

PROGRAM FÖR FREKVENSPANALYS PÅ PDP11/03

CHRISTER LARSSON  
HARRIET LINDGREN

Institutionen för Reglerteknik  
Lunds Tekniska Högskola  
Maj 1978

# FREKVENSPANALYS PÅ PDP11/03

Examensarbete utfört av

Christer Larsson  
Harriet Lindgren

Handledare: Ivar Gustavsson

Institutionen för Reglerteknik  
vid Lunds Tekniska Högskola

Lund Maj 1978

## SAMMANFATTNING

Rapporten beskriver vidareutvecklingen av programpaketet INMAT, så att detta förutom mätvärdesinsamling, mätvärdesanalys och enklare regleruppgifter även kan utföra automatisk frekvensanalys med hjälp av korrelationsmetoden.

Programpaket är utvecklat för en PDP11/03 dator och är avsett att användas tillsammans med en förgrunds/bakgrundsmonitor.

## INNEHALLSFÖRTECKNING

INLEDNING	5
1. FREKVENSPANALYS	6
1.1 Allmänt	6
1.2 Korrelationsmetoden	6
1.3 Frekvensanalys på dator	7
1.4 Yttre lågpassfilter	8
1.5 Synkroniserade signaler	8
1.6 Tidsfönster	9
1.7 Trendelimination	9
1.8 Slutlig uppkoppling och utförande av korrelationsmetoden	10
2. PROGRAMUPPBYGGNAD	12
2.1 Användning och ändringar av programpaketet INMAT	12
2.2 Systemets funktion	13
2.3 Nytt kommando	16
3. RUTINER	18
3.1 Rutiner i förgrundsprogrammet	18
3.2 Rutiner i bakgrunden	18
3.3 Övriga rutiner i bakgrunden	21
3.4 DUMP	22
4. KÖRNINGAR	23
4.1 Körning 1	23
4.2 Körning 2	28
4.3 Körning 3	37
4.4 Exempel på trendelimination	39
APPENDIX A Beskrivning av resultatfilen	
APPENDIX B Länkning	
APPENDIX C Programlistor	

## INLEDNING

Syftet med detta examensarbete har varit att utvidga programpaketet INMAT så att detta förutom mätvärdesinsamling och mätvärdesanalys även skall kunna utföra frekvensanalys.

Programpaketet är avsett att användas på datorn PDP 11 tillsammans med en RT-11 förgrunds/bakgrundsmonitor. Systemet kräver ett primärminne på 28 K ord samt ett massminne bestående av två floppy diskar à 120 K ord. Vid utförande av frekvensanalys är samplingstiden begränsad till 80 ms, medan den vid enbart mätvärdesinsamling har sin absolut nedre gräns vid 40 ms.

Vid utförande av frekvensanalys skall datorn ställa ut en sinussignal, vilken får vara insignal till det yttre systemet som skall undersökas. Samtidigt skall datorn mäta systemets utsignal. Ur dessa mätvärden bestämmas sedan med hjälp av korrelationsmetoden systemets förstärkning och argument.

Vid frekvensanalys aktiveras behövliga förgrunds- och bakgrundsnodeer automatiskt. För att detta skall kunna ske har vissa förändringar av stommen i INMAT, se (3), behövt göras. Trots dessa förändringar kan systemet användas för sitt ursprungliga syfte. Parallellt med frekvensanalys kan även andra förgrundsnodeer köras, men man bör dock i detta sammanhang vara uppmärksam på att samplingstiden eventuellt måste förlängas.

I kapitel 1 beskrivs frekvensanalys med hjälp av korrelationsmetoden. Kapitel 2 behandlar programuppbyggnaden och i kapitel 3 beskrivs de införda rutinerna närmare. Slutligen ges i kapitel 4 exempel på olika körningar.

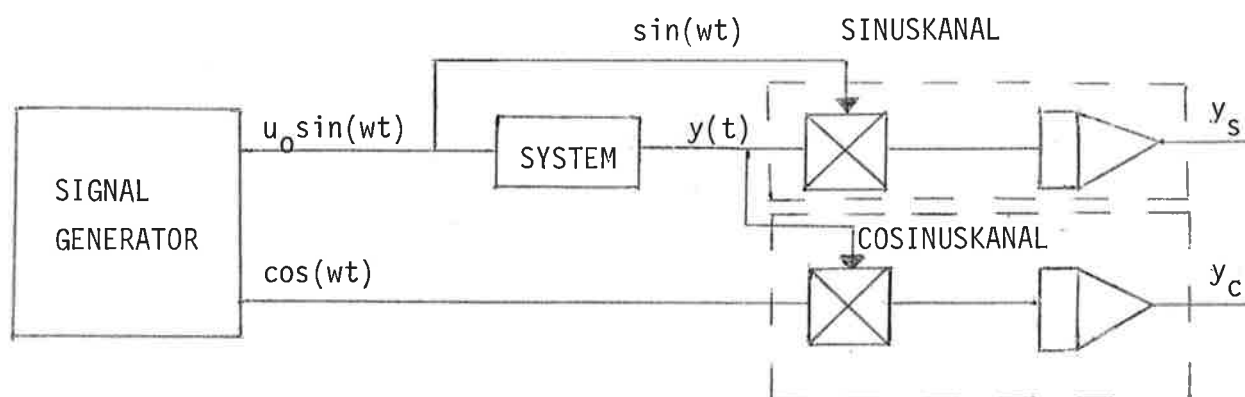
## 1. FREKVENSPANALYS.

### 1.1 Allmänt

Frekvensanalys baseras på det faktum att om man till ett linjärt stabilt system med överföringsfunktionen  $G(s)$  har insignalen  $u(t) = u_0 \sin(\omega t)$ , kan utsignalen efter insvängningsförloppet beskrivas som  $y(t) = u_0 |G(i\omega)| \sin(\omega t + \arg G(i\omega))$ . Värdet av överföringsfunktionens belopp  $|G(i\omega)|$  fås genom att bilda kvoten mellan utsignalens och insignalens amplituder och argumentet genom att bestämma utsignalens fasläge i förhållande till insignalens. Genom att upprepa denna procedur för olika värden på frekvensen  $\omega$  kan överföringsfunktionens värden bestämmas för varje önskat frekvensintervall.

### 1.2 Korrelationsmetoden

Den metod för frekvensanalys som använts i detta examensarbete är korrelationsmetoden. Fördelarna med denna metod är främst god noggrannhet och en effektiv elimination av störningar. I princip går metoden ut på att filtrera ut den sinuskomponent i utsignalen som har samma frekvens som insignalen, se figur 1.



Figur 1. Konfiguration för frekvensanalys.

Insignalen till systemet är  $u(t) = u_0 \sin(\omega t)$  och utsignalen  $y(t) = y_0 \sin(\omega t + \phi)$ , där  $y_0 = |G(i\omega)| u_0$ ,  $\phi = \arg G(i\omega)$  och  $G$  är systemets överföringsfunktion.

