

PRESENTATION FÖR TONSÄTTARE AV ANVÄNDANDET AV EMS-I

1. Allmänt

1.1 EMS-I är ett apparatorienterat språk, vilket betyder att tonsättarna får översätta sina musikaliska föreställningar till apparattermer. Andra möjliga typer av språk är exempelvis notationsorienterat språk, där tonsättaren slår en bro mellan sina musikaliska föreställningar och ett notations-system, och ett kompositionstekniskt orienterat språk, där bron slås från den musikaliska föreställningen till kompositionsregler. Riktlinjer för utarbetandet av EMS-I har varit det sätt varpå manöverbordet fungerar.

2. Apparater

2.1 Namn

I EMS-I används förkortningar för apparaternas namn. Därvidlag tillämpas en förkortningsregel, som säger att om apparatnamnet består av ett ord, används de tre första bokstäverna i ordet, består apparatnamnet av flera ord, används den första bokstaven i varje ord.

Exempel: channel = CHA, frequency generator = FG

Här följer en fullständig apparatförteckning:

AM	AMPLITUDE MODULATOR
AMP	AMPLIFIER
AT	ANALOGUE TAPE
CD	CHANNEL DISTRIBUTOR
CHA	CHANNEL
FF	FREQUENCY FILTER
FG	FREQUENCY GENERATOR
FS	FREQUENCY SHIFTER
NG	NOISE GENERATOR
REV	REVERBERATOR
RM	RING MODULATOR

2.2 Apparaternas parametrar

Apparaternas parametrar skrivs på så sätt att apparatnamnet följs av vänsterparentes följt av parametrarna med kommatecken emellan, följt av högerparentes. Parametrarna skrivs i exemplet nedan med sina namn, inte sina

värden. Exempelvis skriver vi FG(NUMMER,FREKVENNS,NIVÅ,VÅGFORM). Apparatnamn plus apparatens parametrar kallas term.

I det praktiska användandet, när värden på parametrarna anges, kan motsvarande exempel se ut på följande sätt:

FG(1,440,100,2)

Exempel (samtliga apparater):	<u>FÖRKORTNING</u>	<u>FULLSTÄNDIG EMS-I-TERM</u>
	AM	AM(NR, INGÅNG, NIVÅ)
	AMP	AMP(NR, NIVÅ)
	AT	AT(NR, NIVÅ)
	CD	CD(NR, KANAL, NIVÅ)
	CHA	CHA(NR, NIVÅ)
	FF	FF(NR, FILTERKANAL, NIVÅ)
	FG	FG(NR, FREKVENNS, NIVÅ, VÅGFORM)
	FS	FS
	NG	NG(NR, NIVÅ, FÄRG1, FÄRG2)
	REV	REV(NR, EKOTID, NIVÅ)
	RM	RM(NR, INGÅNG, NIVÅ)

Exempel med värdet på parametrarna angivna:

FG(1,440,100,2)	RM(1,B,100)
FF(2,20,110)	REV(2,6,100)
RM(1,A,100)	CHA(4,80)

2.3 Förkortningar

Observera att inte alla apparaternas parametrar behöver skrivas ut! I de fall då någon parameter inte skrivs ut, behåller den sitt värde från förr gången parametern sattes.

Exempel:

```

FG(1,440,100,2)
.
.
(Ny tidsangivelse)
.
.
FG(1,,20)

```

Exemplet visar att frekvensen och vågformen bibehålls, men nivån ändras

från 100 till 20 dB.

2.4.1 Kopplingar

Apparaterna kan kopplas ihop till apparatkedjor med kopplingstecknet

>

och uppkopplingen kan brytas med nedkopplingstecknet

#

2.4.2 Koppling av apparater och apparatkedjor

I exempel A här nedan kopplas frekvensgenerator 1 till utgående kanal 1, så att vi kan lyssna på frekvensgeneratorns ton, sedan tid angivits. I exempel B kopplas frekvensgenerator 4 och frekvensgenerator 2 till ringmodulator 1 och denna i sin tur till utgående kanal 2, och vi kan lyssna till resultatet, sedan tid angivits. Alla apparatkedjor bör avslutas med semikolon.

- A. FG(1,440,100,2)>CHA(1,80);
- B. FG(4,300,100,2)>RM(1,A,100)>CHA(2,80);
FG(2,350,100,3)>RM(1,B);
- C. FG(1,440,100,2)#CHA(1,80);

Se appendix I för en uppställning av alla tillåtna kopplingar.

2.4.3 Olika apparater kan samlas i en grupp av apparater genom att apparattermer- na knyts ihop med tecknet

&

Det är viktigt att skilja mellan en grupp av apparater och en kedja av apparater och/eller apparatgrupper. En grupp av apparater är exempelvis ett antal frekvensgeneratorer eller ett antal filterkanaler. Här följer några exempel på en apparatkedja, som är sammansatt av grupper av apparater och enskilda apparater:

```
FG(1,440,100,2)&FG(2,660,100,2)>RM(1,A,100)>CHA(1,80)&CHA(2,80);  
NG(1,100)>FF(1,5,110)&FF(1,10,110)>RM(1,B);
```

Grupp av apparater, kopplad till annan grupp av apparater eller enskild apparat, medför att samtliga apparater i första gruppen kopplas till samtliga apparater i andra gruppen, resp att apparaterna i gruppen kopplas till en enskild apparat.

Exempel:

```
FG(1)&FG(4)&FG9&REV(1)
```

```
FG(1,100)&FG(2,200)&FG(3,300)&FG(4,400)>CHA(1,100);
```

Den senare raden innebär att frekvensgeneratorerna 1,2,3 och 4 ges var sin frekvens och sedan kopplas till kanal 1, som ges nivån 100 dB.

2.4.4 För frekvensgeneratorerna och filterkanalerna gäller att grupper av dessa kan samlas med tecknet



En förutsättning är att man inte vill utesluta någon apparat mellan det lägsta och det högsta numret. Så till exempel betyder

FF(1,5←12,100)

att kanalerna 5 till och med 12 i frekvensfilter nr 1 skall ställas in på 100 dB.

Exemplet

FG(1←6)

betyder att frekvensgeneratorerna 1 till och med 6 skall kopplas någonstans med bibehållande av sina gamla värden.

Exempel:

FG(1)&FG(7←12,,80,2)&FF(1,1←3,100)»

CHA(1)&CHA(2)&CHA(3,98);

Detta skrivsätt medför att frekvensgenerator 1 och dessutom nr 2 och nr 3, alla med sina gamla parametervärden, och frekvensgeneratorerna 7,8,9,10,11 och 12, alla med sina gamla frekvenser, med den nya nivån 80 dB och den nya vågformen 2 samt filter 1 kopplas till kanalerna 1,2 och 3. Kanal 1,2 och 3 i filter 1 ställs också på 100 dB och kanal 3 på 98 dB.

Övningsuppgift 1:

Gör en eller flera grupper bestående av frekvensgeneratorgrupperna 3,6,9 och 12 och eko 1, koppla det som går till kanal 1 via filter 1 eller eljest direkt till kanal 1. Låt enheterna behålla de värden de hade sig tilldelad tidigare.

Övningsuppgift 2:

Gör en lämplig grupp bestående av 5 eller 6 frekvensgeneratorer. Låt dessa amplitudmoduleras till 90% (= 99 dB) av frekvensen 100 Hz i AM(2). Koppla den resulterande signalen till kanal 1. A-ingången på en amplitudmodulator modulerar B-ingången.

(Lösningar av uppgifterna, se sid 6)

3. Tid

Vi skiljer mellan TID och DURATION, på så sätt att TID står för när någon händelse inträffar, medan DURATION (DUR) står för händelsens varaktighet, t ex tonens duration.

3.1 Allmänt

På manöverbordet har vi tre sätt att beskriva duration. För det första ordern TIDMÄTA, som betyder att studion fås att klinga under den tid TIDMÄTA-knappen hålls nedtryckt, alltså en duration. Det andra sättet är ordern LYSSNA KONTINUERLIGT, som betyder att studion fås att klinga, när knappen för ordern LYSSNA KONTINUERLIGT trycks ned, och fås att sluta klinga när ordern STOPPA KONTINUERLIGT LYSSNANDE trycks ned. Det tredje sättet att beskriva tid vid manöverbordet är ordern TIDL YSSNA, som fungerar så, att den tid man vill att studion skall klinga sätts i indexregistret varefter knappen för ordern TIDL YSSNA trycks ned. Studion klingar då under den tid som angivits i indexregistret.

Tids- och durationsbeskrivningarna i EMS-I sammanfaller inte med durationsorderna vid manöverbordet. Vid manöverbordet orienterar man sig i musikstrukturen, eller rättare sagt digitalbandet, via rekordnummer. I EMS-I orienterar man sig i musikstrukturen via olika tids- och durationsbegrepp.

3.2 Global Time

Uttrycket Global Time (GT) hänför sig till den löpande kompositionstiden. För ett stycke som är 10 minuter långt menar man med Global Time noll styckets början, Global Time 5 minuter mitten av stycket och Global Time 10 minuter slutet.

I EMS-I skrivs efter varje färdigt block eller PART - se punkt 7.18 och exempel på sid - på kommentarmediet automatiskt ut GLOBAL TIME, begynnelse-tid och sluttid, uttryckta i Global Time. Man kan räkna fram den aktuella GT genom att addera tiden för slutet på föregående block och till denna addera begynnelse-tiden för den klang man arbetar med plus den aktuella Local Time, d v s där tonsättaren nu befinner sig.

3.3 Local Time

Tiden mellan varje enstaka ton har däremot sin motsvarighet i EMS-I och kallas Local Time, förkortat LT(MS) eller LT(SEC,MS). Millisekundparameter får inte innehålla ett tal större än 99 999, och sekundparameter inte ett tal större än 8 333. LT-termen kan mera korrekt sägas ange när de koppling och apparatinställningar som följer fram till nästa LT-term skall utföras.

Exempel på LT-termer:

```
LT(0) FG(1,440,100,2)>CHA(1,80);
```

```
LT(2000)FG(1,660,100,3)>CHA(1,80);
```

LT(52,230)

LT(52,0)

LT(52,)

Det är inget krav att LT-termer kommer i tidsföljd.

LT(900)FG(1,200)

LT(0)FG(1,100,100)>CHA(1,100);

Lösning av övningsuppgift 1:

FG3>FG6;

FG9>FG12;

FG6&FG12>FF(1);

FF(1)&REV(1)>CHA(1);

Lösning av övningsuppgift 2:

FG15>FG18>AM(2,B,100)>CHA(1,100);

FG(24,100,99)>AM(2,A);

3.4 Duration

Medan LT är en tidpunkt där något skall hända, t ex att en frekvensgenerator skall börja klinga, finns termen duration - förkortad D(MS) - för den tid vilken anger hur länge en viss ljudnivå skall hållas.

Exempel:

LT(20,200)FG(1,440,100,2)>CHA(1,80)>D(2000);

Övningsuppgift 3

FG(1) och FG(4) är inställda, men inte kopplade. FG(1) skall börja ljuda efter 31 sek och 120 ms räknat från blockets början, och ljuda under 2316 ms. FG(4) skall börja klinga efter 23 sek och durera i 20000 ms. Båda ljuden kanalen 3, Ange den del av programmet som ombesörjer detta.

(Lösning av uppgiften på sid 8.)

Medan termen LT gäller för alla apparater fram till nästa LT, gäller termen D endast för den apparat eller apparatgrupp, som den är hopkopplad med.

Införande av ny LT-term görs antingen för att införa ändringar i apparaturparameterar - en ny ton, en ny tonstyrka - eller speciellt för att införa 0-värden i nivåparametern, vilket betyder att apparaten fr o m nu inte längre skall klinga. (Om D-termen i förbindelse med envelopper och glissa

se punkt 4.4 och 4.5).

3.5 Tidmäta

är en term som inte finns i EMS-I, utan endast vid manöverbordet.

3.6 Tidlyssna

finns även den endast vid manöverbordet. En nära motsvarighet är emellertid ordern PLAY (se punkt 7.8).

3.7 Lyssna kontinuerligt

finns endast vid manöverbordet, men kan naturligtvis även nås via EMS-I med ordern PLAY, under förutsättning att det man vill lyssna på redan har givits en mycket lång duration.

4. Envelopper

4.1 Enveloppen, d v s tonstyrkeändringarna, på manöverbordet är planerad att utföras med hjälp av regler. Motsvarigheten i EMS-I är termen ENVELOP, förkortad till ENV.

4.2 Den fullständiga termen med parametrar skrivs:

ENV(STARTNIVÅ,SLUTNIVÅ,DUR,KURVTYP.TIDSSTEG)

vilket betyder att en utgångsnivå kan ändras till en slutnivå under en specificerad duration i millisekunder, och följande en bestämd kurvform med ett visst tidssteg mellan ändringarna. Tidssteget är skillnaden i ms mellan två på varandra följande förändringar. Tidssteget får ej vara större än envelopptiden.

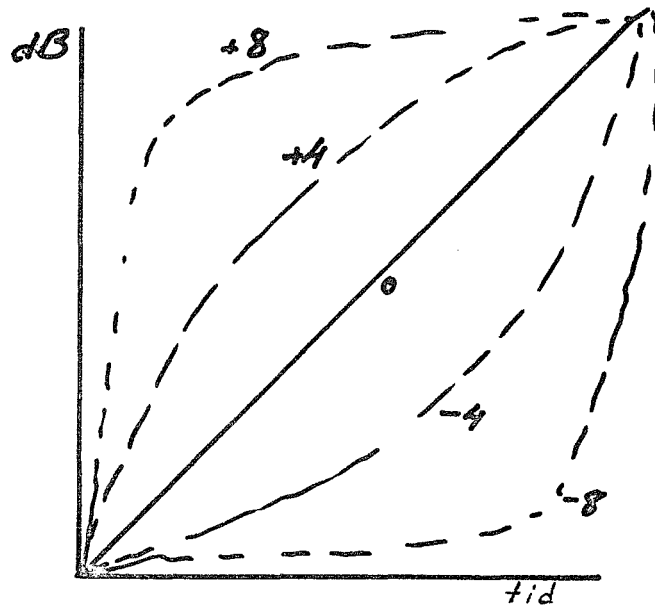
4.3 Kurvformer

Vi har 19 olika kurvformer som är uppbyggda i tre klasser. Den första klassen består av en enhet, nämligen en linjär ändring från ett värde till ett annat värde. Den andra klassen består av 9 enheter, som har det gemensamt att de är konvexa, och den tredje klassen av 9 enheter som är konkava. Den linjära klassen beskrivs i EMS-I med talet \emptyset . De 9 konvexa enheterna beskrivs med siffrorna 1-9, där 1 är den minst konvexa och 9 den mest konvexa. De 9 konkava enheterna beskrivs med siffrorna -1 t o m -9, där -1 är den minst och -9 den mest konkava.

