsager dog ikke lavet. I stedet beregnes støjen vha. variansmetoden, da projektgruppen mener, at denne metode giver det bedste billede af signal/støj forholdet.

D.2.2 Støj i kompressor

Støjbidraget i kompressoren kommer primært fra de to IIR filtre, der benyttes i frekvensopdelingen. Disse er konstrueret som Direkte Form 1, og har derfor een støjkilde hver. Dette ses på figur D.6.



Figur D.6: *To IIR filtre med støjkilder.*

I beregningen af disse støjkilders indflydelse på outputtet er følgende antaget:

$$e_y[n] = e1[n] + e2[n] \tag{D.25}$$

Hvilket betyder at:

$$\delta_f^2 = \delta_{f_{low}}^2 + \delta_{f_{high}}^2 \tag{D.26}$$

Hvor variansen af støjen for de 2 filtre findes ved:

$$\delta_f^2 = \delta_e^2 \cdot (1+1,7991+0,8175+1+1,7991+0,8175) = \frac{2^{-30}}{12} \cdot 7,2332 = 5,614 \cdot 10^{-10} \tag{D.27}$$

Med et sinus signal bliver δ_x^2 lig $\frac{1}{2}$, hvilket giver et SNR på:

$$SNR_{DB} = 10 \cdot log_{10} \frac{\delta_x^2}{\delta_f^2} = 89,50dB \tag{D.28}$$

D.2.3 Støj i flangeren

Flangereffekten består egentligt et 1.ordens FIR filter. Derved er trunkeringsstøjen på outputtet lig med e[n]. Denne er givet ved [[9], s. 411] :

$$\delta_y^2 = \frac{2^{-2 \cdot B}}{12} \tag{D.29}$$

$$\Downarrow$$

$$\delta_y^2 = \frac{2^{-30}}{12} = 77,6 \cdot 10^{-12} \tag{D.30}$$

Hvorved SNR for flangeren bliver (δ_x^2 er stadig $\frac{1}{2}$):

$$SNR_{flanger} = 10 \cdot log_{10} \frac{\frac{1}{2}}{77,6 \cdot 10^{-12}} = 98,09dB \tag{D.31}$$

Da der benyttes 16bit talrepræsentation kan det maksimale SNR være 96dB (2^{16}). Derfor kan SNR i flangeren ikke være 98dB. Dog kan det konkluderes, at flangeren ikke bidrager med trunkeringsstøj.

D.2.4 Støj i rumklang

Der er 6 støjkilder i rumklangseffekten. De 4 kamfiltre har hver en støjkilde, og allpassfiltret har to støjkilder. Der foretages en simplificering af beregning af støjen i dette system. Der antages, at alle støjbidragende adderes til output. Dette giver

$$\delta_f^2 = \delta_{f_{comb1}}^2 + \delta_{f_{comb2}}^2 + \delta_{f_{comb3}}^2 + \delta_{f_{comb4}}^2 + \delta_{f_{all}}^2 \tag{D.32}$$

Kamfiltret
Kamfiltrets SNR blev udregnet i indledningen af dette afsnit, hvor støjbidraget δ_f var givet ved:

$$\delta_y^2 = \delta_e^2 \cdot \sum_{n=0}^{\infty} |h_{ey}[n]|^2 \tag{D.33}$$

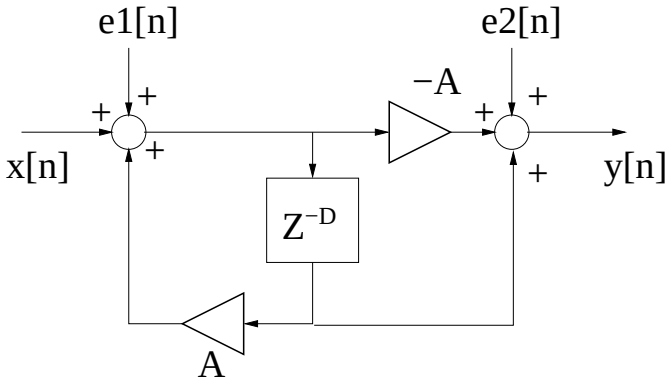
$$\Downarrow \tag{D.34}$$

Hvor $|h_{ey}[n]|^2$ er afhængig tilbagekoblingsforstærkningen A. Ved maksimal rumklangseffekt er δ_f for hver af de 4 kamfiltre:

$$\delta_{f_{comb1}}^2 = 7,29 \cdot 10^{-10}$$

$$\delta^2_{f_{comb2}} = 5,67 \cdot 10^{-10}$$
$$\delta^2_{f_{comb3}} = 4,38 \cdot 10^{-10}$$
$$\delta^2_{f_{comb4}} = 2,43 \cdot 10^{-9}$$

Allpassfiltret
Der er 2 støjkilder i allpassfiltrene. Disse er vist på figur D.7. Disse bliver tilsammen til $\delta_{f_{all}}$



Figur D.7: *De 2 støjkilder i allpassfiltret.*

Differensligningen for allpassfiltret er:

$$y[n] = A \cdot y[n-D] - A \cdot x[n] + x[n-D] \tag{D.35}$$

Ved at sætte variansen af støjkilden e1[n] som input, og beregne y[n] numerisk, findes dennes indflydelse på outputtet. Denne er i MATLAB beregnet til:

$$\delta^2_{f1} = 1,5399 \cdot 10^{-10} \tag{D.36}$$

Støjbidraget fra e2[n] er en enkel produktsum, der foretages lige før signalet sendes ud af allpassfiltret. Derfor er variansen af e2[n]:

$$\delta^2_{f2} = \frac{2^{-30}}{12} \tag{D.37}$$

Derved bliver $\delta_{f_{all}}$:

$$\delta_{f_{all}} = \delta^2_{f2} + \delta_{f1} = 1,5399 \cdot 10^{-10} + \frac{2^{-30}}{12} \tag{D.38}$$

SNR for rumklang

Som førnævnt er det antaget, at støjbidragende summeres tilsidst ved y[n]. Derved bliver den samlede varians på outputtet:

$$\delta^2_f = \delta^2_{f_{comb1}} + \delta^2_{f_{comb2}} + \delta^2_{f_{comb3}} + \delta^2_{f_{comb4}} + \delta_{f_{all}} \tag{D.39}$$
$$\Downarrow$$
$$\delta^2_f = (7,29 + 5,67 + 4,38 + 0,243) \cdot 10^{-10} + 1,5399 \cdot 10^{-10} + \frac{2^{-30}}{12} \tag{D.40}$$

Dette giver et SNR på:

$$SNR_{DB} = 10 \cdot log_{10} \frac{\frac{1}{2}}{17,76 \cdot 10^{-10}} = 84,52dB \tag{D.41}$$

D.2.5 Opsamling

Det kan på baggrund af ovenstående udregninger konkluderes, at alle de digitale moduler har et signal/støj forhold på over 60dB. Dog skal det nævnes, at der i denne udledning er foretaget en del simplificeringer for at holde udregningerne på et forholdsvis simpelt niveau. Derfor er det muligt, at SNR kan være dårligere end det beregnede. Ydermere kan de digitale effektenheders signal/støj forhold være lavere pga. udregninger (f.eks. lineær interpolation i flangeren).