Sinus- och cosinuskanalerna kan beskrivas av uttrycken:

$$Y_S(T) = \int_0^T y(t) \sin(\omega t) dt = \int_0^T y_0 \sin(\omega t) \sin(\omega t + \phi) dt = \frac{T y_0}{2} \cos \phi - \frac{y_0}{2} \int_0^T \cos(2\omega t + \phi) dt$$

$$Y_C(T) = \int_0^T y(t) \cos(\omega t) dt = \int_0^T y_0 \cos(\omega t) \sin(\omega t + \phi) dt = \frac{T y_0}{2} \sin \phi + \frac{y_0}{2} \int_0^T \sin(2\omega t + \phi) dt$$

Om man nu väljer integrationstiden så att  $\omega t$  är en multipel av  $\pi$  fås:

$$Y_S(T) = 1/2 \cdot y_0 T \cos \phi = 1/2 u_0 \operatorname{Re}(G(i\omega)) T$$

$$Y_C(T) = 1/2 \cdot y_0 T \sin \phi = 1/2 u_0 \operatorname{Im}(G(i\omega)) T$$

Väljes alltså integrationstiden som en multipel av  $\pi/\omega$  blir resultatet av integrationerna att  $Y_S$  och  $Y_C$  blir proportionella mot  $G$ 's real och imaginärdelar. Ett konstant fel i mätningen av utsignalen kommer inte att påverka värdena av  $y_S$  och  $y_C$  för argument som är multiplar av  $2\pi/\omega$ . (1)

### 1.3 Frekvensanalys med dator.

I det nuvarande utförandet av examensarbetet genererar datorn en sinus-signal och ställer ut denna via sitt interface. Eftersom utställningen av sinussignalen endast sker vid varje samplingstillfälle kommer den utställda signalen att bli en approximation av en sinussignal. Utseendet av denna styckvis konstanta signals frekvensspektrum kommer att bero på förhållandet  $T/\Delta T$ , där  $T$  är periodtiden och  $\Delta T$  är samplingsintervallets längd. Om antalet samplingar per period är för litet så kommer sidofrekvenserna att få för stor inverkan. Som minsta värde på detta antal har vi valt 8. Se (2).

#### 1.4 Yttre lågpasfilter

I den tidigare beskrivningen har korrelationsmetoden betraktats som ett analogt filter med en kontinuerlig insignal. Den verkliga situationen är istället att vi har ett digitalt filter med en styckvis konstant insignal. Vid integration över ett jämnt antal grundvågsperioder kommer i det kontinuerliga fallet övertoner vars frekvens är jämna multiplar av grundvågsfrekvensen, ej att inverka på korrelationsmetoden. I det digitala fallet ger emellertid övertonerna ett oönskat bidrag till slutresultatet. Detta bidrag minskar då antalet samplingsintervall per period ökar och då övertonernas amplitud minskar. För att få bättre noggrannhet kan man låta den utställda signalen passera ett lågpasfilter, vilket dämpar övertonerna kraftigt. Dimensioneringen av detta filter blir emellertid besvärlig eftersom filtrets bandbredd bör vara frekvensberoende. Därför har denna metod ej använts här. Se (2).

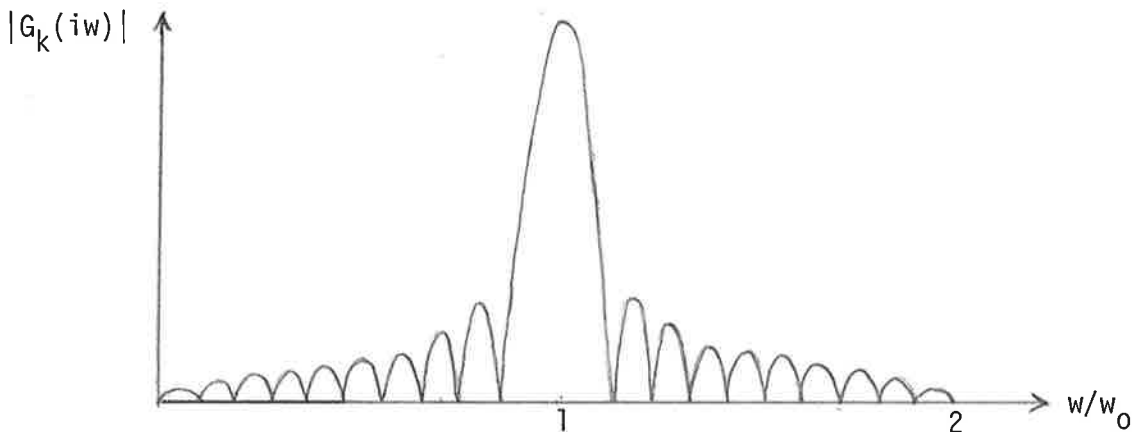
#### 1.5 Synkroniserade signaler

I korrelationsmetoden ingår uttrycken  $\sin(\omega t)$  och  $\cos(\omega t)$ , dessa skall vara synkroniserade med den yttre uppmätta signalen  $y(t)$ . I de använda programmen sker emellertid inläsning av de yttre signalerna innan uträkning och utställning av sinussignalen sker. Man får därmed en viss tidsskillnad mellan den inlästa och utställda signalen. Vi har därför valt att läsa in denna sinussignal igen och använda dessa värden i beräkningarna. Ett annat sätt vore t.ex att ha sinusvärdena i en tabell, så att dessa kunde ställas ut samtidigt med inläsningen. Cosinusvärdena fås på samma sätt som sinussvärdena, dvs en cosinussignal genereras synkront med sinussignalen, ställs ut, och läses omedelbart in igen för att användas i beräkningarna. Alternativt kunde man derivera sinussignalen för att få fram cosinusvärdena.



## 1.6 Tidsfönster

Som tidigare nämnts kan de två korrelationskanalerna (sinus resp. cosinuskanalen) betraktas som filter. Dessa filter är av bandpass-karaktär, med en amplitudkurva av utseende enligt figur 2.



Figur 2. Amplitudkurva för en korrelationskanal.

Detta medför att dessa filter släpper igenom även övertonerna hos sinus- och cosinussignalerna. Sidbanden kan emellertid reduceras genom att de båda signalerna multipliceras med funktionen  $1 - \cos(2\pi t)/T$  innan de multipliceras med det undersökta systemets utsignal. (1)

## 1.7 Trendelimination

Som tidigare nämnts kommer en konstant nivå på systemets utgång ej att påverka korrelationskanalerna om mättiden väljs som en multipel av insignalens period. Däremot kommer en linjär trend att resultera i ett fel på sinuskanalen, alltså även med ovanstående förutsättningar. Se (1). Den metod som använts här för trendelimination är "Perssons metod", då speciellt för fallet med en driftsstörning i form av en linjär trend. Utsignalen från sinuskanalen kan skrivas:

$$Y_S = \int_0^T y(t) \sin(\omega t) dt = \int_0^T y(\tau/w) \sin(\tau) d\tau = \frac{1}{w} \int_0^{wT} y(\tau/w) \sin \tau d\tau = \frac{1}{w} \sum_{k=0}^N \int_{(k-1)\pi}^{k\pi} y(\tau/w) \sin \tau d\tau$$

Där alltså N är antalet halvperioder som integrationen skall ske över.