I det fall då tonstyrkan ändrar sig från svagare till starkare kan konvex också beskrivas som ett crescendo, där den största tonstyrkeändringen sker i början av tonen. I det fall då tonstyrkan ändrar sig från ett högre värde till ett lägre betyder valet av en konvex kurva att den största tonstyrke

ändringen kommer i slutet. I de konkava fallen är förhållandena de motsatta.

Kurvorna kan grafiskt illustreras på följande sätt:



Lösning av övningsuppgift 3:

LT(31,120)

FG(1)>D(23168)>CHA(3,100);

LT(23,)

FG(4)>D(20000)>CHA(3,100);

Kommentar: Det är inget krav att LT-termerna kommer i tidsföljd. Observera att D-termen måste komma före kanal-termen för att önskad effekt skall uppnås. Om vi skrivit

FG(1)>CHA(3,100)>D(23168);

resp

FG(4)>CHA(3,100)>D(20000);

och haft samma starttider som ovan för ljuden hade detta medfört att båda ljuden upphört efter 23 sek + 20000 ms (= 43 s) i st f vid tidpunkterna 54 s 288 ms resp 43 s.

- 4.4 I tidshänseende utgör envelopptermen ett specialfall i o m att tiden ingår som en av parametrarna. Av det skälet uppstod under utformningen av språket två tolkningsmöjligheter, nämligen 1) skall en ton, som är kopplad till en envelop och därigenom får ett tonstyrkeförlopp från en nivå till en annan nivå, fortsätta att klinga efter det att slutnivån har nåtts, eller 2) skall man anse att när slutnivån har nåtts, denna ton skall sluta klinga?

I EMS-1 har den lösningen valts att man anser att tonen fortsätter att klinga och att tonsättaren måste ge en order, om han vill att den inte skall klinga efter uppnådd slutnivå; denna order är

Z

Om tonsättaren vill att tonen endast skall klinga under den tid det tar att utföra enveloppen, skrivs exempelvis på följande sätt:

ENV(0,100,500,2,20)>Z;

vilket betyder att den ton, som enveloppen är kopplad till, börjar med tonstyrkan 0 dB, växer till 100 dB under en tid av 500 MS, följande kurvtyp 2, varefter nivån går tillbaka till 0 dB och tonen följaktligen inte hörs längre. Talet 20 anger tidssteget mellan ändringarna (samplingshastigheten).

- 4.5 D-termen kan också appliceras på en enveloppterm på så sätt att ljudkällan klingar under den tid som specificerats i envelopptermen plus tiden i den D-term som envelopptermen är kopplad till.

Exempel:

ENV(0,100,500,2,20)>D(1000)

vilket betyder, liksom i föregående exempel, att tonstyrkan växer från 0 till 100 dB under en tid av 500 MS, men att den i detta fall fortsätter att klinga på slutnivån under ytterligare 1000 MS, d v s D-termens tid.

- 4.6 En enveloppedja kan skapas genom användande av kopplingstecknet mellan envelopptermerna.

Exempel:

ENV(0,100,500,20)>ENV(100,100,10)>

ENV(100,0,500,2,10)

- 4.7 Enveloppterm och enveloppedja kan med hjälp av kopplingstecknet kopplas till apparater och/eller apparatgrupper.

Exempel:

LT(0)FG(1,1000,100,2)>ENV(0,100,500,2,10)>ENV(100,100,1000)>

ENV(100,50,500,2,10)>Z;

FG(1)>CHA(1,100)&CHA(2,100);

Termen ENV(100,100,1000) är ett specialfall av envelopptermen, där start- och slutvärdet är lika.

- 4.8 ESTEP och GSTEP

Tonsättaren kan specificera hur ofta datorn skall sätta ett nytt värde under genereringen av envelopper och glissandon. Om inget värde anges kommer datorn att sätta nytt värde var tionde millisekund. Om man önskar att datorn alltid skall sätta ett annat värde sker detta med termen

ESTEP(MS) när det gäller enveloppen, och termen GSTEP(MS) när det gäller glissandon.

Exempel: ESTEP(1)

vilket betyder att datorn sätter ett nytt värde på nivån varje ms;

GSTEP(20)

vilket betyder att datorn sätter ett nytt värde på frekvensen var 20:e ms.

Om man önskar ändra på tidssteget endast under en enveloppterm, skrivs det önskade tidssteget som en parameter i envelopptermen. Därefter gäller åter det värde som satts med ESTEP resp GSTEP.

Av tekniska skäl får en envelop eller ett glissando inte innehålla fler än 2047 tidssteg.

5. Glissando

5.1 Glissandotermens parametrar

Glissando skrivs med en term som liknar envelopptermen, nämligen GLIS, följt av utgångsfrekvens, slutfrekvens, duration i ms, kurvform och tidssteg.

Exempel:

GLIS(FREKVENS,FREKVENS,DUR,KURVFORM,GSTEP) (Betr GSTEP se 5.3)

GLIS(440,800,10000,2,1)

5.2 Kurvformer

Glissandotermens kurvformer är desamma som envelopptermens.

5.3 GSTEP

GSTEP är glissandotermens motsvarighet till ESTEP hos envelopptermen, alltså hur man anger tiden mellan varje gång datorn skall ge ett nytt värde - samplingstätheten. Är värdena för glest satta riskerar man att få en skala i stället för ett glissando. Är värdena för tätt satta använder man onödig information.

5.4 En enveloppterm hänföres till närmast föregående apparatterm. Starttiden i en enveloppterm är senast angivna Local Time eller vid envelopkedjor sluttid för närmast föregående enveloppterm (om ej någon apparatterm står emellan och delar envelopkedjan i två kedjor). Detsamma gäller för GLIS-termen.

Observera att envelopptider och glissandotider ignorerar varandras existens och följaktligen kan vävas in i varandra. Envelopptermer och kedjor kan blandas med glissandotermer och kedjor.

Exempel:

```
FG(1←3)&FG(7←9,,,2)>
```

```
ENV(50,100,500)>GLIS(220,440,200,2)>
```

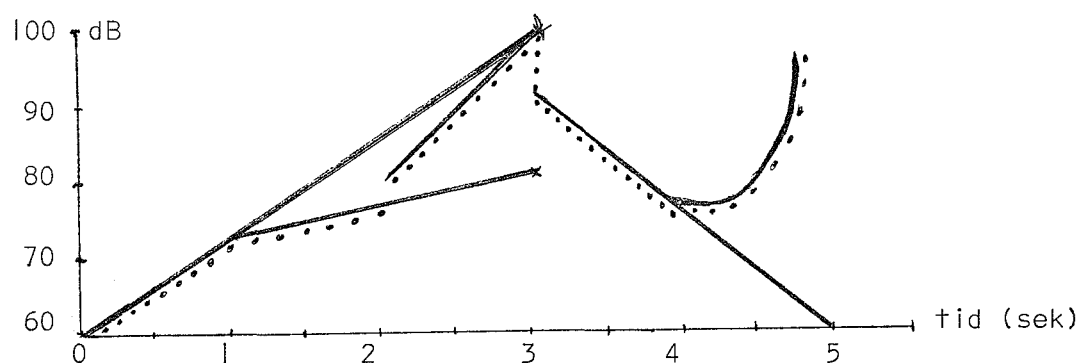
```
GLIS(440,880,1000)>ENV(100,90,1200,1,10);
```

```
CHA(1,100)>ENV(80,100,2000,1)>D(10000);
```

5.5 Vid överlappande envelopper gäller den senast definierade.

Exempel:

Vi har givit en frekvensgeneratornivå följande envelopper:



```
LT(0,0)FG(1)>ENV(60,100,3000);
```

```
LT(1,0)FG(1)>ENV(73,80,2000);
```

```
LT(2,0)FG(1)>ENV(80,100,1000);
```

```
LT(3,0)FG(1)>ENV(90,60,2000);
```

```
LT(4,0)FG(1)>ENV(75,100,1000,-3);
```

Den resulterande enveloppen prickad.

6. Frekvensgeneratorernas kopplingsnät

På manöverbordet har de 24 frekvensgeneratorerna ett eget kopplingsnät. Kopplingsnätet är utformat så, att det har 24 ingångar, samma antal som antalet frekvensgeneratorer, men endast 8 utgångar. Frekvensgeneratorerna är alltså kopplade i grupper om tre på så sätt att frekvensgeneratorerna 1, 2 och 3 har en utgång som kallas FG3, 4, 5 och 6 har utgång FG6, etc. Dessa 8 kopplingspunkter benämns FG3, FG6, FG9, FG12, FG15, FG18, FG21 och FG24.

6.1 Uppbyggande av en klang

Du vill ha ett ljud som består av en grundton med en frekvens av 200 Hz och med en styrka på 70 dB och med harmoniska övertoner, som är 1.5, 2 resp 3 gånger högre än grundtonen, var och en med 3 dB lägre styrka än den närmast föregående tonen. Varje ton skall matas ut i en egen kanal (var sitt hörn av lyssningsrummet). Vågformen skall vara sinus, d v s nr 0 eller nr 1. Ljudet skall klinga under 30 sekunder. Hur gör man detta med EMS-1?

Lösning:

```

1  PART(ONE)
2  LT(0)
3  FG(1,200,70,0)>CHA(1,100);
4  FG(4,300,67,0)>CHA(2,100);
5  FG(7,400,64,0)>CHA(3,100);
6  FG(10,600,61,0)>CHA(4,100)>
7  D(30000);
8  PLAY

```

Kommentar till lösningen:

Rad 1. Av tekniska skäl består alla EMS-1-program av block. Ett block består av högst 7000 termer, och tonsättaren måste ange var blocket börjar och var det slutar. Början av blocket anges med ordet PART(NUMMER) och slutet med END. Varje PART skall ha ett namn. Det här heter (ONE).

Rad 2. LT(0) - Local Time - talar om när i ljudobjektet någonting skall hända. Vid tidpunkten 0 utföres alltså de närmast följande raderna (3 - 7).

Rad 3. Här ställs frekvensgenerator nr 1 på frekvensen 200 Hz, nivån 70 dB och sinusvåg. Frekvensgeneratorn 1 samt 2 och 3 som alltid är hopkopplade med 1, kopplas vidare till utkanal 1 och dennas utnivå ställs på 100 dB, vilket innebär att man inte får någon relativ förstärkning, d v s att ljudnivån vid lyssning blir 70 dB.

Rad 4-6. För att kunna koppla frekvensgeneratorerna vidare till olika ställen måste de väljas så, att de inte är fast kopplade till varandra.

Rad 7. D(30000) är en enveloppterm. Den är kopplad till kanal 4, som därigenom fås att bibehålla sin nivå under 30 sekunder (30000 ms) och sedan tystna. Eftersom inga ytterligare tidsangivelser finns, slutar allt efter 30 sekunder.

Rad 8. Vi provlyssnar resultatet med ordern PLAY. Tekniskt resulterar ordern i att objektet sorteras - d v s alla LT-termer ordnas i rätt följd - översätts till EMS-kod, spelas in på magnetband och återskapas i stora studion.

6.2 För att sudda ut fel använder man i EMS-I principen att skriva om. Har man exempelvis skrivit

```
LT(2100) FG(4,440,92)»CHA(1,100);
```

```
LT(5400) FG(6,356,70,3)»CHA(4);
```

och vill ändra 92 till 80, så görs detta genom att första ta reda på efter vilken LT-term felet ligger, därefter i vilken apparat och slutligen i vilken parameter. Håller man på att skriva en text, hittar man LT-termen med felet i genom att skriva LT(2100). Man kan alltså rätta fel var som helst i den text som finns, när som helst. Efter att ha hittat LT-termen hittar man apparaten genom att skriva dess namn - i detta fall frekvensgenerator nr 4 - följt av apparatens parametrar inkl den rättade parametern. Man skriver alltså

```
LT(2100)FG(4,440,80)
```

eller

```
LT(2100)FG(4,,80)
```

och felet har rättats.

6.3 CLEAR

Vid manöverbordet kan man använda ordern NOLLSTÄLL, om man vill vara säker på att inga gamla inställningar finns kvar när man börjar skapa en ny klang. Motsvarande kommando i EMS-I är CLEAR.

SAMMANFATTNING AV ELEMENT I EMS-I

Apparattermer

AM(NR, INGÅNG, NIVÅ)
AMP(NR, NIVÅ)
CD(NR, KANAL, NIVÅ)
CHA(NR, NIVÅ)
FF(NR, FILTERKANAL, NIVÅ)
FG(NR, FREKVENNS, NIVÅ, VÅGFORM)
NG(NR, NIVÅ, FÄRG1, FÄRG2)
REV(NR, EKOTID, NIVÅ)
RM(NR, INGÅNG, NIVÅ)

Studiopunkter

FS
FG3, FG6, FG9, FG12, FG15, FG18, FG21, FG24

Kommandon

CLEAR

TEMP-kommandon

PLAY

Lokal tid

LT(MS), LT(SEK, MS) Observera att LT(1,5) betyder detsamma som
LT(1,005), ej LT(1,500).

Envelopptermer

ENV(NIVÅ, NIVÅ, MS, KURVFORM, STEG)
Z
D(MS)

Glissandoterm

GLIS(FREKVENNS, FREKVENNS, MS, KURVFORM, STEG)

Stepptermer

ESTEP(MS)
GSTEP(MS)

EXIT avbryter körningen och startar upp EMS-I för ett nytt program.

>, #, & står mellan element i kopplingskedjor, envelopkedjor och glissandokedjor. Om kedjan fortsätter över flera rader skall den avdelas så att ett av dessa tecken står sist på raden.

;

bör avsluta varje kedja.

←

får användas i FG- och FF-termer.

'KOMMENTAR' bör användas relativt flitigt. Man glömmer annars mycket snart vad ett program handlar om.

EXEMPEL PÅ FELUTSKRIFTER

I det följande finns en del exempel på felutskrifter. Det första exemplet skall kommenteras här.

\ raderar föregående tecken. Kommentarer kan insättas mellan enkel-
apostrofer ('...'). Det är en god vana att i kommentarer som stöd för
minnet skriva upp vad programmet handlar om.

```
:PART(ONE)
:LT(Ø)
:FG(1,23Ø #20)»CHA(1,1ØØ);
  FG(1,23Ø # )
Ø2 Ø11      ILLEGAL DELIMITER
!FG(1,23Ø,2Ø)
!
:'FELET AER NU RAETTAT'
```

Raderna 1 och 2 är korrekta, men i tredje raden finns ett fel. Tonsättaren får ett meddelande om detta från datorn efter det att han skrivit tredje raden och gjort ett carriage return. Datorns felutskrift gör tonsättaren uppmärksam på dels den symbol där felet finns och dels den symbol som därefter är korrekt och som tonsättaren skall avsluta rättelserna med. På en ny rad skrivs koden för felet ut med två tal samt beskrivningen av felet "ILLEGAL DELIMITER". I detta fall är felet att symbolen för nedkoppling har skrivits som skiljetecken i en frekvensgeneratorterm. Felet rättas till av tonsättaren genom att han skriver om hela FG-termen efter det att datorn har meddelat honom att den är klar att ta emot rättelser genom att skriva ut ett utropstecken i början av ny rad. Efter rättelsen gör tonsättaren carriage return, varefter datorn skriver ut ett kolon i början av nästa rad och meddelar med detta att den är klar att ta emot mera EMS-1 text, som i detta fall placeras efter termen CHA(1,100).