APPENDIKS E

Assemblerkode

BILAG I

Europarack

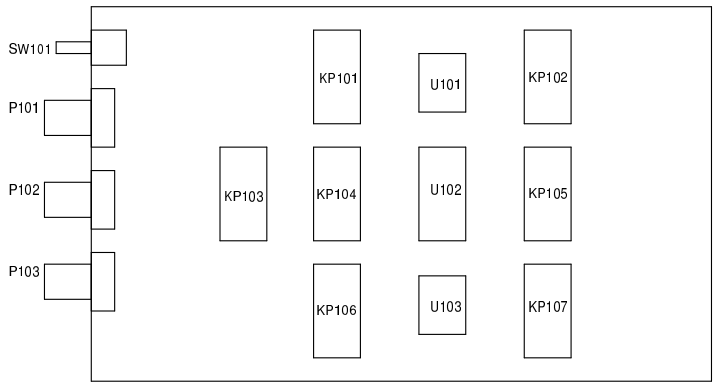
I.1 Busforbindelser

Bus-ben	Forbindelse	Bus-ben	Forbindelse
A1	Ingen Forbindelse	C1	+5V
A2	Ingen Forbindelse	C2	Ingen Forbindelse
A3	Ingen Forbindelse	C3	Output D/A / Input Equalizer
A4	Ingen Forbindelse	C4	Output Equalizer / Input Forvrænger
A5	Ingen Forbindelse	C5	Output Forvrænger
A6	Ingen Forbindelse	C6	Output Indgangstrin / Input A/D
A7	Ingen Forbindelse	C7	Ingen Forbindelse
A8	Ingen Forbindelse	C8	Ingen Forbindelse
A9	Ingen Forbindelse	C9	Ingen Forbindelse
A10	D0	C10	Ingen Forbindelse
A11	D1	C11	SYNC
A12	D2	C12	Clocks
A13	D3	C13	Data in DSP
A14	D4	C14	Data out DSP
A15	D5	C15	Ingen Forbindelse
A16	D6	C16	Ingen Forbindelse
A17	D7	C17	Ingen Forbindelse
A18	D8	C18	Ingen Forbindelse
A19	D9	C19	Ingen Forbindelse
A20	D10	C20	Ingen Forbindelse
A21	D11	C21	Ingen Forbindelse
A22	D12	C22	Ingen Forbindelse
A23	D13	C23	Ingen Forbindelse
A24	D14	C24	Ingen Forbindelse
A25	D15	C25	Ingen Forbindelse
A26	Ingen Forbindelse	C26	Ingen Forbindelse
A27	Ingen Forbindelse	C27	Ingen Forbindelse
A28	Ingen Forbindelse	C28	Ingen Forbindelse
A29	Ingen Forbindelse	C29	-15V
A30	Ingen Forbindelse	C30	Analog stel/jord
A31	Ingen Forbindelse	C31	+15V
A32	Digital stel/jord	C32	Ingen Forbindelse

Tabel I.1: Busforbindelser for Europarack

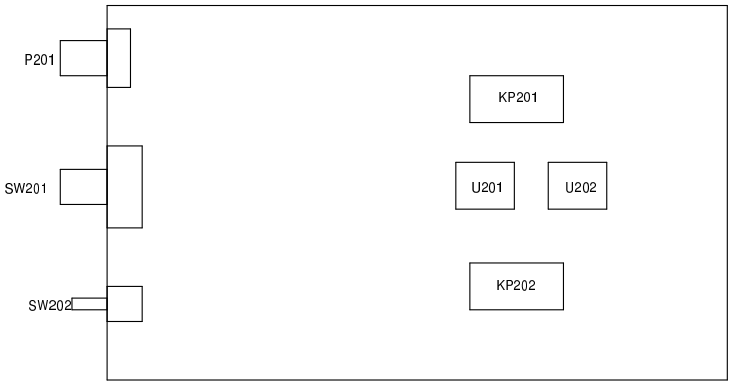
I.2 Komponentplaceringer på Europakort

Equalizer



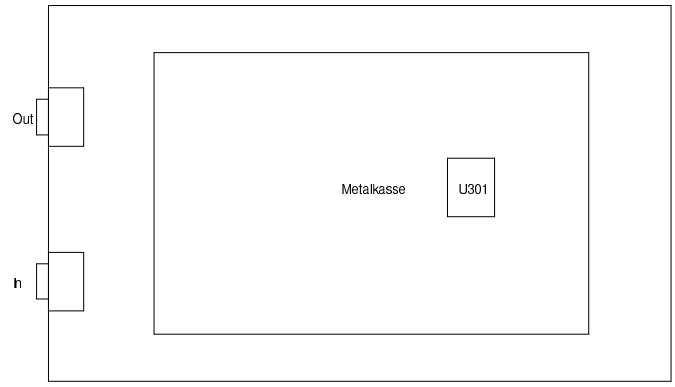
Figur I.1: *Komponentplacering for equalizer europakort*