Idén med metoden är att istället skriva ovanstående uttryck på följande sätt:

$$Y_S = 1/w \sum_{k=1}^N \alpha_k \int_0^{k\pi} y(\tau/w) \sin \tau d\tau$$

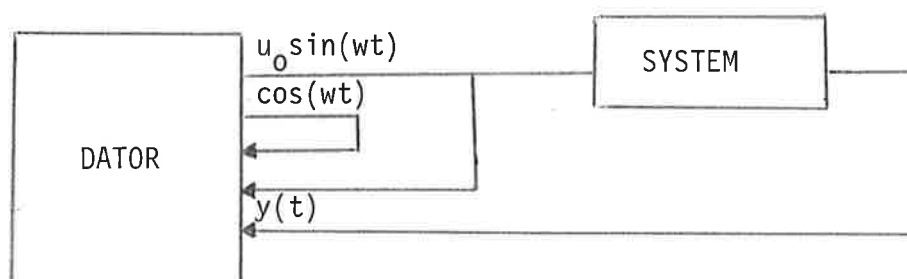
Man kan då visa att parametern  $\alpha_k$  kan väljas så att ett fel i systemets utgång i form av ett polynom helt kan elimineras. Detta är då möjligt om  $N=L+2$ , där  $L$  är polynomets ordningstal, och om  $\alpha_k$  väljs proportionellt mot binomialkoefficienten:

$$\binom{L+1}{k-1} = \frac{(L+1)!}{(k-1)! (L-k+2)!}$$

I fallet med en linjär trend fås med  $L=1$  och  $N=3$ :  $\alpha_1 \sim 1$ ,  $\alpha_2 \sim 2$ ,  $\alpha_3 \sim 1$ .

Normaliserat fås 0.25, 0.50 och 0.25. Men om  $N=3$  innebär ju detta att integrationen skall ske över en och en halv period, därför har vi istället valt att integrera över tre perioder. Då får man med  $N=6$  och normaliserat:  $\alpha_k = 0.75, 1.50, 0.75, 0.75, 1.50, 0.75$ , med  $k=1..6$ . Man multiplicerar alltså mätvärdena med 0.75 under första halvperioden, med 1.50 under andra, osv.

### 1.8 Slutlig uppkoppling och utförande av korrelationsmetoden



Figur 3: Uppkoppling för korrelationsmetoden

Datorn ställer ut de styckvis konstanta cosinus- och sinussignalerna. Dessa matas åter direkt in i datorn, sinussignalen får dessutom utgöra insignal till det betraktade systemet, varefter systemets utsignal matas in i datorn. Signalen efter systemet multipliceras sedan med cosinus respektive sinussignalen. De erhållna värdena summeras upp i var sin summa.

$$S_c = \sum_i \cos(\omega t_i) y(t_i) \Delta T$$

$$S_s = \sum_i \sin(\omega t_i) y(t_i) \Delta T / u_0$$

Där  $y$  är utsignalen efter systemet,  $\cos(\omega t_i)$  den utställda cosinus-signalen,  $\sin(\omega t_i)$  insignalen till systemet,  $u_0$  dess amplitud och  $\Delta T$  samplingstiden. Systemets överföringsfunktion fås då som:

$$|G_s(i\omega)| = \sqrt{4(S_s^2 + S_c^2)} / (T_I \cdot u_0) \quad (T_I = \text{mättiden})$$

$$\text{Arg } G(i\omega) = \arctan(S_c / S_s)$$

Den högsta tillåtna frekvensen fås ur  $f = 1/(\Delta T \cdot 8)$ . Med  $\Delta T = 0.08$  sekunder fås  $f = 1.56$  Hz.

## 2. PROGRAMUPPBYGGNAD

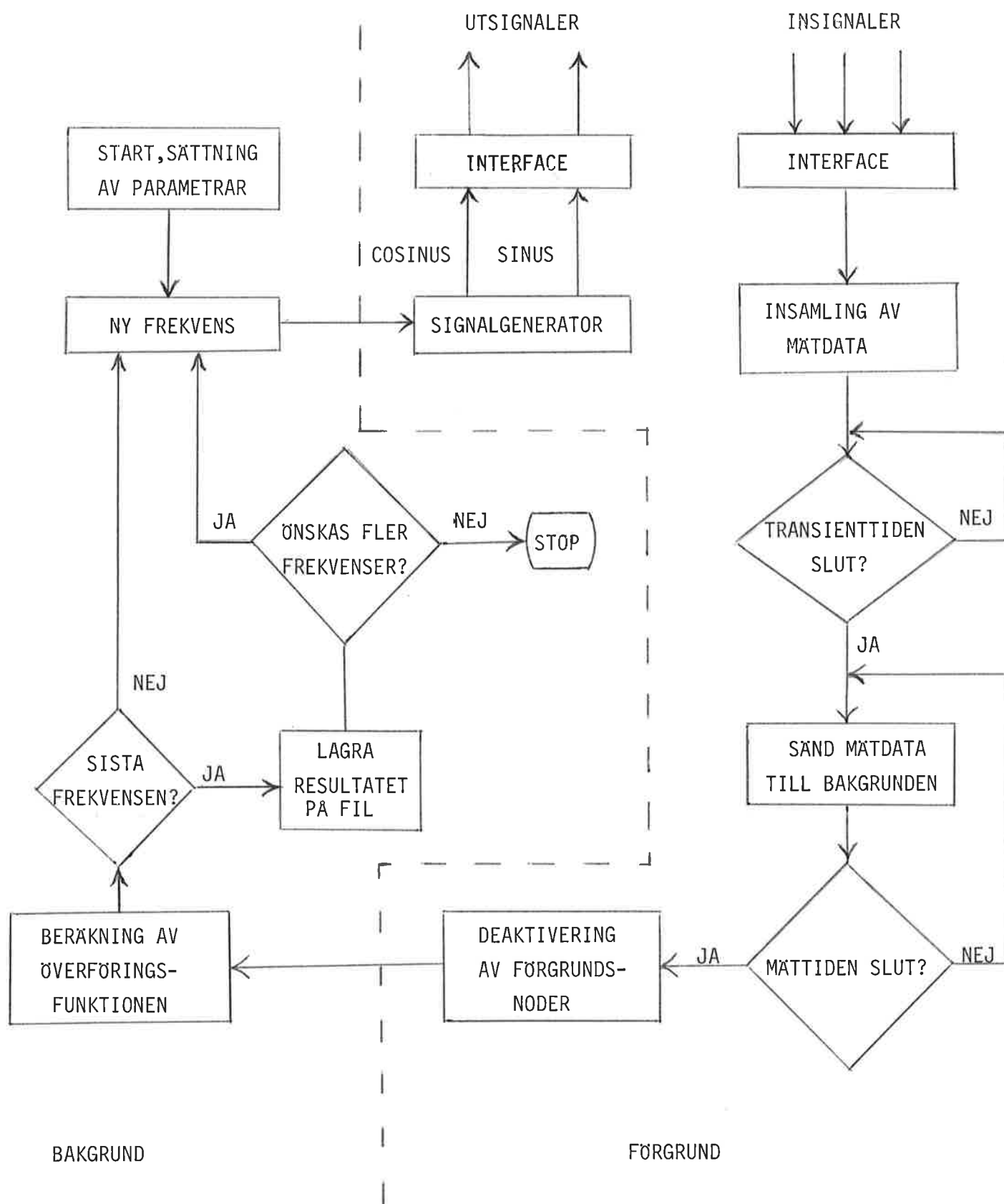
### 2.1 Användning och ändringar av programpaketet INMAT.