INTERACTIVE MODE? YES OR NO?

YES

```
:PART(FELU)
:LT(0)
:FG(1,100#80,5)
  FG(1,100#    )
02  011      ILLEGAL DELIMITER
!FG(1,100,80,5)
!
:FG(7,A,75)
  FG(7,A,    )
02  012      ILLEGAL PARAMETER
!FG(7,300,75)
!
:'THE SYMBOL A IS RESERVED TO DENOTE AN ENTRY IN AM- OR RMTERMS'
```

INTERACTIVE MODE? YES OR NO?

YES

```
:FG(1,2,3,4,5)
  FG(    )
01  020      PART COMMAND FIRST THING IN A BLOCK!
!
:PART(2)
:FG(1,2,3,4,5)
  FG(1,2,3,4, )
02  014      ILLEGAL NUMBER OF PARAMETERS
!FG(1,2,3,4)
!
:FG(25,100)
  FG(25,    )
02  121      ILLEGAL DEVICE NUMBER
!FG(24,100)
!
:FG(16,16000)
  FG(16,16000)
02  022      ILLEGAL FREQUENCY GENERATOR FREQUENCY
!FG(16,15999)
  FG(16,15999)
02  011      ILLEGAL DELIMITER
!FG(16,15999)
!
:FG(3,,8)
  FG(3,,8)
02  023      ILLEGAL FREQUENCY GENERATOR WAVEFORM
!FG(3,,7)
!
:FG(12,,120.25)
  FG(12,,120.25)
02  024      ILLEGAL FREQUENCY GENERATOR INTENSITY
!
:FG(12,,120)
```

```

:FF(1,29,121)
  FF(1,29, )
02 121 ILLEGAL DEVICE NUMBER
!FF(1,28,121)
  FF(1,28,121)
02 033 ILLEGAL FREQUENCY FILTER INTENSITY
!FF(1,28,120)
!
:AMP(3)
  AMP(3)
02 121 ILLEGAL DEVICE NUMBER
!AMP(2)
!
:FG(24)>CHA(1);
  ;
02 051 ILLEGAL (DIS-) CONNECTION
!FG(24)>FG24>CHA(1);
!
:REV(2,16,90)
  REV(2,16, )
02 062 ILLEGAL REVERBATION TIME
!REV(2,15,90)
!
:REV(2,0,130)
  REV(2,0,130)
02 063 ILLEGAL REVERBATION INTENSITY
!REV(2,0,120)
!
:D( )
  D( )
02 070 TIME MISSING IN D-TERM
!D(100)
!
:D(0)
  D(0)
02 071 ILLEGAL TIME
!D(1)
  D(1)
02 245 ENV OR GLISTIME LESS THAN STFP
!ESTEP(1)
!D(1)
!
:NG(1,90,BLUE)
  NG(1,90,BLUE)
04 020 SYMBOL NOT DEFINED
!NG(1,90,A)
  NG(1,90,A)
02 031 ILLEGAL NOISE COLOUR
!NG(1,90,PINK)
!
:NG(1,150)
  NG(1,150)
02 032 ILLEGAL NOISE INTENSITY
!NG(1,120)
!
:AM(3,B,100)
  AM(3, )
02 121 ILLEGAL DEVICE NUMBER
!AM(2,PINK)
  AM(2,PINK)
02 092 ILLEGAL AMPLITUDE MODULATOR ENTRY
!AM(2,B,170)
  AM(2,B,170)
02 093 ILLEGAL AMPLITUDE MODULATOR INTENSITY
!

```

```

:RM(1,PINK,90)
  RM(1,PINK,
02  102          ILLEGAL RING MODULATOR ENTRY
!RM(1,B,140)
  RM(1,B,140)
02  103          ILLEGAL RING MODULATOR INTENSITY
!RM(1,B,120)
!
:CD(1,4,150)
  CD(1,4,150)
02  112          ILLEGAL CHANNEL DISTRIBUTOR INTENSITY
!
:AT(3,-10)
  AT(3,-10)
02  150          ILLEGAL ANALOG TAPE INTENSITY
!AT(3,10)
!
:CHAC(1,1000)
  CHAC(1,1000)
02  142          ILLEGAL CHANNEL INTENSITY
!CHAC(1,100)
!
:ESTEP(0)
  ESTEP(0)
02  201          ILLEGAL ESTEP VALUE
!ESTEP(10)
!
:FG(1)>ENV(100,120);
  >ENV(100,120);
02  241          ILLEGAL ENV SYNTAX
!FG(1)>ENV(100,80,1000)
!
:>ENV(100,130,1000)
  >ENV(100,130, )
02  242          ILLEGAL ENV AMPLITUDE
!>ENV(100,120,1000)
!
:>ENV(80,110,0)
  >ENV(80,110,0)
02  245          ENV OR GLISTIME LESS THAN STEP
!
:>ENV(110,106,10,10)
  >ENV(110,106,10,10)
02  247          ILLEGAL ENVELOPE TYPE
!>ENV(110,106,10,9)
!
:ENV(80,96,1000)
  ENV( )
02  251          NO DEVICE TO ENV GIVEN
!
:LT(-5,170)
  LT(-5, )
02  261          ILLEGAL LOCAL TIME VALUE
!LT(5,170)
!
:GSTEP(-4)
  GSTEP(-4)
02  271          ILLEGAL GSTEP VALUE
!GSTEP(4)
!
:FF(1,28)>GLISC(440,880,1000);
  >GLISC( )
02  301          NO DEVICE TO GLISSANDO GIVEN
!FG(1,,82)>GLISC(440,880,1000);
!

```

```
:FG(7)>GLIS(1);
      >GLIS(1);
02    302          ILLEGAL GLISSYNTAX
!FG(7)>GLIS(100,300,2100)
!
:FG(8)>GLIS(300,100,0)
      >GLIS(300,100,0)
02    245          ENV OR GLISTIME LESS THAN STEP
!>GLIS(300,100,300,-10)
      >GLIS(300,100,300,-10)
02    304          ILLEGAL GLIS TYPE
!>GLIS(300,100,300,-9)
!
:>GLIS(0,16000,3000,-4)
      >GLIS(0,16000,      )
02    305          ILLEGAL GLIS FREQUENCY
!>GLIS(0,15999,3000,-4)
!
```

EMS-27. Arbetsprocedurer7.1 Allmänt

I det följande skildras en av flera möjliga arbetsprocedurer. Den här beskrivna vilar på antagandet att en tonsättare bygger upp sin musikstruktur med utgångspunkt i att han låter en grupp enstaka toner bilda ett ljudobjekt och att han vidare bygger upp sin struktur med hjälp av sådana ljudobjekt. De enstaka tonerna har vi döpt till klanger, en grupp av klanger till objekt, och en grupp objekt till PART.

Med hjälp av denna teknik kan tonsättaren bygga upp en struktur i form av en följd av objekt och PART. Han kan också arbeta i två steg, d v s först bygga upp en villkorlig följd av objekt och senare placera dessa objekt i rätt följd med hjälp av deras namn.

7.2 TEMP, ACC och AUX

Manöverbordet fungerar ju så, att en klang lagras i ett register som i sin tur överförs till och lagras på digitalband. Vid avspelning passerar informationen ett buffertminne innan den åter lagras i registret och åter-skapar klangen. Det som i EMS-2 bäst motsvarar manöverbordets register kallas TEMP, ACC och AUX. EMS-2 har m a o tre olika lagringsställen, där manöverbordet har ett. Detta kan utnyttjas så, att en klang byggs upp i TEMP och överförs till ACC (kommandot heter MIX eller APP). En ny klang kan byggas upp i TEMP, och de båda klangerna mixas och återförs till ACC. Man har även tillgång till en provmixning, där tonsättaren kan lyssna till resultatet och kanske ändra på den senaste klangen, till skillnad från den definitiva mixningen, där resultatet sedan inte går att åter särskilja. Man får givetvis försäkra sig om att objekten går att mixa, d v s att inte det ena objektet använder en apparat som även återfinns i det andra. När ett PART är färdigt, skriver man END, varvid innehållet i ACC överförs till magnetband.

7.3 Överförande av en klang i TEMP till ACC

En klang i TEMP överförs till ACC med kommandot MIX(START). Rent tekniskt leder kommandot till att klangen i TEMP sorteras, d v s ljuden sorteras i

den ordning de skall börja låta. Därefter förs klangen över till ACC. TEMP förbereds så att en ny klang kan skrivas in i den. (START) anger tidsreferensen för klangen i TEMP när TEMP sorteras ihop med eventuellt tidigare innehåll i ACC. När klangerna i TEMP och ACC mixas behöver de med andra ord inte börja vid samma tidpunkt. ACC börjar alltid vid LT(0) medan TEMP + ex kan förskjutas 20 ms genom att det skrivs MIX(20).

7.4 Provmixning

Vill man endast provlyssna på en mixning används kommandot TRY(START). Är man sedan nöjd med resultatet och vill göra en definitiv mixning, används därpå MIX(START). I annat fall har man chansen att göra TRY med någon annan starttid eller använda CLEAR.

7.5 CLEAR och CLEAR(MIX)

Kommandot CLEAR åstadkommer att TEMP raderas på gamla inställningar. På samma sätt kan man radera ut ACC genom ordern CLEAR(MIX). Både TEMP och ACC kan raderas på samma gång med ordern CLEAR(ALL). Dock ändras vid CLEAR inga inställningar, som redan finns i ACC, eller vid CLEAR(MIX) inget som ligger kvar från ett föregående block. Vid MIX sker även CLEAR. Vid END sker CLEAR(MIX). Observera att PART(NUMBER) inte raderas ut, då den inte är någon inställning.

7.6 Nollställningen av TEMP vid MIX kan upphävas om KEEP följer omedelbart. Detsamma gäller ACC om KEEP(MIX) följer omedelbart efter END. Vi kan förstå detta så, att man efter ett CLEAR börjar om i TEMP och skriver över de rader som fanns där, men att vi efter KEEP fortsätter skriva där vi slutade. Om man t ex vill sortera in samma objekt vid flera tidpunkter kan man skriva MIX(tid 1); KEEP (för att få tillbaka objektet); MIX(tid 2) o s v.

7.7 Manöverbordets order SPELA motsvaras i EMS-2 av PLAY, PLAY(MIX), PLAY(ALL), PLAY(ONE) och PLAY(ONE,TWO).

7.8 Kommandot PLAY betyder att den klang som byggts upp i TEMP spelas.

7.9 Kommandot PLAY(MIX) betyder att den klang som lagras i ACC spelas.

7.10 Kommandot PLAY(ALL) betyder att alla färdiga PARTS spelas. Därefter spelas ACC.

7.11 Kommandot PLAY(PART 1,PART 2) betyder att den del av kompositionen som

börjar med PART 1 och slutar med PART 2 spelas. PLAY(PART 1) betyder spela avsnittet PART 1.

- 7.12 Vill man betrakta den i TEMP uppbyggda klangen som ett klangobjekt med ett eget objektnamn, ger man klangen detta objektnamn genom ordern SAVE(NAME), exempelvis SAVE(ADAM). Motsvarande text lagras då på DEC-tapen.
- 7.13 Vill man i stället för att bygga upp en ny klang i TEMP kalla på en tidigare klang, som man givit ett objektnamn och sålunda gjort om till ett objekt, sker detta med kommandot CALL(NAME), exempelvis CALL(ADAM). Vill man dessutom att texten till objektet ADAM skall skrivas ut på kommentarmediet, sker detta med TOP(NAME) följt av LOOK(NUMBER), där NUMBER är ett heltal, det antal rader man vill titta på.
- 7.14 Man fogar objekt till objekt med kommandot APP, vilket betyder att det i TEMP lagrade objektet läggs efter det objekt som lagrats i ACC. Har man exempelvis först kallat in objektet ADAM 3 till TEMP och sedan lagrat ADAM 3 i ACC med hjälp av ordern MIX(START), så kan man låta ADAM 3 följas av objektet ADAM 4 genom att kalla in ADAM 4 till TEMP följt av kommandot APP:
- | | |
|--------------|--------------|
| TOP(ADAM 3) | TOP(ADAM 4) |
| LOOK(30) | LOOK(50) |
| CALL(ADAM 3) | CALL(ADAM 4) |
| MIX(0) | APP |
- 7.15 Om man är osäker på om det klingar bra att låta ADAM3 följas av ADAM4, kan man först provlyssna på objektföljden med kommandot TRAPP (ordet är sammansatt av TRY och APP), varefter man definitivt kan bestämma sig med ordern APP.
- 7.16 För att få en bättre förståelse för de olika kommandon med vilkas hjälp man orienterar sig i ett redan lagrat material, t ex objekt i det egna objektbiblioteket, införs ett datatekniskt begrepp, nämligen pointer. En pointer pekar på det som skall användas i en kommande operation. Pointern kan fås att peka på första raden i en fil med namnet NAME, och denna fil innehåller - eller är - objektet. Detta sker med kommandot TOP(NAME). Tonsättaren kan sedan göra något med objektet med detta namn, t ex läsa i

rader från det med ett IN-kommando. Man får då in dessa rader från filen och slipper skriva det som står där på kommentarmediet. Inom objektet kan pointern peka ut de olika raderna, och tonsättaren kan då via kommandon göra något med den rad pointern pekar på, t ex med ett SKIP-kommando flytta pointern till nästa rad. Efter TOP(ADAM), där man placerar pointern i början av objektet ADAM, kan man med hjälp av SKIP(50) flytta pointern så att den pekar på rad 51.

Även ett annat tekniskt begrepp bör nu införas. När vi säger att ett objekt finns i TEMP, betyder det att objektet är lagrat på en disk som vi hädanefter kallar TEMP-disken. Att ett objekt är utskrivet på teletypen betyder alltså inte att det därmed automatiskt befinner sig i TEMP. Om vi endast vill titta på vad det var vi hade skrivit i objektet, räcker det med att skriva ut det på teletypen, och då behöver inte objektet befinna sig i TEMP.

Vill vi endast titta på ett objekt gör vi detta med kommandot TOP(NAME), följt av LOOK(NUMBER). På teletypen skrivs då ut det antal rader av objektet som vi bett om. Första raden som skrivs ut är också första raden i objektet. Vill vi emellertid inte få de första 20 raderna utskrivna, utan de tio raderna fr o m 21 t o m 30, skriver vi

```
TOP(NAME)
SKIP(20)
LOOK(10)
```

Likaså kan vi skriva

```
TOP(NAME)
SKIP(30)
LOOK(10)
SKIP(2)
LOOK(5)
```

etc. etc.