Forvrænger



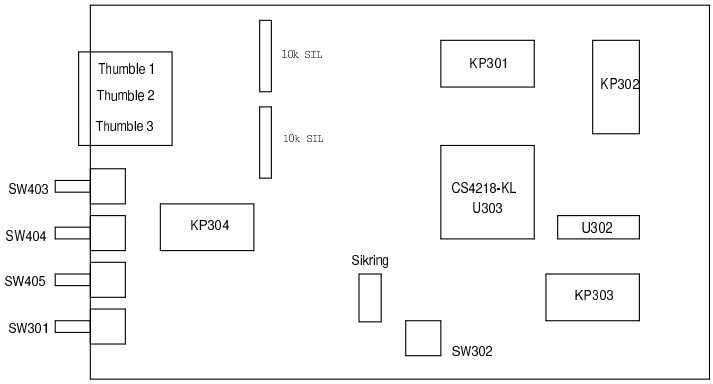
Figur I.2: *Komponentplacering for forvrænger europakort*

Indgangstrin



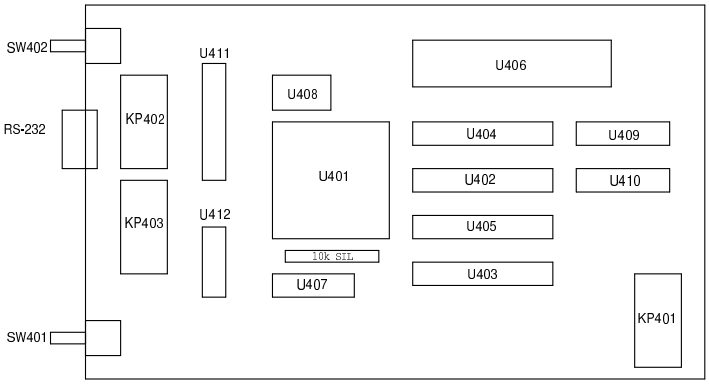
Figur I.3: *Komponentplacering for indgangstrin europakort*

Codec og Brugerinterface



Figur I.4: *Komponentplacering for codec- og brugerinterface-europakort*

DSP



Figur I.5: *Komponentplacering for DSP-europakort*

BILAG II

Diagram Equalizer

BILAG III

Diagram Forvrænger

BILAG IV

Diagram Codec

BILAG V

Diagram DSP

BILAG VI

Den vedlagte CD-ROM

I Windows vil CD-ROM'en starte af sig selv, hvis der findes en internet-browser. Der vil herefter kunne klikkes videre til datablade, programmerings-filer, samlede projekt og andet. Der findes på CD-ROM'en en *readme.txt*, hvor der står hvad indhold af CD-ROM'en er og hvordan den opstartes i en internet-browser. Filen *readme.txt* kan åbnes i en almindelig tekst editor, og findes i roden af CD-ROM drevet:
<CD-ROM>:*readme.txt*
Hvor <CD-ROM> er drev-bogstavet for CD-ROM drevet (normalt D).
Indhold:

\datablade
Her ligger datablade for anvendte komponenter i pdf format.

\matlab
Her er de skrevne matlab-filer.

\software
Den skrevne programkode til assembler programmer.

\pdf
Her ligger diagrammer o.a. i pdf format.

\diagrammer
Samlet diagram over hardware til programmet OrCAD.

Mere information om de nævnte biblioteker findes på CD-ROM'en.