Frekvensanalys bygger på programpaketet INMAT. INMAT ombesörjer insamlingen av mätdata och kommunikationen mellan bakgrunden och förgrunden. INMAT är både kommando och fråga-svars styrt, emedan parametrarna vid frekvensanalys endast sätts en gång varefter aktivering och deaktivering av noder sker automatiskt. Vi har därför fått göra vissa ändringar och tillägg i redan befintliga program. Dessutom finns av programmen SPAR och SNOD två versioner, där den ena versionen (RPAR och RNOD) används vid frekvensanalys. Med hjälp av flaggor bestäms vilken version som skall användas.

Vid frekvensanalys ligger de tidskritiska programmen som förgrundsnoder, medan de icke tidskritiska ligger dels som bakgrundsnoder och dels som styrande program vid sidan om nodinterpretationen i bakgrunden.

## 2.2 Systemets funktion

### 2.2.1 Blockschema



### 2.2.2 Parametrar

För att frekvensanalys skall kunna utföras krävs att operatören sätter vissa parametrar och gör vissa val om hur systemet skall köras: (se exemplen i kapitel 4)

Huvud-samplingstid	-Anger hur ofta nodinterpreteringen skall ske.
Antal insignaler	-Är vid frekvensanalys tre stycken.
Antal utsignaler	-Är vid frekvensanalys två stycken.
Frekvensgränser	-Det finns vissa begränsningar vid sättningen av ändfrekvenserna: <ol style="list-style-type: none"> <li>1. Den högsta frekvensen bestäms av det faktum att vi har valt det lägsta antalet samplingar per period till 8. Ändfrekvensen är naturligtvis då också beroende av samplingstiden. Med ovanstående samplingar per period och samplingstiden 0.08 sekunder blir denna frekvens lika med <math>1/(0.08 \cdot 8) = 1.56</math> Hz.</li> <li>2. Den lägre frekvensen får ej väljas för låg i förhållande till samplingintervallets längd och antalet mätperioder. Detta beroende på att antalet samplingsperioder man mäter över räknas. Man riskerar alltså då "integer overflow" i denna räknare. Frekvensen i fråga fås ur uttrycket: <math>\frac{\text{antal mätperioder}}{\text{samplingstiden} \cdot 32767}</math>, t.ex. <math>f = \frac{3}{0.08 \cdot 32767} = 1.1 \cdot 10^{-3}</math> Hz.</li> <li>3. Eftersom mätresultatet lagras i en vektor är det totala antalet mätpunkter maximerat till 50. Kombinationen av frekvensintervallets längd och antalet mätpunkter måste alltså väljas så att antalet mätpunkter ej överstiger 50.</li> </ol>
Frekvensenhet	-Herz eller radianer/sekund.
Antal mätpunkter per dekad	-Möjliga värden är 16, 8, 4, 2 och 1.

Transienttid	-Ges i enheten sekunder.
Mättid	-Ges i antal mätperioder. Om trendelimination begärts måste detta antal vara en mutipel av tre.
Amplitud	-Utsignalens amplitud, ges som ett tal mellan 0 och 1 vilket motsvarar 0 till 10 volt.
Tidsfönster eller ej	-
Manuell-automatisk körning	-Se kapitel 2.2.4.
Trendelimination eller ej	-Egentligen linjär trendelimination.
Logisk enhet	-Möjlighet att välja enhet för resultatutskriften.

### 2.2.3 Körning

Dessa parametrar sätts vid uppstarten av systemet varvid också bakgrundsnoderna aktiveras. Med hjälp av frekvensgränserna, antal mätpunkter per dekad och huvudsamplingstiden bestäms ett antal frekvenser vilka ligger ungefär ekvidistant på en logaritmisk skala. Därefter levereras aktuella parametrar till förgrundsnoderna och dessa aktiveras.

Genom kommandot STAR startas nodinterpreteringen och signalerna ställs ut via datorns interface. Samtidigt börjar insamlingen av mätdata. Överföringen av mätdata till bakgrunden startar dock inte förrän efter transienttidens slut. Överföringen pågår sedan under hela mättiden, varefter förgrundsnoderna desaktiveras. Under den dödperiod som härvid uppstår slutför bakgrundsnoderna sina beräkningar och därefter beräknas en ny frekvens och proceduren upprepas.

Andra bakgrundsnoder än de i frekvensanalys förekommande får användas, men man bör observera att bakgrundsnoderna endast har tillgång till mätvärden under integrationstiden i korrelationsmetoden.

Om man under frekvensanalysen använder operatörskommunikationen (3) bör man tänka på att exekveringstiden för förgrundsprogrammet ökar och därmed risken för förlängda samplingsintervall. Bildandet av datafil med kommandot CRNF är under frekvensanalys förbjudet, då denna extra fil skulle störa lagringen och läsningen av mätvärdesfilen.

Efter sista frekvensen har operatören möjlighet att lagra sina resultat i en fil på disketten. Han (hon) har också möjlighet att begära en ny körning med tätare icke-ekvidistanta frekvenser. Se vidare exemplen på körning av systemet (kapitel 4).

#### 2.2.4 Manuell körning

Vid automatisk körning levereras en ny frekvens genast efter det att föregående frekvens färdigbehandlats. Vid analys av system med egensvängningar är detta till stor nackdel. Därför har vi infört möjligheten att köra manuellt. Vid manuell körning bestämmer operatören själv när det är tid för en ny frekvens. Operatören har då även möjlighet att ändra sinussignalens amplitud mellan frekvenserna. Detta är speciellt användbart vid analys av system med resonanstoppar.

#### 2.3 Nytt kommando

Förutom de i INMAT redan befintliga kommandona har ett nytt sådant införts.

CHSP -Kontrollerar om nodinterpreteringen i förgrunden skett vid rätt tillfälle. Varje gång samplingstiden varit längre än den önskade, räknas en räknare i förgrunden upp. Kommandot CHSP anger värdet på denna räknare, varefter denna nollställs.

Man skall observera att utförandet av kommandot STAR oftast medför att ett samplingsintervall blir för långt. Även vid sättning av förgrunds-noder blir samplingstiden ibland för lång. Man bör alltså ge kommandot CHSP omedelbart före eller efter kommandot STAR så att räknaren nollställs.



Vid frekvensanalys sker sättning av förgrundsnoder samt start av nodinterpretationen (STAR) automatiskt före varje ny frekvens. För att få en uppfattning om hur många gånger systemet varit för långsamt, bör man göra CHSP efter uppstarten, dvs efter det att frågorna besvarats. Då man sedan efter avslutad frekvensanalys gör CHSP, skall man tänka på att samplingsintervallet kan ha varit för långt minst en gång per frekvens utan att detta påverkat resultatet eftersom förgrundsnoderna då ej varit aktiva.

### 3. RUTINER

#### 3.1 Rutiner i förgrundsprogrammet

I förgrundsprogrammet ligger tidskritiska reelltidsrutiner som t.ex rutiner för signalgenerering. Vid frekvensanalys utnyttjas två sådana rutiner, nämligen en för generering av cosinus- och sinussignaler och en rutin som kontrollerar tiden. Det är den sistnämnda som är ny i sammanhanget.

##### 3.1.1 TIME

Rutinen TIME kontrollerar tiden genom att räkna samplingar, dvs de tillfällena som nodinterpretering sker. Rutinen startar och stoppar överföringen av data till bakgrunden, dessutom deaktiverar den alla förgrunds-noder efter mättidens slut. TIME aktiveras automatiskt vid igångkörningen av frekvensanalys.