Skillnaden mellan att endast titta på objektet och att också ha det i TEMP, är t ex att man inte kan överföra till ACC ett objekt som man endast tittar på. För att kunna operera med kommandon som MIX krävs det att objektet befinner sig i TEMP. Vi placerar ett objekt i TEMP med kommandot CALL(NAME). Hela objektet läses då in på TEMP-disken. Om vi inte behöver hela objektet i TEMP, kan vi använda ordena IN(NUMBER) och SKIP(NUMBER). Ordern SKIP används alltså både när vi vill utesluta något i ett objekt i teletypen, och när vi vill utesluta något i TEMP-disken. Däremot har kommandot LOOK ersatts av kommandot IN, när det är fråga om att läsa in i TEMP-

filen. Vi kan alltså skriva:

```
CALL(ADAM)
```

om vi vill ha in hela objektet på TEMP-disken. Om vi endast vill ha in de första tio raderna, skriver vi

```
TOP(ADAM)
```

```
IN(10)
```

Vill vi först titta på ett objekt för att senare eventuellt endast läsa in delar av objektet på TEMP-disken, kan vi exempelvis skriva så här:

```
TOP(ADAM)
```

```
LOOK(1000)
```

```
TOP(ADAM)
```

```
SKIP(1)
```

```
IN(20)
```

```
SKIP(30)
```

```
IN(27)
```

Raderna 2 + o m 21 plus raderna 52 + o m 78 har då lästs in i TEMP-disken och kan då göras till ett objekt med ett eget namn med ordern SAVE(NAME).

Även om det är nödvändigt att objektet är lagrat i TEMP för att kunna överföras till ACC, så är det inte nödvändigt att objektet befinner sig i TEMP för att vi skall kunna rätta fel i det. Följande felrättningsprocedur kan användas.

I ett objekt är första raden fel. Vi kallar in objektet till displayen med TOP, följt av ett LOOK. Vi observerar att första raden är fel. Med ordern SKIP(1) eliminerar vi första raden och skriver in den korrekta. Därefter ger vi kommandot IN och läser därmed objektet över till TEMP. Alltså:

```
TOP(ADAM)
```

```
LOOK(5)
```

```
SKIP(1)
```

```
CLEAR
```

```
LT(1,440,80,2)
```

```
IN(56)
```

Den femte raden är den korrekta texten. Kommandot CLEAR gör vi om vi vill lägga upp den rättade versionen av objektet i biblioteket igen, men undvika att få med några gamla termer, som eventuellt redan ligger i TEMP.

Ett sista exempel:

Vill man rätta ett fel i ett objekt i sitt objektbibliotek, t ex ha vågform 4 i st f 2 i generator nr 5 som börjar klinga vid LT(2000) och är skriven i rad nr 14 i objektet ADAM3, och utföra denna rättelse på TEMP-disken, så skriver man så här:

1. TOP(ADAM3)
2. LOOK(20)
3. IN(13)
4. SKIP(1)
5. LT(2000)FG(5,263,83,4)
6. IN(27)

Kommentarer:

Rad 2. LOOK skriver ut objektets text på displayen.

Rad 3. Talet i parentesen säger hur många rader vi vill ha inlästa.

Rad 4. Hoppa över den felaktiga raden!

Rad 5. Inför den korrekta texten!

Rad 6. Läs in resten av texten av ADAM 3.

Tag alltid till ett tal i överkant, så att du kan vara säker på att alla raderna kan komma med.

- 7.17 Har vi gjort så stora ändringar att vi vill ge objektet ett nytt namn, gör vi detta med kommandona

SAVE(EVA)

ERASE(ADAM)

vilket betyder att det nya namnet är EVA och att ADAM är utsuddat på DT. (Var försiktig med ERASE eftersom hela objektet ADAM suddas ut, och det ju kan vara trevligt att vara helt säker på att man inte behöver det mera.)

- 7.18 PART(NUMBER) och END

Tonsättaren kan foga ihop sina objekt till delar av en komposition som här kallas PART(NUMBER). Tonsättaren måste av tekniska skäl alltid börja uppbyggandet av den första klangen med kommandot PART(NUMBER), exempelvis PART(ONE). Vidare får tonsättaren av tekniska skäl inte använda fler än 7.000 termer inom varje PART av kompositionen. En PART avslutas med kommandot END. Det finns inga tekniska hinder för att tonsättaren är frikostig med det antal PART han vill dela upp sin komposition i. Efter END finns avsnittet på magnetband under det namn, man givit efter PART, och i sådan form, att det omedelbart kan spelas i studion. Nästa PART kommer omedelbart efter END, och behåller alla värden som föregående PART lämnade efter sig.

7.19 Med kommandot STDTIM - standard time - kan tonsättaren variera det tempo han vill ha kompositionen uppspelad med. Referensen för tempot skrivs som STDTIM(1000)

varmed menas att kompositionen spelas upp i det tempo han själv har bestämt genom angivande av sina Local Time-termer. Talet 1000 står här för 1000 ms, och termen betyder att programmerade 1000 ms skall spelas upp under loppet av 1000 ms, alltså utan några modifieringar.

Skriver man däremot

STDTIM(2000)

så har man fördubblat tempot eftersom 2000 ms nu skall spelas upp inom loppet av 1000 ms. Motsvarande kommando för att spela kompositionen med halv hastighet är givetvis

STDTIM(500)

STDTIM kan alltså betraktas som ett metronomtempo, närmare bestämt antalet "taktslag med längden en formell millisekund" under en sekund verklig tid. STDTIM har betydelse endast för spelkommandona och gäller för alla uppspelningar tills man åter ändrar den.

Realisera de två korta kompositioner som beskrivs i nedanstående diagram.

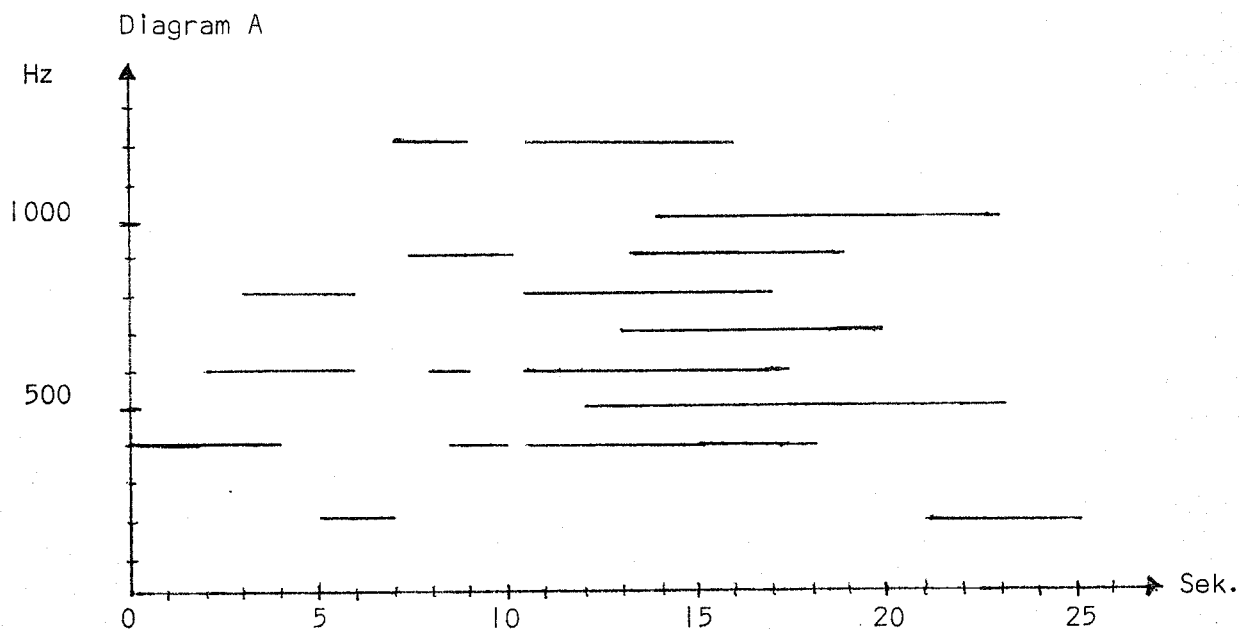
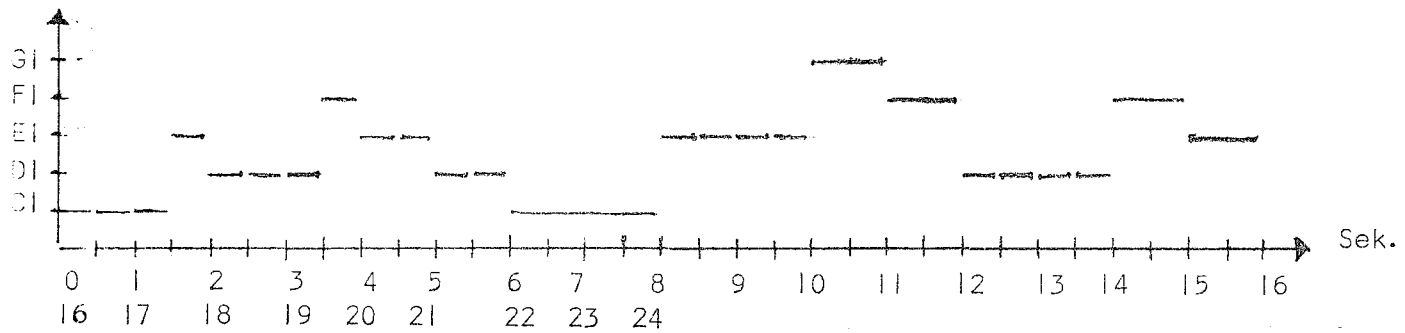


Diagram B



Lösning, diagram A:

PART(TWO)

LT(0)

FG3&FG6&FG9&FG12>CHA(1,100);

FG(1,100)FG(2,200)FG(3,300)FG(4,400)FG(5,500);

FG(6,600)FG(7,700)FG(8,800)FG(9,900)FG(9,900)FG(10,1000);

FG(11,1100)FG(12,1200);

LT(0) FG(4,,80)>D(4000)

LT(2,) FG(6,,80)>D(4000)

LT(3,) FG(8,,80)>D(3000)

LT(5,) FG(2,,80)>D(2000)

LT(7,) FG(12,,80)>D(2000)

LT(7,500) FG(9,,80)>D(2500)

LT(8,) FG(6,,80)>D(1000)

LT(8,500) FG(4,,80)>D(1500)

LT(10,500) FG(4,,80)>D(7500)

FG(6,,80)>D(7000)

FG(8,,80)>D(6500)

FG(12,,80)>D(5500)

LT(12,) FG(5,,80)>D(11000)

LT(13,) FG(7,,80)>D(7000)

LT(13,500) FG(9,,80)>D(5500)

LT(14,) FG(10,,80)>D(9000)

LT(21,) FG(2,250,80)>D(4000)

MIX

END

8. Makron och variabler8.1 Makron

För att realisera diagram B, går man tillväga enligt följande.

CI, DI, o s v står för toner i ettstrukna oktaven.

PART(ONE)

LT(0) FG3>CHA(1,100);

FG(1,262,80,3)>D(400);

LT(500) FG(1,,80)>D(400);

LT(1,) FG(1,,80)>D(400);

LT(1,500) FG(3,330,80)>D(400);

LT(2,) FG(2,294,80,3)>D(400);

LT(2,500) FG(2,,80)>D(400);

Vi har bara hunnit tre sekunder och märker att detta börjar bli arbetsamt. Finns det ingen genväg?

Ett fåtal frekvenser uppträder, ett fåtal durationer likaså. Man kan för var och en av dessa återkommande klanger definiera s k makron för motsvarande termer eller termgrupper. Med hjälp av makron skulle sekvensen få nedanstående utseende.

I raderna 2-6 ger man namnen CI, DI, o s v åt de fem FG-termer, som behöver användas. I rad 7-9 ges namn åt de tre D-termer, som behövs. I fortsättningen kan man skriva CI i stället för hela termen, o s v och man får med mindre skrivarbete ett läsligare program.

1 PART(ONE)

2 CI="FG(1,262,80,3)";

3 DI="FG(2,294,80,3)";

4 EI="FG(3,330,80,3)";

5 FI="FG(4,349,80,3)";

6 GI="FG(5,392,80,3)";

7 DUR19="D(1900)";

8 DUR9="D(900)";

9 DUR4="D(400)";

LT(0)FG(1←6)>CHA(1,100);

CI>DUR4;

LT(500)CI>DUR4;

LT(1,)CI>DUR4;

LT(1,500)EI>DUR4;

LT(2,)DI>DUR4;

```

LT(2,500)DI>DUR4;
LT(3,)DI>DUR4;
LT(3,500)FI>DUR4;
LT(4,)EI>DUR4;
LT(4,500)EI>DUR4;
LT(5,)DI>DUR4;
LT(5,500)DI>DUR4;
LT(6,)CI>DUR19;
SAVE(ADAM)
CLEAR
LT(0)FG(1>6)>CHA(1,100);
EI>DUR4;
LT(,500)EI>DUR4;
LT(1,)EI>DUR4;
LT(1,500)EI>DUR4;
LT(2,)GI>DUR9;
LT(3,)FI>DUR9;
LT(4,)DI>DUR4;
LT(4,500)DI>DUR4;
LT(5,)DI>DUR4;
LT(5,500)DI>DUR4;
LT(6,)FI>DUR9;
LT(7,)EI>DUR9;
SAVE(BERTIL)
CLEAR
CALL(ADAM)
MIX
CALL(BERTIL)
MIX(8000)
CALL(ADAM)
MIX(16000)
PLAY(MIX)
END

```

Här definierar vi de två objekten ADAM och BERTIL. Därefter görs ett CLEAR för att få TEMP-filen nollställd som utgångsläge. Symboltabellen påverkas inte av CLEAR och makrodefinitionerna finns därför fortfarande kvar. Med CALL(ADAM) läses klangobjektet ADAM.

Vid MIX förs ADAM till ACC. Effekten blir också som efter ett CLEAR, och

efter CALL(BERTIL) blir ljudobjektet BERTIL det enda, som står i TEMP. Med MIX(8000) infogas BERTIL 8 sekunder in i ACC. Ljudobjektet ADAM återkommer efter 16 sekunder från början av ACC.

- 8.2 Ett makro definieras på följande sätt:
 makronamn = "ett stycke text som kan vara termer eller termgrupper";
 Namnet skall bestå av maximalt sex bokstäver och/eller siffror varav det första en bokstav.

Övningsuppgift 4

Vi vill skapa en tonika-dominant-subdominant sekvens (T-D-T-S) och vill att denna sekvens skall klinga med tre olika nivåer, 70, 80 och 90 dB. I appendix 3 finns en frekvenstabell.