Noden ifråga kan sättas ledig genom anropet: STOP:TIME3

Om man vill sätta alla förgrunds-noder som används vid frekvensanalys lediga, göres detta genom kommandot: STOP:SINU1,SINU2,TIME3

#### 3.2 Rutiner i bakgrunden

I bakgrunden ligger beräkningsrutinerna och deras respektive startupp-rutiner. De sistnämndas primära uppgift är att allokeras minnesutrymme i parameterareorna samt att sätta vissa parametrar. Vid frekvensanalys används tre bakgrunds-noder och fyra noder som ligger vid sidan av nod-interpreteringen.

##### 3.2.1 FREQST

FREQST är startupp-rutin till rutinen FREQ. FREQST frågar bl.a efter frekvensgränserna och antalet frekvenser per dekad. Med hjälp av dessa data och en datasats bestående av ett antal standardfrekvenser, bestäms den frekvensföljd som skall användas i frekvensanalysen.

Antalet tillåtna frekvenser per dekad är 16, 8, 4, 2 och 1.

Vidare frågar rutinen efter enheten på frekvensen, antalet mätperioder och transienttiden.

Om operatören svarar 0 på frågan om antalet mätperioder sätts detta antal beroende av frekvensen enligt följande schema:

1 om	frekvensen $f < 0.1$ Hz
3 om	$0.1 \leq f < 0.2$ Hz
5 om	$0.2 \leq f < 0.5$ Hz
10 om	$0.5 \leq f < 1.0$ Hz
20 om	$f \geq 1.0$ Hz

Transienttiden är begränsad uppåt eftersom den i programvaran är uttryckt i antal samplingar. Om för lång transienttid väljs, ger datorn besked om detta och man får välja om transienttiden. Transienttiden 0 sekunder är tillåten.

### 3.2.2 FREQ

FREQ har till uppgift att med hjälp av data från FREQST beräkna en aktuell frekvens och leverera denna till förgrundsnoderna. FREQ räknar också om transienttid och mättid till antal samplingar och skickar dessa data till förgrunden. Samtidigt aktiveras också förgrundsnoderna.

### 3.2.3 CRLMST

CRLMST är startupprutin till CRLM. CRLMST ger operatören möjlighet att välja huruvida tidsfönster och trendelimination skall användas eller ej. Vidare frågar rutinen om automatisk eller manuell körning av frekvensanalysen skall användas samt på vilken logisk enhet man vill ha resultatutskriften.

#### 3.2.4 CRLM

I CRLM ingår beräkningarna av den sökta överföringsfunktionen enligt korrelationsmetoden. För att kunna utföra detta har rutinen tre insignaler, dvs en cosinus- och en sinussignal samt utsignalen från det yttre analyserade systemet.

Vid varje frekvensskifte raderas den datafil på disketten ut, vilken används vid överföring av data från förgrunden till bakgrundsprogrammen. En ny datafil för detta ändamål bildas därefter omedelbart. I rutinen ingår också möjligheterna att bilda en datafil för slutresultatet samt igångsättning av en extra analys med tätare frekvenser.

#### 3.2.5 FRCLST

Denna rutin används i det fall då operatören begärt en ny körning med tätare frekvenser och är för övrigt startupprutin till FRCL. FRCLST begär uppgifter om frekvensgränser och önskat antal mätpunkter. Detta antal är på grund av resultatvektorns storlek begränsat till 50. Rutinens huvuduppgift är att beräkna frekvenserna till den nya analysen. Frekvenserna bestäms med utgångspunkt från att motsvarande period skall innehålla ett jämnt antal samplingsintervall. Detta antal bestäms alltså för ändfrekvenserna, varefter möjliga frekvenser sedan kan beräknas med utgångspunkt från alla heltal mellan dessa gränser. En aktuell frekvens bestäms sedan ur t.ex. varje, vartannat eller vart tredje heltal i intervallet beroende på hur många frekvenser som har önskats. Detta kan medföra att vissa avvikelser mellan önskat antal frekvenser och det antal man verkligen får uppkommer. Om operatören t.ex. har begärt 10 frekvenser och det maximala antalet är 12, så får han i verkligheten 12 stycken.

#### 3.2.6 FRCL

FRCL används vid en extra körning med tätare, logaritmiskt icke-ekvidistanta frekvenser. Rutinens uppgift är den samma som för rutinen FREQ, dvs att

beräkna aktuella parametrar, sända dessa till förgrundsnoderna och samtidigt aktivera dessa noder.

De ovan uppräknade bakgrundsnoderna kan sättas lediga genom kommandot:

```
>STOP:FREQ1,CRLM1,FRCL1
```

### 3.3 Övriga rutiner i bakgrunden

Denna rubrik innefattar rutiner som inte ingår i nodinterpreteringen.

#### 3.3.1 FREQAN

Denna rutin är en slags överordnad startupprutin för frekvensanalysen. Den initierar aktiveringen av CRLM och FREQ samt ombesörjer att parametrarna för den första mätpunkten blir satta.

#### 3.3.2 WAIT

WAIT används vid manuell körning och bestämmer när en avbruten analys skall fortsätta. Rutinen ombesörjer också ändringen av utsignalens amplitud vid begäran från operatören.

#### 3.3.2 RESO och RESU

Dessa rutiner används vid utskrift av för experimentet karaktäristiska data. RESO används dock endast tillsammans med DUMP.

#### 3.3.3 RFIL

RFIL anropas av CRLM och har till uppgift att skapa en resultatfil på massminnet. Filens namn bestäms av operatören. RFIL ombesörjer även att resultatet skrivs in på filen efter avslutad körning av ett frekvensintervall.

### 3.4 DUMP

Rutinen DUMP ligger helt utanför det ordinarie systemet. DUMP används för att läsa en på disketten lagrad resultatfil. (Se Appendix A och exempel 3 i kapitel 4.)

## 4. Körningar

I detta kapitel visas tre exempel på körningar, nämligen två analyser och en läsning av en resultatfil på diskette. Slutligen visas också inverkan av trendelimination.

Understruken text är operatörens kommandon och svar på ställda frågor. Tecknet ↵ betyder nedtryckning av tangenten "RETURN".

### 4.1 Körning 1

I denna körning visas en analys av systemet  $1/(s+1)$ .

```
.SET USR NO SWAP↵ ; Förhindrar att systemrutinen USR
                    ; swappas.
.FRUN DX1:FG/N!500↵ ; Start av förgrundsprogrammet.
.RUN DX1:EG↵       ; Start av bakgrundsprogrammet.
```

DO YOU WANT FREQUENCY ANALYSIS?(Y=YES,N=NO)Y↵

```
*****
* WELCOME TO FREQUENCY ANALYSIS *
*****
```

#### SHORT DESCRIPTION:

THE COMPUTER HAS TWO OUTPUT SIGNALS AND NEEDS THREE INPUT SIGNALS.  
ON OUTPUT 1(A) YOU HAVE A SINUS SIGNAL, WHICH IS THE INPUT SIGNAL TO THE SYSTEM YOU WANT TO ANALYSE.  
THIS SIGNAL SHALL ALSO BE DIRECTLY COUPLED TO THE SECOND INPUT ON THE INTERFACE.  
ON OUTPUT 2(B) YOU HAVE A COSINUS SIGNAL. THIS SIGNAL SHALL BE DIRECTLY COUPLED TO THE THIRD INPUT.  
FINALLY YOU PUT THE OUTPUT SIGNAL FROM THE SYSTEM ON INPUT 1.

THE FOLLOWING QUESTION SHALL BE ANSWERED IN THE UNIT:  
"NUMBER OF TICKS". 1 TICK=0.020 SEC.

```
MAIN SAMPLE PERIOD:4↵ ; Sätter samplingsperioden.
NUMBER OF INPUT SIGNALS:3↵ ; De tre första signalerna läses från
                             ; A/D-omvandlaren.
NUMBER OF OUTPUT SIGNALS:2↵ ; De två första signalerna i utsignalearen
                             ; ställs ut till D/A-omvandlaren.
PLEASE DEFINE ON WHICH INPUTS YOU HAVE THE DIFFERENT
SIGNALS. THE FIRST SIGNAL IS THE OUTPUT SIGNAL FROM THE
ANALYSED SYSTEM. THE SECOND SIGNAL IS THE INPUT SINUS SIGNAL
TO THIS SYSTEM. THE THIRD SIGNAL IS THE COSINUS SIGNAL.
THE ANSWER SHALL IN GENERAL BE: INPU1, INPU2, INPU3
INPUT SIGNAL: INPU1, INPU2, INPU3↵
```

IF YOU ANSWER YES TO THE FOLLOWING QUESTION  
YOU HAVE TO CHOOSE AN INTEGRATION TIME OF 3 PERIODS.  
DO YOU WANT TREND ELIMINATION?N↵

DO YOU WANT TIME WINDOW?N↓

YOU HAVE NOW THE POSSIBILITY TO DECIDE IF THE SYSTEM IS TO BE RUN AUTOMATICALLY OR NOT. IF NOT, THE RUNNING OF THE ANALYSIS IS INTERRUPTED BETWEEN TWO FREQUENCIES, AND YOU CAN DECIDE YOURSELF WHEN IT IS TIME FOR A NEW FREQUENCY. YOU CAN ALSO AT THAT OCCASION CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL OF THE COMPUTER.

DO YOU WANT AUTOMATIC FREQUENCY ANALYSIS?Y↓

ON WHICH DEVICE DO YOU WANT THE FINAL RESULT?  
(ANSWER TT OR LP ON THE FOLLOWING QUESTION!)  
OUTPUT ON DEVICE:IT↓

PLEASE DEFINE THE FREQUENCY LIMITS. IF YOU WANT TO WORK IN THE UNIT HZ ANSWER H AND IF YOU WANT RAD/S ANSWER R.

WHICH UNIT DO YOU WANT?H↓

LOWER FREQUENCY LIMIT:.1↓

HIGHER FREQUENCY LIMIT:1↓

HOW MANY FREQUENCIES PER DECADE DO YOU WANT  
(POSSIBLE VALUES ARE 1, 2, 4, 8 AND 16)?16↓

PLEASE CHOOSE HOW MANY PERIODS OF THE FREQUENCY THAT THE INTEGRATION IN THE CORRELATION METHOD WILL LAST.

IF YOU ANSWER 0 THE NUMBER WILL BE SET DEPENDING ON THE FREQUENCY AS FOLLOWS:

- 1 IF  $F < 0.1 \text{ HZ} (0.6284 \text{ RAD/SEC})$
- 3 IF  $0.1 < F < 0.2 \text{ HZ} (1.2568 \text{ RAD/SEC})$
- 5 IF  $0.2 < F < 0.5 \text{ HZ} (3.1420 \text{ RAD/SEC})$
- 10 IF  $0.5 < F < 1 \text{ HZ} (6.2840 \text{ RAD/SEC})$
- 20 IF  $F > 1 \text{ HZ}$

INTEGRATION TIME (NUMBER OF PERIODS):3↓

TRANSIENT TIME (SEC):10↓

AMPLITUDE OF THE OUTPUT SIGNAL:.8↓

THE FREQUENCIES ARE GIVEN IN THE UNIT HZ.

LOWER FREQUENCY LIMIT= 0.10E+00

HIGHER FREQUENCY LIMIT= 0.10E+01

NUMBER OF FREQUENCIES PER DECADE= 16

INTEGRATION TIME (PERIODS)= 3

TRANSIENT TIME (SEC)= 10.0

NO TREND ELIMINATION WILL BE DONE.

NO TIME WINDOW WILL BE USED.

THE SYSTEM IS TO BE RUN AUTOMATICALLY.

AMPLITUDE OF THE OUTPUT SIGNAL: 0.80



DO YOU WANT TO CHANGE ANY PARAMETERS? N ↓

; Man har här möjlighet att ändra parametervärden. Om så önskas måste man dock skriva in alla värdena igen.

↓  
>STAR ↓

```

#####
FREQUENCY= 0.10E+00 ARGUMENT= -31.3 GAIN= 0.84E+00
FREQUENCY= 0.12E+00 ARGUMENT= -36.0 GAIN= 0.79E+00
FREQUENCY= 0.15E+00 ARGUMENT= -42.3 GAIN= 0.72E+00
FREQUENCY= 0.17E+00 ARGUMENT= -45.4 GAIN= 0.68E+00
FREQUENCY= 0.20E+00 ARGUMENT= -49.6 GAIN= 0.62E+00
FREQUENCY= 0.22E+00 ARGUMENT= -52.4 GAIN= 0.58E+00
FREQUENCY= 0.25E+00 ARGUMENT= -55.5 GAIN= 0.53E+00
FREQUENCY= 0.30E+00 ARGUMENT= -59.6 GAIN= 0.46E+00
FREQUENCY= 0.35E+00 ARGUMENT= -62.7 GAIN= 0.41E+00
FREQUENCY= 0.40E+00 ARGUMENT= -65.5 GAIN= 0.36E+00
FREQUENCY= 0.45E+00 ARGUMENT= -67.0 GAIN= 0.33E+00
FREQUENCY= 0.50E+00 ARGUMENT= -68.5 GAIN= 0.29E+00
FREQUENCY= 0.60E+00 ARGUMENT= -70.5 GAIN= 0.25E+00
FREQUENCY= 0.69E+00 ARGUMENT= -72.0 GAIN= 0.22E+00
FREQUENCY= 0.78E+00 ARGUMENT= -73.0 GAIN= 0.19E+00
FREQUENCY= 0.89E+00 ARGUMENT= -73.5 GAIN= 0.17E+00

LAST FREQUENCY
FREQUENCY= 0.96E+00 ARGUMENT= -74.3 GAIN= 0.16E+00

```

DO YOU WANT TO SAVE THE RESULT ON A FILE? N ↓

```

MAIN SAMPLE PERIOD=0.080SEC
THE FREQUENCIES ARE GIVEN IN THE UNIT HZ.
LOWER FREQUENCY LIMIT= 0.10E+00
HIGHER FREQUENCY LIMIT= 0.10E+01
NUMBER OF FREQUENCIES PER DECADE= 16
INTEGRATION TIME (PERIODS)= 3
TRANSIENT TIME (SEC)= 10.0
NO TREND ELIMINATION WILL BE DONE.
NO TIME WINDOW WILL BE USED.
THE SYSTEM IS TO BE RUN AUTOMATICALLY.