Lösning

```
PART(TWO)
FREMAT="FG(1,262,,3);FG(2,330,,3);FG(3,392,,3)";
FREMAD="FG(3,392,,3);FG(2,494,3);FG(1,587,,3)";
FREMAS="FG(1,262,,3);FG(2,208,,3);FG(3,175,,3)";
NIV1="FG(1<-3,,70)";
NIV2="FG(1<-3,,80)";
NIV3="FG(1<-3,,90)";
NIV="NIV1";
SEKVMA="LT(0);NIV;FREMAT;LT(2,500)FREMAD;
LT(4,)FREMAT;LT(6,)FREMAS;FG(1<-3)>D(1500)";
LT(0)FG3>CHA(1,100);
SEKVMA
MIX(0)
NIV="NIV2"
SEKVMA
MIX(7500)
NIV="NIV3"
SEKVMA
MIX(15000)
PLAY(MIX)
END
```

Observera att NIV och SEKVMA är makron, som i sin tur anropar andra makron. Då SEKVMA anropas, anropar den i sin tur NIV, som anropar NIV1, som ställer nivån till 70 dB i de tre första frekvensgeneratorerna. Kontrollen återlämnas till SEKVMA, som sedan anropar FREMAT o s v. TEMP sorteras över

till ACC med MIX(\emptyset). Efter detta är TEMP CLEARad, men makrodefinitionerna finns fortfarande kvar i symboltabellen. Makrot NIV definieras om och utför NIV2, nästa gång det används av SEKVMA. Vid MIX(75 $\emptyset\emptyset$) sorteras det som SEKVMA åstadkommit, d v s en ackordserie, nu med nivån 80 dB in i ACK. Den tredje ackordsföljden realiserar på samma sätt med NIV3 och MIX(15 $\emptyset\emptyset\emptyset$).

8.3 OLD

I det fall startvärdet för en envelop- eller glissandoterm sammanfaller med föregående terms (slut-)värde kan man skriva OLD i stället. Om man vill förskjuta värdet från föregående term, t ex med 10 uppåt eller neråt, så får man skriva OLD +10 resp OLD -10.

Exempel:

LT(0)

FG(1,100,0)>ENV(OLD,100,500,2)>ENV(OLD,OLD,100)>

ENV(OLD,50,500,2)>Z;

LT(1800)FG(1,OLD-50,100)>ENV(OLD+10,0,600,3);

Observera att OLD inte är en variabel och därför endast får användas i ENV-, GLIS-, och apparattermer.

8.4 Konstanter

Man har i EMS-2 möjligheten att i st f att explicit ange siffervärden, beteckna dessa med en symbol, d v s namnge siffervärdet. Detta görs på följande sätt:

KALLE=44 \emptyset ;

ANDERS=23

A1=44 \emptyset

MF=7 \emptyset ;F=8 \emptyset ;FFM=9 \emptyset ;

A \emptyset =22 \emptyset ;F \emptyset =175;CI=262;

E1=33 \emptyset ;G1=392;H1=494;

D2=582;

Lösningen till uppgiften ovan kan då utformas så här:

FREMAT="FG(1,CI,,3)FG(2,FI,,3)FG(3,G1,,3)";

FREMAH="FG(3,G1,,3)FG(2,HI,,3)FG(1,D2,,3)";

FREMAS="FG(1,CI,,3)FG(2,A \emptyset ,,3)FG(3,F \emptyset ,,3)";

NIV1="FG(1 \leftarrow 3,,MF)";

NIV2="FG(1 \leftarrow 3,,F)";

NIV3="FG(1 \leftarrow 3,,FFM)";

8.5 Variabler

Man kan om man vill tilldela siffervärden till namn och använda namnet på siffrornas plats.

```
AB=100;
FG(1,AB);
AB=200;
FG(1,AB);
```

Om man skriver AB=17 är variabeln AB skyddad från vidare ändringar. Speciellt användbar är denna möjlighet att ändra värden i samband med makron. Vi illustrerar detta genom att realisera exemplet i 6.1 (EMS-1) på ett enklare sätt:

```
PART(ONE)
CLEAR
LT(Ø)
MI="FG(NR,FREQ,NIV,Ø)>CHA(NO,1ØØ);";
NR=1;FREQ=2ØØ;NIV=7Ø;NO=1;
MI;
NR=4;FREQ=3ØØ;NIV=67;NO=2;
MI;
NR=7;FREQ=4ØØ;NIV=64;NO=3;
MI;
NR=1Ø;FREQ=6ØØ;NIV=61;NO=4;
MI;
LT(3Ø,)CHA(1)>Z;
SAVE(ADAM)
MIX(Ø)
PLAY(MIX)
END;
```

Övningsuppgift 5

Realisera ackordsexemplet på likartat sätt genom att använda 3 makron i st f 8 stycken.

Lösning

```
PART(TWO)
FREMAA="FG(1,FREQ1,,3)FG(2,FREQ2,,3)FG(3,FREQ3,,3)";
NIVAA="FG(1>3,,DB)>CHA(1,1ØØ)>D(75ØØ);"
SEKVMA="LT(Ø)NIVAA;FREQ1=262;FREQ2=33Ø;
```

```

FREQ3=392;FREMAA;
LT(2,500)FREQ2=294;FREQ1=587;FREMAA;
LT(4,)FREQ1=262;FREQ2=330;FREMAA;
LT(6,)FREQ2=208;FREQ3=175;FREMAA;"
LT(0)
DB=70;SEKVMA;
MIX(0)
DB=80;SEKVMA;
MIX(7500)
DB=90;SEKVMA;
MIX(15000)
PLAY(MIX)
END

```

Den s k makroexpansionen av SEKVMA ser ut som följer:

```

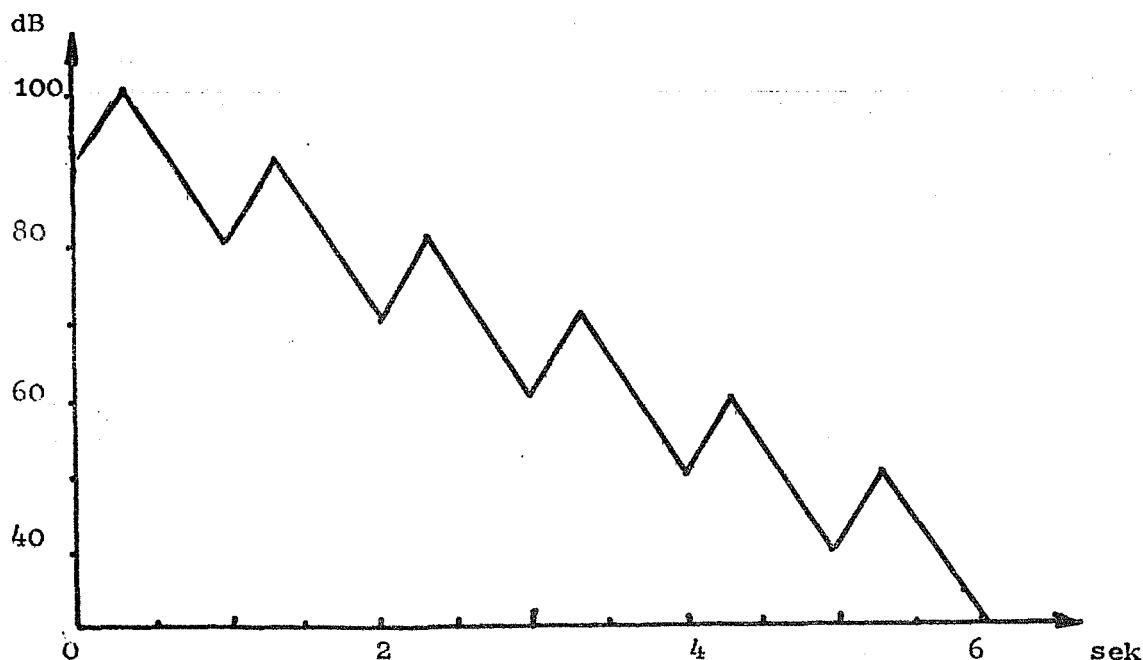
LT(0)
FG(13..DB)>CHA(1,100)>D(7500);
FREQ1=262;FREQ2=330; o s v

```

d v s allt som använda makron har utfört. Med MIX(7500) kommer därför allt att sorteras in igen med början vid 7.5 sekunder, men vi ger först variabeln DB ett nytt värde, 80. På samma sätt sorteras ackordsföljden in vid 15 sekunder med nivån 90. En variation är också, att durationen sätts på en CHA-term i nivåmakrot.

Övningsuppgift 6

Realisera nedanstående. Förbjudet att använda andra variabler än OLD!



Lösning:

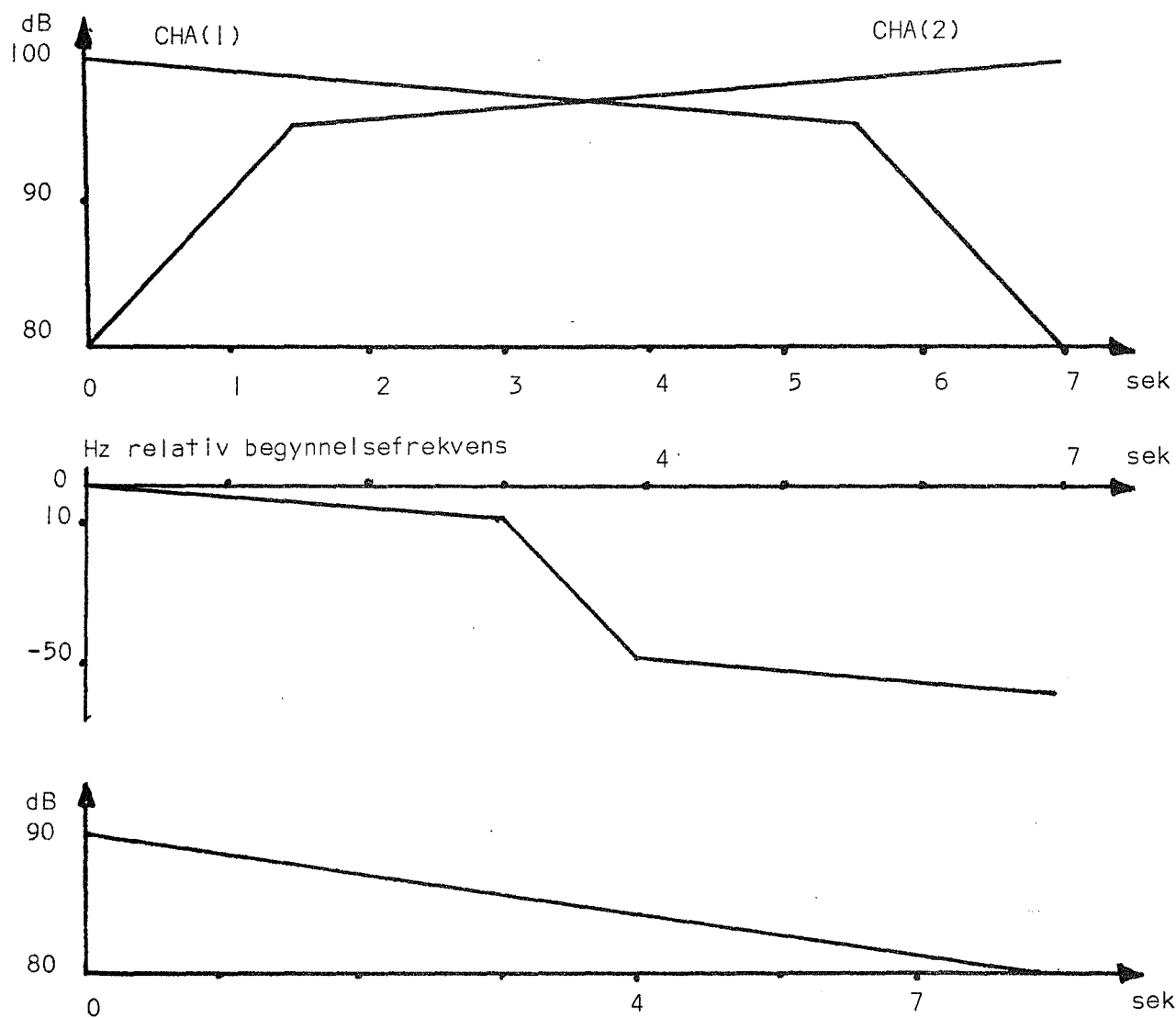
```

PART(THREE)
CLEAR
MI="ENV(OLD,OLD+10,300)>ENV(OLD,OLD-20,700)";
LT(0)
FG(1,100,90,6)>MI>MI>MI>MI>MI>CHA(1,100);
SAVE(DESCND)
MIX
PLAY(MIX)
END

```

Övningsuppgift 7

Realisera exemplet:



Kanal 1 och kanal 2 skall variera mellan 80 och 100 dB enligt den första figuren. Frekvensgeneratorerna ges följande begynnelsefrekvenser:

FG(1) 400 Hz

FG(2) 590 Hz

FG(3) 780 Hz

Var och en av dessa tre frekvenser minskas med 60 Hz enligt den andra figuren.

Nivån på de tre frekvensgeneratorerna går från 90 till 80 dB under de sju sekunder ljudet varar.

Lösning

PART(FOUR)

CLEAR

LT(Ø)

FG(1,4ØØ)&FG(2,59Ø)&FG(3,78Ø)>

ENV(9Ø,8Ø,7ØØØ)>

GLIS(OLD,OLD-1Ø,3ØØØ)>

GLIS(OLD,OLD-4Ø,1ØØØ)>

GLIS(OLD,OLD-1Ø,3ØØØ);

FG3>CHA(1)>ENV(1ØØ,96,55ØØ)>ENV(96,8Ø,15ØØ);

FG3>CHA(2)>ENV(8Ø,96,15ØØ)>ENV(96,1ØØ,55ØØ);

SAVE(SWOCHE)

MIX

PLAY(MIX)

END

SAMMANFATTNING AV ELEMENT I EMS-2

Apparattermer

AM(NR, INGÅNG, NIVÅ)
AMP(NR, NIVÅ)
CD(NR, KANAL, NIVÅ)
CHA(NR, NIVÅ)
FF(NR, FILTERKANAL, NIVÅ)
FG(NR, FREKVENNS, NIVÅ, VÄGFORM)
NG(NR, NIVÅ, FÄRG1, FÄRG2)
REV(NR, EKOTID, NIVÅ)
RM(NR, INGÅNG, NIVÅ)

Studiopunkter

FS
FG3, FG6, FG9, FG12, FG15, FG18, FG21, FG24

Kommandon

APP
CLEAR, CLEAR(MIX), CLEAR(ALL)
DELETE(SYMBOL)
KEEP, KEEP(MIX), KEEP(ALL)
MIX(TID)
STDTIM(MS)

TEMP-kommandon

CALL(FILNAMN)
ERASE(FILNAMN)
IN(RADANTAL)
LOOK(RADANTAL)
PLAY, PLAY(MIX), PLAY(PARTNAMN), PLAY(PART1, PART2)
REPL(FILNAMN) (fungerar som ERASE följt av SAVE)
SAVE(FILNAMN)
SKIP(RADANTAL)
TOP(FILNAMN)
TRAPP
TRY, TRY(MS), TRY(SEK, MS)

Ett kommando bör stå på en egen rad.