```

; Här följer en utskrift av resultatvektorerna, vilka alltså listas på önskad logisk enhet.

FREQUENCY	GAIN	ARGUMENT	NUMBER OF
-----------	------	----------	-----------

0.10E+00	0.84E+00	-31.3	3.0
0.12E+00	0.79E+00	-36.0	3.0
0.15E+00	0.72E+00	-42.3	3.0
0.17E+00	0.68E+00	-45.4	3.0
0.20E+00	0.62E+00	-49.6	3.0
0.22E+00	0.58E+00	-52.4	3.0
0.25E+00	0.53E+00	-55.5	3.0
0.30E+00	0.46E+00	-59.6	3.0
0.35E+00	0.41E+00	-62.7	3.0
0.40E+00	0.36E+00	-65.5	3.0
0.45E+00	0.33E+00	-67.0	3.0
0.50E+00	0.29E+00	-68.5	3.0
0.60E+00	0.25E+00	-70.5	3.0
0.69E+00	0.22E+00	-72.0	3.0
0.78E+00	0.19E+00	-73.0	3.0
0.89E+00	0.17E+00	-73.5	3.0
0.96E+00	0.16E+00	-74.3	3.0

YOU HAVE NOW THE POSSIBILITY TO DO A NEW MEASUREMENT  
WITH SHORTER INTERVALS BETWEEN THE FREQUENCIES.  
DO YOU WANT TO DO THIS?N

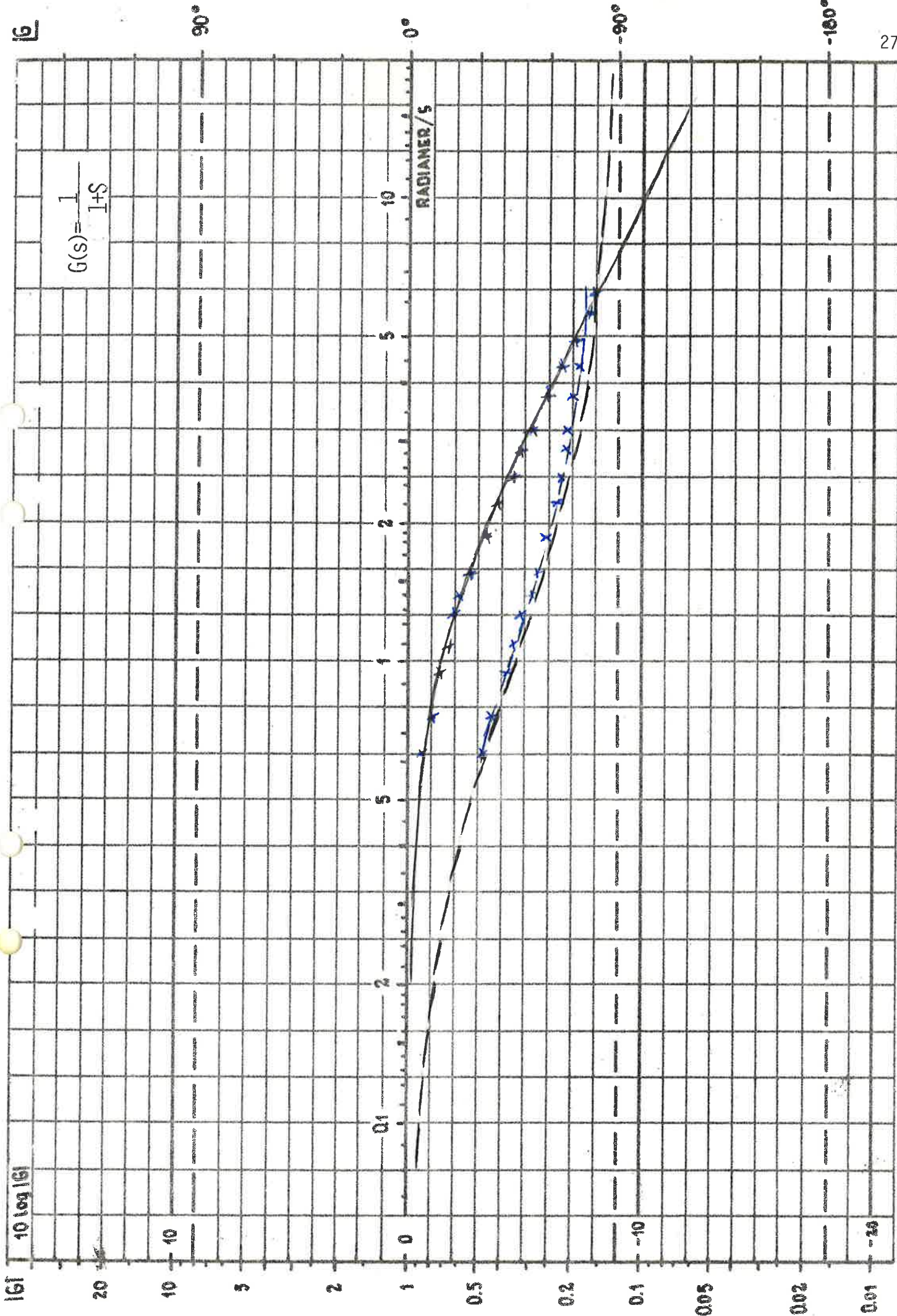
^C

; Genom att göra kommandot "control C"  
kommer man ur bakgrundsprogrammet och  
en ny körning kan utföras. Förgrunds-  
programmet är då fortfarande aktivt.

Kurvorna på nästa sida visar dels de korrekta värdena, dels de av  
ovanstående analys framkomna värdena. Observera att amplitud-kurvorna  
sammanfaller.

$10 \log |G|$

$$G(s) = \frac{1}{1+s}$$



## 4.2 Körning 2

Analys av systemet  $1/(s \cdot s + 0.1s + 1)$ .

Detta system har en resonanstopp vid 1 rad/s. Därför har manuell körning valts med möjlighet att ändra amplituden hos systemets insignal. Dessutom kan man vid manuell körning vänta med en ny frekvens tills egensvängningarna dött ut. Extra körning med tätare frekvenser kring resonanstoppen har också utnyttjats.

.SET USR NO SWAP↓

.FRUN DX1:FG/N!500↓

.RUN DX1:BG↓

DO YOU WANT FREQUENCY ANALYSIS?(Y=YES,N=NO)Y↓

\*\*\*\*\*  
\* WELCOME TO FREQUENCY ANALYSIS \*  
\*\*\*\*\*

SHORT DESCRIPTION:

THE COMPUTER HAS TWO OUTPUT SIGNALS AND NEEDS THREE INPUT SIGNALS.

ON OUTPUT 1(A) YOU HAVE A SINUS SIGNAL, WHICH IS THE INPUT SIGNAL TO THE SYSTEM YOU WANT TO ANALYSE.

THIS SIGNAL SHALL ALSO BE DIRECTLY COUPLED TO THE SECOND INPUT ON THE INTERFACE.

ON OUTPUT 2(B) YOU HAVE A COSINUS SIGNAL. THIS SIGNAL SHALL BE DIRECTLY COUPLED TO THE THIRD INPUT.

FINALLY YOU PUT THE OUTPUT SIGNAL FROM THE SYSTEM ON INPUT 1.