Lokal tid

LT(MS), LT(SEK,MS)

Observera att LT(1,5) betyder detsamma som LT(1,005), ej LT(1,500).

Envelopptermer

ENV(NIVÅ,NIVÅ,MS,KURVFORM,STEG)

Z fungerar som ENV(OLD,Ø,|Ø,Ø,|Ø)

D(MS) fungerar som ENV(OLD,OLD,MS)>Z

Glissandotermer

GLIS(FREKVENSS;FREKVENSS,MS,KURVFORM,STEG)

Steptermer

ESTEP(MS)

GSTEP(MS)

EXIT avbryter körningen och startar upp EMS-1 för ett nytt program.

>, #, & står mellan element i kopplingskedjor, envelopkedjor och glissandokedjor. Om kedjan fortsätter över flera rader skall den avdelas så att ett av dessa tecken står sist på raden.

; bör avsluta varje kedja,

← får användas i FG- och FF-termer.

'KOMMENTAR' kommentarer bör användas relativt flitigt. Man glömmar annars mycket snart vad ett program handlar om.

MAKRONAMN="...TEXT..." makrodefinition. Inget ytterligare får stå på samma rad efter sluttecknet.

MEX får ingå i makrotext (MAKRO-EXIT).

Makron får anropa varandra upp till 100 nivåer. Ett makro får inte innehålla CALL.

↑ aritmetisk operation för exponentiering.

EXEMPEL PÅ FELUTSKRIFTER I EMS-2

```
:APCDEFFG=7
      >
04    001          SYMBOL WITH MORE THAN 6 CHARACTERS.
!ABCDEF=7
!
:120000
 120000
04    002          INTEGER WITH MORE THAN 5 FIGURES
!
:FG(1,100.32.3)
  FG(1,100.32  )
04    003          TWO DECIMAL POINTS
!
:ARCDEF=ADB
  ABCDEF=ADF
04    020          SYMBOL NOT DEFINED.
!
:LOOK
  LOOK
04    035          TOP NOT EXECUTED OR SEC. FILE EMPTY
!
:FG(17.2,100)
  FG(17.2,  )
02    012          ILLEGAL PARAMETER
!

:PART(T4)
:LT(0)
:FG(1,1000)
:MIX
  MIX
01    041          DURATION OF MIX =0!
!END
  END
01    010          NO SORTRECORDS ON THE 'MIX' DISK
```

```

:=3
04 100 A '=' IS POSSIBLE ONLY AFTER A SYMBOL OR A '='.
!
:IN=17
IN=17
04 101 ATTEMPT TO ASSIGN A FIXED SYMBOL
OR A SYMBOL STARTING WITH A NONALPHABETIC CHAR
!JN=17
!
:FG(12"100)
)
04 104 A '"' FOUND IN A NON-MACRO TEXT.
!
:PLAY
PLAY
01 004 NOTHING TO PLAY
!MIX
MIX
01 042 NO RECORDS ON THE TEMP DISK
!
:PART(T3)
:LT,
LT,
02 011 ILLFGAL DELIMITER
!
:LT(0)
:FG17,100)
FG17 )
04 020 SYMBOL NOT DEFINED.
!
:KEEP
KEEP
01 010 KEEP WITHOUT MEANING HERE
!
:CLEAR(17)
CLEAR(17)
01 007 THE ONLY POSSIBLE PARAMETERS TO 'CLEAR' ARE
'MIX' OR 'ALL'.
!
:KEEP(13)
KEEP(13)
01 009 THE ONLY POSSIBLE PARAMETERS TO 'KEEP' ARE
'MIX' OR 'ALL'.
!
:STDTIM-4
STDTIM-4
01 024 LEFT PARANTHESIS MISSING IN 'STDTHM'-TERM
!
:CLFAR
:FG(15,,80)>GLIS(220,330,4000);
:SAVE(FG15GL)
:CLFAR
:FG(15,,90)>GLIS(330,220,4000);
:SAVE(FG15GL)
SAVE(FG15GLSRC)
01 030 FILE ALREADY PRESENT
!

```


EN KÖRNING UTAN FEL

KM15 V5X
\$A DK7 -4,1/DK6 -15/DK5 -14/DK4 -10/DT1 2,6/DT2 3/LP 5/MT 7,10
\$E FMS1

EXECUTE V4A

FMS1 T1.3
NAME OF INPUT FILE?
Carriage return
INTERACTIVE MODE? YES OR NO?

YES

:PART(ONE)
:LT(0)
:FG(1,440)>ENV(85,40,3000)>CHA(4,100);
:MIX
OBJECT NR: 1. BLOCK TIME: 0.000 3.000 FREE MIXRFC.S: 13320
:FG(1)>ENV(85,40,1000);
:AFP
OBJECT NR: 2. BLOCK TIME: 0.000 4.000 FREE MIXRDC.S: 13319
:END
BLOCKLABEL:ONE.GLOBAL TIME: 0.000 4.000 FREE MIXRFC.S: 13319
:PART(TWO)
:LT(0)
:FG(1,330)>ENV(85,40,2000);
:MIX(1000)
OBJECT NR: 1. BLOCK TIME: 0.000 3.000 FREE MIXRFC.S: 13322
:END
BLOCKLABEL:TWO.GLOBAL TIME: 4.000 7.000 FREE MIXRFC.S: 13322
:EXIT
-----EXIT. END OF THIS RUN.-----

FMS1 V1.3
NAME OF INPUT FILE? (ALT MODE)
---- END OF RUN. EXIT TO MONITOR ----

KM15 V5X
\$

EMS-38.5 Aritmetiska operationer

Förutom möjligheten att tilldela ett tal till en symbol, så är det också möjligt att ge en symbol ett värde, som är ett resultat av ett aritmetiskt uttryck.

Exempel:

A1=20; A2=23; A3=5;

A4=A2-A1+A3; A4=A4-5;

medför att A4 blir lika med 3.

Särskilt användbart är detta förfarande tillsammans med makron. En variabel kan då ökas eller minskas med en viss kvantitet varje gång ett visst makro anropas.

Vi återvänder till exemplet i 6.1.

```

1  PART(ONE)
2  CLEAR
3  MI="FG(NR,FREQ,NIV,Ø)>CHA(NO,1ØØ)>D(3ØØØØ);
4  NR=NR+3;NO=NO+1;NIV=NIV-3;
5  LT(Ø)NR=1;NIV=7Ø;NO=1;FREQ=2ØØ;
6  MI
7  FREQ=3ØØ;MI;
8  FREQ=4ØØ;MI;
9  FREQ=6ØØ;MI;
1Ø SAVE(ADAM)
11 MIX(Ø)
12 PLAY(MIX)
13 END

```

Rad 3-4 Makrot definieras som i 8.5 med ett tillägg bestående av:

NR=NR+3; NO=NO+1; NIV=NIV-3;

Detta tillägg medför att varje gång makrot användes ökas värdet av NR med 3, NO ökas med 1 och NIV minskas med 3, vilket ger dessa variabler de värden som skall stå i FG-uttrycket nästa gång makrot används.

Rad 5 Här sätts NR=1; NO=1; NIV=70; FREQ=200;

Dessa värden används i makrot vid anropet på rad 6 och förändras därpå automatiskt till 4,2 resp 67.

Rad 7 FREQ sätts till 300 och används tillsammans med värdena
 NR=4, NO=2 och NIV=67 i makrot. NR, NO och NIV ändras därmed
 till resp 7,3,64 o s v.

Tiden anges i EMS-I med klockslag, d v s i absolut tid. Man kunde i
 vissa fall i stället önska sig en relativ tid, d v s kunna ange hur lång
 en viss tidsangivelse skall gälla utan att behöva räkna ut tidpunkten
 för nästa händelse.

Vi skall nu visa hur man med hjälp av variabler som automatiskt inkra-
 teras i ett makro kan använda relativ tidsangivelse och därvid dessutom
 med hjälp av ett makro på högre nivå kan representera en hel sekvens av
 händelser att utföras vid skilda tidpunkter.

8.6 Exempel på automatisk uppdatering i ett makro

Antag att ljudobjektet inte är längre än 99999 ms. Lokal tid kan då ang
 i enbart millisekunder. Antag vidare att vi vill ha ett makro som till-
 sammans med variablerna TICK och TACK fungerar på följande sätt: I börj
 av ljudobjektet nollställs TICK. Objektet förutsätts börja med ett LT(0)
 TICK ökas sedan vartefter och betecknar absolut tid inom ljudobjektet.

PART(ONE)

LT(0)

'definition av makrot RELTID'

TICK=0;

'andra makrodefinitioner, operationer o s v att utföras vid tidpunkter
 noll'

TACK=tiden fram till nästa händelse

RELTID;

andra makroanrop och operationer som skall utföras vid denna tidpunkt

TACK=tiden fram till nästa händelse

RELTID

o s v

Övningsuppgift 8

Beskriv makrodefinitionen för RELTID.

Lösning

Makrot RELTID ser ut på följande sätt:

RELTID = "TICK = TICK + TACK ; LT(TICK)"

Man kan givetvis blanda användningen av relativ tid och absolut tid, eller
 nollställa TICK och börja beskriva en parallell händelseföljd.

Övningsuppgift 9

Använd makrot RELTID i ackordsexemplet i 8.2. Tonikan är vid udda tillfällen 2.5 sekunder lång (d v s 1:a, 3:e, 5:e ... gången) och vid jämna tillfällen 2 sekunder lång. Dominanten är liksom subdominanten 1.5 sekunder lång.

```

PART(TWO)
.
.
.   som tidigare
.
.
SEKVMA="RELTID;NIV;FREMAT;
TACK=2500;RELTID;FREMAD;
TACK=1500;RELTID;FREMAT;
TACK=2000;RELTID;FREMAS;"
TICK=0;TACK=0;SEKVMA
NIV="NIV2"
TACK=1500;SEKVMA;
NIV="NIV3"
TACK=1500;SEKVMA;
FG(1<-3)>D(1500);
MIX;
PLAY(MIX)
END;

```

Vi kan också stoppa in TACK=...;RELTID i makrodefinitionerna:

```

.
.
.
.
.
FREMAT="FG(1,262,,3)FG(2,330,,3)
FG(3,392,,3);RELTID"
FREMAD="FG(3,392)FG(2,494)FG(1,587);TACK=1500;RELTID"
FREMAS="FG(1,262)FG(2,208)FG(3,175);TACK=1500;RELTID"
SEKVMA="NIV;TACK=2500;FREMAT;FREMAD;TACK=2000
      FREMAT;
      FREMAS"
.
.
.

```


VARIABEL är (enligt tidigare) en symbol om högst sex alfanumeriska tecken, varav det första är en bokstav.

IFPOS(VARIABEL) betyder att det som står mellan högerparentesen,), och närmaste <-symbol göres då variabeln är större än noll. Sammaledes för IFZER(.....) där motsvarande sträng utföres då variabeln inom parenteserna har värdet noll och likaledes för IFNEG(VARIABEL) där strängen utföres endast då variabeln är mindre än noll.

Exempel:

NR=1;NO=0;

a) IFPOS(NR)NR=NR-1;NO=NO+1;<

b) IFPOS(NR)NR=NR-1;NO=NO+1;<

I a) är NR=1, d v s större än noll vilket medför att strängen utföres och NR blir lika med 0 och NO blir 1.

I b) är NO=0 d v s inte större än noll, vilket innebär att strängen inte utföres och programmet fortsätter efter <-symbolen.

Om vi vill att NO skall bli lika med 1000 för NR=0 och endast för NR=0 kan vi skriva:

IFZER(NR)NO=1000;<

Det allmänna uttrycket för konditionaler eller villkorssatser är:

KONDITIONAL(VARIABEL) STRÄNG ATT UTFÖRAS OM;
VILLKÖRET UPPFYLLETT avslutningssymbol

Programmet fortsätter efter avslutningssymbolen om konditionalen ej skall utföras.

Vi återvänder till det inledande exemplet.

RELTID: För att undvika att millisekuddelen blir för stor, kan vi testa dess värde och minska dess värde med 1000 om millisekuddelen är större än 1000, samtidigt som sekuddelen ökas med 1.

- 1) RELTID="TICK=TICK+TACK-1000;
- 2) IFPOS(TICK)STICK=STICK+1;<
- 3) IFZER(TICK)STICK=STICK+1;<
- 4) IFNEG(TICK)TICK=TICK+1000;<
- 5) LT(STICK,TICK)";

Rad 1 TICK sättes lika med TICK+TACK (TACK är tiden till nästa händelse). Detta är som tidigare men dessutom subtraheras därefter 1000 (ms).

Rad 2 Om resultatet fortfarande är större än eller lika med noll, d v
 - 3 TICK var större än 1000 innan subtraktionen utfördes, ökas sekun-
 delen STICK (som antas redan ha ett värde) med 1 (sek.) och följ-
 aktligen kan millisekuddelen hållas nere genom att sekuddelen
 utnyttjas.

Rad 4 Om TICK+TACK var mindre än 1000 blir resultatet efter subtrak-
 tionen med 1000 negativt och vi kan (och måste!) återställa det
 gamla värdet genom att addera 1000 till TICK.

Huvuddelen av exemplet bestod i att upprepa makrot SEKVMA 25 gånger. Detta
 kan nu lösas på följande enkla sätt genom att införa ett makro på högre
 nivå och en räknare.

```

.
.
.
.
RAEKN=25;TICK=Ø;STICK=Ø;
T25="IFPOS(RAEKN)RAEKN=RAEKN-1;SEKVMA;
      T25;<"

```

Detta innebär att om RAEKN är större än noll minskas RAEKN med 1.
 Därefter utföres SEKVMA och proceduren upprepas tills RAEKN är lika med
 noll.

Om man inte använt sig av en variabel någon enda gång kan denna ej ha fått
 sig tilldelat något värde och kallas därför odefinierad. Det finns två
 konditionaler som undersöker huruvida en symbol är definierad eller ej.
 Dessa är:

```

IFUND(SYMBOL)ÅTGÄRD;<om odefinierad ... ..
IFDEF(SYMBOL)ÅTGÄRD;<om definierad ... ..

```

Om man anser det praktiskt att slippa definiera ett makros variabler inna
 det anropas, kan man använda sig av ovanstående konditionaler.

En praktisk användning av detta ger RELTID dess slutgiltiga utformning:

```

RELTID=IFUND(STICK)STICK=Ø;<
      IFUND(TICK)TICK=Ø;<

```

```

IFUND(TACK)TACK=Ø;◀
TICK=TICK+TACK-1ØØØ;
IFPOS(TICK)STICK=STICK+1;◀
IFZER(TICK)STICK=STICK+1;◀
IFNEG(TICK)TICK=TICK+1ØØØ;◀
LT(STICK,TICK)"

```

Nu kan man ersätta LT(Ø) med RELTID, ty om variablerna STICK, TICK och TACK är odefinierade då RELTID anropas, definieras och nollställs dessa och man erhåller följaktligen tidsargumentet Ø i LT-termen.

Om man av någon anledning vill att en variabel (eller ett makro) skall tas bort ur den tabell, symboltabellen, där alla använda namn och motsvarande värden finns lagrade, finns det en möjlighet att göra detta. Kommandot härför är DELETE(NAMN).

Man kanske vill använda makrot RELTID i varianten med IFUND och i flera ljudobjekt eller för parallella förlopp. Då vill man förmodligen att STICK och TICK skall nollställas. Ett sätt att göra detta är att göra dem odefinierade, så att RELTID automatiskt definierar dem igen och sätter värdet till Ø. Kommandot blir DELETE(STICK,TICK);

10. Några ytterligare kommandon

Kommandon av typ WRITE(VARIAB) kan användas för att kontrollera exempelvis förloppet hos komplicerade makron.

Exempel:

```

EMS1 V1.4

NAME OF INPUT FILE?

INTERACTIVE MODE? YES OR NO?

```

YES

```

:TOP(ACVIBF)
:LOOK(88)
ACCVIB="LT(TID)FG(1)>GLIS(434,446,GTID)>GLIS(446,434,GTID);
WRITE(TID,GTID);
SLASK=GTID/2Ø;
GTID=GTID-SLASK;
TID=TID+GTID;
TEST=GTID-5Ø;
IFPOS(TEST)ACCVIB<:"
Ø5      ØØ2      END OF FILE ON ADDITIONAL INPUT FILE.

```



```
:CALL(ACVIBF)
```

```
05 002 END OF FILE ON ADDITIONAL INPUT FILE.
```

```
:PART(ACVIBF)
```

```
:LT(0)FG(1,,80)>CHA(4,100);
```

```
:TID=0;GTID=400;
```

```
:ACCVIB;
```

```
'TID = 0 TYPE: 01 '
'GTID = 400 TYPE: 01 '
'TID = 380 TYPE: 01 '
'GTID = 380 TYPE: 01 '
'TID = 741 TYPE: 01 '
'GTID = 361 TYPE: 01 '
'TID = 1084 TYPE: 01 '
-----
```

```
'GTID = 55 TYPE: 01 '
'TID = 6981 TYPE: 01 '
'GTID = 53 TYPE: 01 '
'TID = 7032 TYPE: 01 '
'GTID = 51 TYPE: 01 '
-----
```

```
:PLAY
```

Med kommandot MESS(UTSKRIFT) åstadkommer man att programmet stannar, skriver ut texten UTSKRIFT, sätts i felsättningsmode och väntar på kommandon från avsändaren.

Exempel:

```
EMS1 V1.4
NAME OF INPUT FILE?
```

```
INTERACTIVE MODE? YES OR NO?
```

```
..S
```

```
:TOP(GLIMSF);LOOK(99);
```

```
GLIMES="IFUND(TID)TID=0;FN=1;FR=55;<
```

```
FN=FN-1;
```

```
IFZER(FN)FN=3<;
```

```
MESS(NEXT FR)
```

```
IFPOS(FR)
```

```
LT(TID)
```

```
FG(FN)>GLIS(FR,440,3000);
```

```
TID=TID+1000;
```

```
GLIMES<;"
```

```
05 002 END OF FILE ON ADDITIONAL INPUT FILE.
```

```
:IN(99)
```

```

05 002      END OF FILE ON ADDITIONAL INPUT FILE.
:PART(MESDEM)
:LT(0)FG(1>3,,80)>CHA(2,100);
:GLIMES
NEXT FR
!FR=220
!
NEXT FR
!FR=110
!
NEXT FR
!FR=55
!
NEXT FR
!FR=83
!
NEXT FR
!FR=110
!
NEXT FR
!FR=165
!
NEXT FR
!FR=220
!
NEXT FR
!
NEXT FR
!
NEXT FR
!PLAY
!FR=1760
!
NEXT FR
!FR=1320
!
NEXT FR
!FR=880
!
NEXT FR
!FR=660
!
NEXT FR
!
NEXT FR
!FR=440
!
NEXT FR
!
NEXT FR
!FR=0
!
:PLAY
:MIX
:END

```

I icke-interaktivt mode (om man svarar NO på frågan INTERACTIVE MODE? YES OR NO?) är MESS ej i funktion och WRITE skriver ej på kommentarmediet utan enbart på listmediet (logisk enhet 5).

EMS1 V1.4
 NAME OF INPUT FILE?

INTERACTIVE MODE? YES OR NO?

YES

```
:PART(ONE)
:MM="MS;MM"
:MS="  - - - "
:WRITE(CNT);
: CNT= CNT- 1;
:IFZER(CNT)MEX<"
: CNT= 2;
:MM;
' CNT = 2 TYPE: 01 '
' CNT = 1 TYPE: 01 '
:
:'PAA DETTA SETT KAN ALLTSAA ETT UNDERORDNAT
:MAKRO AVBRYTA ETT OEVERORDNAT'
```

11. ^ LIST och NOLIST

Den fil som bildats vid SAVE kan givetvis innehålla makroanrop. Det finns ett sätt att få bättre kontroll över hur makron utvecklas. Man kan nämligen med kommandot LIST i texten få, inte makroanrop, utan de termer som makron utför. Om makrot anropar ytterligare makron får man dessa termer, o s v.

Man kan härigenom enklare se konsekvenserna av ett makroanrop, om detta eljest är svårt att genomskåda.

Kommandot NOLIST upphäver expansionen. Exempel:

NAME OF INPUT FILE?

INTERACTIVE MODE? YES OR NO?

YES

```
:PART(BONE)
:BMAK="FG(17,131); "
BS="FG(18,262)"
:LIST
:CLEAR
```

```
:FG18>CHA(1,100);
```

```
:BMAK
```

```
:FG(17)>D(50);
```

```
:SAVE(CONE)
```

```
:MIX;END
```

```
OBJECT NR: 1. BLOCK TIME: 0.000 0.059 FREE MIXREC.S: 13820
```

```
BLOCKLABEL: BONE .GLOBAL TIME: 0.000 0.059 FREE MIXREC.S: 13820
```

```
:TOP(CONE);LOOK(9)
```

```
'00005' FG18>CHA(1,100);
```

```
'00006' '---MACRO---: BMAK'
```

```
'00007' FG(17,131);
```

```
'00008' '---MACRO---: BS'
```

```
'00009' FG(18,262)
```

```
'00010' FG(17)
```

```
'00011' >D(50);
```

```
05 002: END OF FILE ON ADDITIONAL INPUT FILE.
```

```
:EXIT
```

```
----- EXIT. END OF THIS RUN.-----
```

```
EMSI V1.4
```

```
NAME OF INPUT FILE?
```

```
---- END OF RUN. EXIT TO MONITOR----
```

12. Mera om aritmetik i EMS-3

De aritmetiska operationerna i EMS-3 är:

+ addition

- subtraktion

* multiplikation

/ division

↑ exponentiering

De används alltid så, att en variabel sätts lika med värdet av ett uttryck, där operationerna kan ingå, opererande på tal eller variabler. Det är alltså inte tillåtet att ha operationer inuti termer. Undantaget är OLD+ och OLD-, som i FG(1,OLD+200), men OLD är ingen variabel. Två operationer kan finnas

i högerledet.

Exempel:

$$x = 2 \times 3 \times 4 \text{ ger } x = 24.$$

Eftersom nivån kan sättas på 1/4 decibel i studion, kan variabler också ange fjärdedelar.

Exempel:

$x = 81.20235$ ger x det avrundade värdet 81.25. Avrundning sker i detta och liknande fall alltid till närmaste fjärdedel. Tal som bildas med hjälp av en decimalpunkt kallas flyttal, i motsats till heltal.

$$y = 60.1 \text{ ger } y = 60.0$$

x och y är flyttal och används i EMS-3-termer på alla ställen som anger nivå, men inte på övriga ställen.

Då en variabel tilldelas ett värde blir den av samma typ, heltal eller flyttal, som det första värdet till höger om likhetstecknet.

Exempel:

$$\text{VAR}=10.25+5$$

VAR blir en flyttalsvariabel

$$\text{VAR}=5+10.25$$

VAR blir en heltalsvariabel

$$\text{VAR}=\text{SYM}+17-\text{NR}$$

VAR blir av samma typ som SYM

Observera att då två heltal divideras sker avrundning nedåt, s k heltalsdivision. Resultatet blir ett heltal.

Exempel:

$$x=99999 \text{ ger } x \text{ det största värde som kan tilldelas direkt.}$$

$$x=90000+41071 \text{ ger } x \text{ det största värde som överhuvudtaget kan representeras, 131071.}$$

$$x=32767.75 \text{ ger } x \text{ det största möjliga flyttalsvärdet. Detta tal är också det största tal som kan bildas vid en multiplikation}$$

Man bör komma ihåg begränsningarna i precision och omfång, och organisera så att de bäst tas till vara. Om man t ex vill öka en variabel VAR med en tjugdedel av sitt gamla värde (t ex i ett makro) finns följande tre dåliga alternativ:

$$\text{VAR}=\text{VAR} \times 21/20 \text{ Om VAR är större än 1560 uppstår "overflow on multiplication"}$$

$$\text{VAR}=\text{VAR}/20 \times 21 \text{ Precision förloras. Produkten kan bara anta värden som är multipler av 21.}$$

$$\text{VAR}=\text{VAR} \times 1.05 \text{ 1.05 avrundas till 1.0, VAR ändras inte alls.}$$

Det bästa sättet att utföra det man vill är: $\text{VAR}=\text{VAR}/20+\text{VAR}$.

SAMMANFATTNING AV ELEMENT I EMS-3

Apparattermer

AM(NR, INGÅNG, NIVÅ)
AMP(NR, NIVÅ)
CD(NR, KANAL, NIVÅ)
CHA(NR, NIVÅ)
FF(NR, FILTERKANAL, NIVÅ)
FG(NR, FREKVENNS, NIVÅ, VÅGFORM)
NG(NR, NIVÅ, FÄRGI, FÄRG2)
REV(NR, EKOTID, NIVÅ)
RM(NR, INGÅNG, NIVÅ)

Stud iopunkter

FS
FG3, FG6, FG9, FG12, FG15, FG18, FG21, FG24

Kommandon

APP
CLEAR, CLEAR(MIX), CLEAR(ALL)
DELETE(SYMBOL)
KEEP, KEEP(MIX), KEEP(ALL)
LIST:
MESS(TEXT)
MIX(TID)
NOLIST
STDTIM(MS)
WRITE(SYMBOL)

TEMP-kommandon

CALL(FILNAMN)
ERASE(FILNAMN)
IN(RADANTAL)
LOOK(RADANTAL)
PLAY, PLAY(MIX), PLAY(PARTNAMN), PLAY(PART1, PART2)
REPL(FILNAMN) (fungerar som ERASE följt av SAVE)

SAVE(FILNAMN)
SKIP(RADANTAL)
TOP(FILNAMN)
TRAPP
TRY, TRY(MS), TRY(SEK,MS)

Ett kommando bör stå på en egen rad.

Lokal tid

LT(MS), LT(SEK,MS)

Observera att LT(1,5) betyder detsamma som LT(1,005), ej LT(1,500).

Envelopptermer

ENV(NIVÅ,NIVÅ,MS,KURVFORM,STEG)

Z fungerar som ENV(OLD,Ø,1Ø,Ø,1Ø)

D(MS) fungerar som ENV(OLD,OLD,MS)→Z

Glissandotermer

GLIS(FREKVENS,FREKVENS,MS,KURVFORM,STEG)

Stepptermer

ESTEP(MS)

GSTEP(MS)

EXIT avbryter körningen och startar upp EMS-3 för ett nytt program.

>, #, & står mellan element i kopplingskedjor, envelopkedjor och glissandokedjor. Om kedjan fortsätter över flera rader skall den avdelas så att ett av dessa tecken står sist på raden.

;

bör avsluta varje kedja.

← får användas i FG- och FF-termer.

'KOMMENTAR' bör användas relativt flitigt. Man glömmer annars mycket vad ett program behandlar om.

MAKRONAMN="...TEXT..." makrodefinition. Inget ytterligare får stå på samma rad efter sluttecknet.

MEX får ingå i makrotext (MAKRO-EXIT).

Makron får anropa varandra upp till 100 nivåer. Ett makro får inte innehålla CALL.

+, -, *, /, ↑ aritmetiska operationer för addition, subtraktion, multiplikation, division, exponentiering.

Exempel: C=DxE+F Maximalt tre tal eller variabler får stå i en

IFPOS(VARIABLE)... <

IFZER(VARIABLE)... <

IFNEG(VARIABLE)... <

IFUND(SYMBOL)... <

IFDEF(SYMBOL)... <

EXEMPEL PÅ FELUTSKRIFTER I EMS-3

```
:FRAC=/10
FRAC=
4      73 NONE SPECIFIED ERROR
!FRAC=)2)
FRAC=
4      73 NONE SPECIFIED ERROR
!EX=†3
EX=
4      73 NONE SPECIFIED ERROR
!
:CNT=90000+90000
CNT=90000+90000
04     074      OVERFLOW. ADD OR SUBTRACT
!CNT=30000+30000
!
:PRO=1000*1000
PRO=1000*1000
04     075      OVERFLOW. MULTIPLICATION
!
```

APPENDIX 1.

TILLAATNA STUDIOKOPPLINGAR.

FRAAN	TILL
FG(1)	CHA(1)
FG(2)	CHA(2)
FG(3)	CHA(3)
FG3	CHA(4)
	CD(1)
	FG6
FG(4)	CHA(1)
FG(5)	CHA(2)
FG(6)	CHA(3)
FG6	CHA(4)
	CD(1)
	FG9
	RM(1,A)
	RM(1,B)
	RM(2,B)
	REV(1)
	REV(2)
	AM(1,B)
	FF(1)
	FF(2)
FG(7)	CHA(1)
FG(8)	CHA(2)
FG(9)	CHA(3)
FG9	CHA(4)
	CD(1)
	FG12
FG(10)	CHA(1)
FG(11)	CHA(2)
FG(12)	CHA(3)
FG12	CHA(4)
	CD(1)
	FG15
	RM(1,A)
	RM(1,B)
	RM(2,B)
	REV(1)
	REV(2)
	AM(1,B)
	FF(1)
	FF(2)
FG(13)	CHA(1)
FG(14)	CHA(2)
FG(15)	CHA(3)
FG15	CHA(4)
	CD(1)
	FG18

.FJECT

FRAAN	TILL
FG(16)	CHA(1)
FG(17)	CHA(2)
FG(18)	CHA(3)
FG18	CHA(4)
	CD(1)
	FG21
	RM(1,B)
	RM(2,B)
	REV(1)
	REV(2)
	AM(1,B)
	AM(2,B)
	FF(1)
	FF(2)
FG(19)	CHA(1)
FG21	CHA(2)
	CHA(3)
	CHA(4)
	CD(1)
	FG24
FG(20)	FG21
	FS
FG(21)	FG21
	RM(1,A)
FG24	CHA(1)
	CHA(2)
	CHA(3)
	CHA(4)
	CD(1)
	RM(1,B)
	RM(2,B)
	REV(1)
	REV(2)
	AM(1,B)
	AM(2,B)
	FF(1)
	FF(2)
FG(22)	FG24
	RM(2,A)
FG(23)	FG24
	AM(1,A)
FG(24)	FG24
	AM(2,A)

.EJECT

FRAAN	TILL
NG	CHA(1) CHA(2) CHA(3) CHA(4) CD(1) RM(2,B) REV(1) AM(2,B) FF(1) FF(2)
FF(1)	CHA(2) CHA(3) CHA(4) CD(1) RM(1,A) RM(1,B) RM(2,B) REV(2) AM(1,A) AM(2,B) FF(2) AMP(1) AMP(2)
FF(2)	CHA(2) CHA(3) CHA(4) CD(1) RM(2,B) REV(2) AM(1,B) AM(2,A) FF(1) AMP(1) AMP(2)
REV(1)	CHA(1) CHA(2) CHA(3) CHA(4) CD(1) AM(1,B) AM(2,B) AMP(1) AMP(2)

.EJECT

FRAAN TILL
REV(2) CHA(1)
CHA(2)
CHA(3)
CHA(4)
CD(1)
AM(1,B)
AM(2,B)
AMP(1)
AMP(2)
RM(1) CHA(1)
CHA(2)
CHA(3)
CHA(4)
CD(1)
REV(2)
AM(1,A)
AM(2,B)
FF(1)
FF(2)
AMP(1)
AMP(2)
RM(2) CHA(1)
CHA(2)
CHA(3)
CHA(4)
CD(1)
REV(2)
AM(1,B)
AM(2,A)
FF(1)
FF(2)
AMP(1)
AMP(2)
RM(3) CHA(1)
CHA(2)
CHA(3)
CHA(4)
CD(1)
REV(2)
AM(2,B)
FF(1)
FF(2)
AMP(1)
AMP(2)
AM(1) CHA(1)
CHA(2)
CHA(3)
CHA(4)
CD(1)

.EJECT

FRILL	TILL
AM(2)	CHA(1) CHA(2) CHA(3) CHA(4) CD(1)
AMP(1)	RM(1,B) RM(2,B) REV(2) AM(1,B) AM(2,B) FF(1) FF(2)
AMP(2)	RM(1,B) RM(2,B) REV(2) AM(1,B) AM(2,B) FF(1) FF(2)

.EJECT

APPENDIX 2.

Samtliga möjliga felutskriften i EMS-1, EMS-2 och EMS-3.

02	011	ILLEGAL DELIMITER
02	012	ILLEGAL PARAMETER
02	013	ILLEGAL MNEMONIC
02	014	ILLEGAL NUMBER OF PARAMETERS
02	021	ILLEGAL FREQUENCY GENERATOR NUMBER
02	022	ILLEGAL FREQUENCY GENERATOR FREQUENCY
02	023	ILLEGAL FREQUENCY GENERATOR WAVEFORM
02	024	ILLEGAL FREQUENCY GENERATOR INTENSITY
02	032	ILLEGAL FREQUENCY FILTER CHANNEL
02	033	ILLEGAL FREQUENCY FILTER INTENSITY
02	041	ILLEGAL AMPLIFIER NUMBER
	042	ILLEGAL AMPLIFIER INTENSITY
02	051	ILLEGAL (DIS-) CONNECTION
02	062	ILLEGAL REVERBATION TIME
02	063	ILLEGAL REVERBATION INTENSITY
02	070	TIME MISSING IN D-TERM
	071	ILLEGAL TIME
02	072	TIME OVERFLOW.
02	073	TEMP DISK OVERFLOW. NOTHING MORE WRITTEN ON THE DISK.
02	081	ILLEGAL NOISE COLOUR
02	082	ILLEGAL NOISE INTENSITY
02	091	ILLEGAL AMPLITUDE MODULATOR NUMBER
02	092	ILLEGAL AMPLITUDE MODULATOR ENTRY
02	093	ILLEGAL AMPLITUDE MODULATOR INTENSITY
02	102	ILLEGAL RING MODULATOR ENTRY
02	103	ILLEGAL RING MODULATOR INTENSITY
02	112	ILLEGAL CHANNEL DISTRIBUTOR INTENSITY
02	121	ILLEGAL DEVICE NUMBER
02	150	ILLEGAL ANALOG TAPE INTENSITY
02	142	ILLEGAL CHANNEL INTENSITY
02	201	ILLEGAL ESTEP VALUE
02	241	ILLEGAL ENV SYNTAX
02	242	ILLEGAL ENV AMPLITUDE
02	245	ENV OR GLISTIME LESS THAN STEP.
02	246	ILLEGAL ENV TIME
02	247	ILLEGAL ENV TYPE
02	251	NO DEVICE TO ENVELOPE GIVEN
02	261	ILLEGAL LOCAL TIME VALUE
02	271	ILLEGAL GSTEP VALUE
02	301	NO DEVICE TO GLISSANDO GIVEN
02	302	ILLEGAL GLIS SYNTAX
02	303	ILLEGAL GLIS TIME
02	304	ILLEGAL GLIS TYPE
02	305	ILLEGAL GLIS FREQUENCY
02	142	ILLEGAL CHANNEL INTENSITY
04	001	SYMBOL WITH MORE THAN 6 CHARACTERS.
04	002	INTEGER WITH MORE THAN 5 FIGURES
04	003	TWO DECIMAL POINTS
04	004	A DECIMAL POINT (.) MUST NOT BE USED AS A DELIMITER.
04	010	IN THE EXPRESSION 'A=B' B IS NOT A DEFINED SYMBOL.
04	020	SYMBOL NOT DEFINED.
04	021	SYMBOL DEFINED BUT NOT POSSIBLE IN THIS PART OF THE TEXT.
04	022	DELETION OF MACROS NOT PERMITTED IN MACROS
04	031	TOP NOT EXECUTED IF READ FROM SECONDARY TEXT INPUT
04	032	UNIT 2 NOT FILEORIENTED
04	033	IN,SKIP,DELETE,LOOK NOT EXECUTED IF READ FROM SEC. TEXT INPUT
04	034	FILE NOT PRESENT ON UNIT 2

```

04 035 NO TOP EXECUTED OR SEC, FILE EMPTY
04 036 PARAMETER NOT INTEGER
04 037 ILLEGAL DELIMITER, SHOULD BE ( ; OR CARR, RET.
04 050 LABEL ERROR
04 064 : / * OR ) NOT POSSIBLE AS FIRST ELEMENT OF AN EXPR.
04 074 OVERFLOW, ADD OR SUBTRACT
04 075 OVERFLOW, MULTIPLICATION
04 076 TRY TO DIVIDE BY ZERO
04 100 A '=' IS POSSIBLE ONLY AFTER A SYMBOL OR A '!'.
04 101 ATTEMPT TO ASSIGN A FIXED SYMBOL
OR A SYMBOL STARTING WITH A NONALPHABETIC CHAR.
04 102 SYMBOL TABLE FULL
04 104 A ' ' FOUND IN A NON-MACRO TEXT.
04 103 A MACRO MUST NOT BE CALLED IN ERROR MODE.
04 105 DECIMAL POINT (.) USED ILLEGALLY
04 106 NESTING TOO DEEP, RETURN TO LEVEL 0.
05 001 END OF INPUT FILE WITHOUT AN 'EXIT'
INPUT FROM UNIT 4.
06 001 SECOND LABEL NOT ON THE MT.
06 002 FIRST LABEL NOT ON THE MT.
09 001 NO EXTENSION TO FILENAME
09 002 INPUT FILE FILEORIENTED, FILENAME ?
09 003 OUTPUT FILE FILEORIENTED, FILENAME ?
09 004 FILE NOT FOUND.
09 005 UNIT 1 NOT FILEORIENTED.
09 006 FILE FMNEMO EMS NOT FOUND
09 007 FILE ERROR EMS NOT FOUND.
09 008 NO FILENAME ON FILEORIENTED DEVICE.
99 99 ----- EXIT, END OF THIS RUN.-----
01 001 NOT A PERMANENT SYMBOL
01 002 NOT A COMMAND
01 003 ILLEGAL PARAMETER IN 'PLAY' TERM
01 004 NOTHING TO PLAY
01 005 TIME PARAMETER MUST BE INTEGER.
01 006 TIME PARAMETER NEGATIVE.
01 007 THE ONLY POSSIBLE PARAMETERS TO 'CLEAR' ARE 'MIX' OR 'ALL'
01 009 THE ONLY POSSIBLE PARAMETERS TO 'KEEP' ARE 'MIX' OR 'ALL'
01 010 KEEP WITHOUT MEANING HERE
01 015 SAVE, ERASE, REPL NOT EXECUTED IF READ FROM SEC, TEXT INPUT
01 016 LEFT PARANTHESIS MISSING AFTER SYMBOL
01 017 , AFTER SYMBOL
01 018 'END' AFTER PREVIOUS 'PART' MISSING
01 019 NO SORTRECORDS ON THE 'MIX' DISK
01 020 PART COMMAND FIRST THING IN A BLOCK!
01 021 'PART' AFTER PREVIOUS 'END' MISSING
01 024 LEFT PARANTHESIS MISSING IN 'STDTIM'-TERM
01 025 ERRONEUS TIMEPARAMETER IN 'STDTIM'-TERM
01 030 FILE ALREADY PRESENT
01 031 FILE NOT PRESENT ON UNIT 6.
01 037 ILLEGAL DELIMITER, SHOULD BE ( ; OR CARR, RET.
01 040 ERROR DETECTED IN "MERGE". NOT ENOUGH SPACE ON OUTPUT UNIT
01 041 DURATION OF MIX =0!
01 042 NO RECORDS ON THE TEMP DISK
01 043 CODEGENERATION ERROR, CODE ON THE MT WILL NOT BE SAVED
MIX AND TEMP DISKS ARE SAVED.
01 050 LABEL ERROR
01 080 GLOBAL TIME OVERFLOW, GLOBAL TIME SET TO 0.

```


05 002 END OF FILE ON ADDITIONAL INPUT FILE.
10 101 FREQUENCY LESS THAN 0. PROGRAM SETS FREQ. TO 0.
10 102 FREQUENCY GREATER THAN 15999. PROGRAM SETS FREQ. TO 0.
10 103 INTENSITY LESS THAN 0. PROGRAM SETS INTENSITY TO 0.
10 104 INTENSITY GREATER THAN 120 DB. PROGRAM SETS INTENSITY TO 0.
10 201 ADDRESS ERROR IN SORTRECORD. SORTRECORD IGNORED.
63 001 .DTSER ERROR
15 254 UNCHECKED ERROR IN DECODE PACKAGE.
32 010 BAD ARGUMENT IN IFMT CALL (<0). REPORT TO SYSTEM MANAGER.
78 008 BUFFER OVERFLOW ON .READ
99 002 EMS1 V1.4A 73.02.21

07 004 NAME OF INPUT FILE?
99 005 UNIT 3 NOT FILEORIENTED.
99 010 NOT ENOUGH FREE CORE. MUST BE AT LEAST 2000 (DEC.)
99 006 INTERACTIVE MODE? YES OR NO?
99 007
00 000
64 000 NONE SPECIFIED ERROR.

APPENDIX 3.

TABELL OEVER KVARTSTONER

C		CISS		D	
16,35	16,83	17,32	17,83	18,35	18,89
32,70	33,66	34,65	35,66	36,71	37,78
65,41	67,32	69,30	71,33	73,42	75,57
130,81	134,65	138,59	142,65	146,83	151,13
261,63	269,29	277,18	285,30	293,66	302,27
523,25	538,58	554,37	570,61	587,33	604,54
1046,50	1077,17	1108,73	1141,22	1174,66	1209,08
2093,00	2154,33	2217,46	2282,44	2349,32	2418,16
4186,01	4308,67	4434,92	4564,88	4698,64	4836,32
8372,02	8617,34	8869,85	9129,75	9397,27	9672,64
DISS		E		F	
19,45	20,02	20,60	21,21	21,83	22,47
38,89	40,03	41,20	42,41	43,65	44,93
77,78	80,06	82,41	84,82	87,31	89,87
155,56	160,12	164,81	169,64	174,61	179,73
311,13	320,24	329,63	339,29	349,23	359,46
622,25	640,49	659,26	678,57	698,46	718,92
1244,51	1280,97	1318,51	1357,15	1396,91	1437,85
2489,02	2561,95	2637,02	2714,29	2793,83	2875,69
4978,03	5123,90	5274,04	5428,58	5587,65	5751,38
9956,07	10247,80	10548,08	10857,17	11175,31	11502,77
FISS		G		GISS	
23,12	23,80	24,50	25,22	25,96	26,72
46,25	47,60	49,00	50,44	51,91	53,43
92,50	95,21	98,00	100,87	103,83	106,87
185,00	190,42	196,00	201,74	207,65	213,74
369,99	380,84	392,00	403,48	415,30	427,47
739,99	761,67	783,99	806,96	830,61	854,95
1479,98	1523,34	1567,98	1613,93	1661,22	1709,90
2959,96	3046,69	3135,96	3227,85	3322,44	3419,79
5919,91	6093,38	6271,93	6455,71	6644,88	6839,59
11839,82	12186,76	12543,86	12911,42	13289,75	13679,17
A		AISS		H	
27,50	28,31	29,14	29,99	30,87	31,77
55,00	56,61	58,27	59,98	61,74	63,54
110,00	113,22	116,54	119,96	123,47	127,09
220,00	226,45	233,08	239,91	246,94	254,18
440,00	452,89	466,16	479,82	493,88	508,36
880,00	905,79	932,33	959,65	987,77	1016,71
1760,00	1811,57	1864,66	1919,29	1975,53	2033,42
3520,00	3623,14	3729,31	3838,59	3951,07	4066,84
7040,00	7246,29	7458,62	7677,18	7902,13	8133,68
14080,00	14492,58	14917,24	15354,35	15804,27	16267,37

APPENDIX 4.

The relation between modulation in % of the signal entering the B-entry in an amplitude modulator, and the level in dB of the modulated signal on the A-entry.

<u>Modulation</u>	<u>Level</u>
<u>%</u>	<u>dB</u>
100	100
90	99
80	98
70	97
60	95.5
50	94
40	92
30	89.5
25	88
20	86
15	83
10	80
5	74
1	60

2 times corresponds to 6 dB

5 " " " 14 "

10 " " " 20 "