THE FOLLOWING QUESTION SHALL BE ANSWERED IN THE UNIT:

"NUMBER OF TICKS". 1 TICK=0.020 SEC.

MAIN SAMPLE PERIOD:4↓

NUMBER OF INPUT SIGNALS:3↓

NUMBER OF OUTPUT SIGNALS:2↓

PLEASE DEFINE ON WHICH INPUTS YOU HAVE THE DIFFERENT SIGNALS. THE FIRST SIGNAL IS THE OUTPUT SIGNAL FROM THE ANALYSED SYSTEM. THE SECOND SIGNAL IS THE INPUT SINUS SIGNAL TO THIS SYSTEM. THE THIRD SIGNAL IS THE COSINUS SIGNAL. THE ANSWER SHALL IN GENERAL BE: INPU1, INPU2, INPU3  
INPUT SIGNAL: INPU1, INPU2, INPU3↓

IF YOU ANSWER YES TO THE FOLLOWING QUESTION YOU HAVE TO CHOOSE AN INTEGRATION TIME OF 3 PERIODS.  
DO YOU WANT TREND ELIMINATION?N ↓

DO YOU WANT TIME WINDOW?Y↓

YOU HAVE NOW THE POSSIBILITY TO DECIDE IF THE SYSTEM IS TO BE RUN AUTOMATICALLY OR NOT. IF NOT, THE RUNNING OF THE ANALYSIS IS INTERRUPTED BETWEEN TWO FREQUENCIES, AND YOU CAN DECIDE YOURSELF WHEN IT IS TIME FOR A NEW FREQUENCY. YOU CAN ALSO AT THAT OCCASION CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL OF THE COMPUTER.

DO YOU WANT AUTOMATIC FREQUENCY ANALYSIS: N↓

ON WHICH DEVICE DO YOU WANT THE FINAL RESULT? (ANSWER TT OR LP ON THE FOLLOWING QUESTION!)  
OUTPUT ON DEVICE: TT↓

PLEASE DEFINE THE FREQUENCY LIMITS. IF YOU WANT TO WORK IN THE UNIT HZ ANSWER H AND IF YOU WANT RAD/S ANSWER R.

WHICH UNIT DO YOU WANT? R↓

LOWER FREQUENCY LIMIT: .4↓

HIGHER FREQUENCY LIMIT: 4↓

HOW MANY FREQUENCIES PER DECADE DO YOU WANT (POSSIBLE VALUES ARE 1, 2, 4, 8 AND 16)? 16↓

PLEASE CHOOSE HOW MANY PERIODS OF THE FREQUENCY THAT THE INTEGRATION IN THE CORRELATION METHOD WILL LAST. IF YOU ANSWER 0 THE NUMBER WILL BE SET DEPENDING ON THE FREQUENCY AS FOLLOWS:

- 1 IF  $F < 0.1\text{HZ}$  ( $0.6284\text{ RAD/SEC}$ )
- 3 IF  $0.1 < F < 0.2\text{HZ}$  ( $1.2568\text{ RAD/SEC}$ )
- 5 IF  $0.2 < F < 0.5\text{HZ}$  ( $3.1420\text{ RAD/SEC}$ )
- 10 IF  $0.5 < F < 1\text{HZ}$  ( $6.2840\text{ RAD/SEC}$ )
- 20 IF  $F > 1\text{HZ}$

INTEGRATION TIME (NUMBER OF PERIODS): 3↓

TRANSIENT TIME (SEC): 60↓

AMPLITUDE OF THE OUTPUT SIGNAL: .8↓

THE FREQUENCIES ARE GIVEN IN THE UNIT RAD/SEC.

LOWER FREQUENCY LIMIT= 0.40E+00

HIGHER FREQUENCY LIMIT= 0.40E+01

NUMBER OF FREQUENCIES PER DECADE= 16

INTEGRATION TIME (PERIODS)= 3

TRANSIENT TIME (SEC)= 60.0

NO TREND ELIMINATION WILL BE DONE.

TIME WINDOW WILL BE USED.

THE SYSTEM IS TO BE RUN MANUALLY.

AMPLITUDE OF THE OUTPUT SIGNAL: 0.80  
DO YOU WANT TO CHANGE ANY PARAMETERS? N↓

↓  
 >STAR ↓ ; Startar nodinterpreteringen i förgrundsprogrammet.

\*\*\*\*\*

FREQUENCY= 0.35E+00 ARGUMENT= -1.7 GAIN= 0.11E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.40E+00 ARGUMENT= -2.1 GAIN= 0.12E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.45E+00 ARGUMENT= -2.5 GAIN= 0.12E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.50E+00 ARGUMENT= -3.1 GAIN= 0.13E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.60E+00 ARGUMENT= -3.7 GAIN= 0.14E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? Y ↓

AMPLITUDE: .6 ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.70E+00 ARGUMENT= -6.6 GAIN= 0.18E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
 WHEN IT IS TIME FOR A NEW FREQUENCY.  
 IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? Y ↓

AMPLITUDE: .4 ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60): G ↓

FREQUENCY= 0.80E+00 ARGUMENT= -11.8 GAIN= 0.27E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .15↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.90E+00 ARGUMENT= -26.0 GAIN= 0.50E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .08↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.99E+00 ARGUMENT= -89.1 GAIN= 0.96E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .2↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.12E+01 ARGUMENT=-163.8 GAIN= 0.20E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .3↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.15E+01 ARGUMENT=-173.1 GAIN= 0.75E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.17E+01 ARGUMENT=-174.7 GAIN= 0.50E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .4↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.20E+01 ARGUMENT=-175.2 GAIN= 0.31E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G (G=60): G ↓

FREQUENCY= 0.22E+01 ARGUMENT=-174.9 GAIN= 0.26E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G (G=60): G ↓

FREQUENCY= 0.25E+01 ARGUMENT=-176.3 GAIN= 0.18E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G (G=60): G ↓

FREQUENCY= 0.30E+01 ARGUMENT=-177.2 GAIN= 0.12E+00

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G (G=60): G ↓

FREQUENCY= 0.36E+01 ARGUMENT=-178.7 GAIN= 0.82E-01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE  
WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL? N ↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G (G=60): G ↓

LAST FREQUENCY

FREQUENCY= 0.39E+01 ARGUMENT=-179.3 GAIN= 0.67E-01

DO YOU WANT TO SAVE THE RESULT ON A FILE? Y ↓

GIVE YOUR FILE A NAME.  
FOR EXAMPLE: DX1:MAT.DAT=  
\*DX1:FRE.ANA= ↓

; En fil för resultatvektorn bildas  
på massminne med nedanstående innehåll.

MAIN SAMPLE PERIOD=0.080SEC

THE FREQUENCIES ARE GIVEN IN THE UNIT RAD/SEC.

LOWER FREQUENCY LIMIT= 0.40E+00

HIGHER FREQUENCY LIMIT= 0.40E+01

NUMBER OF FREQUENCIES PER DECADE= 16

INTEGRATION TIME (PERIODS)= 3



TRANSIENT TIME (SEC)= 60.0

NO TREND ELIMINATION WILL BE DONE.

TIME WINDOW WILL BE USED.

THE SYSTEM IS TO BE RUN MANUALLY.

FREQUENCY	GAIN	ARGUMENT (DEG)	NUMBER OF MEASURE PERIODS
0.35E+00	0.11E+01	-1.7	3.0
0.40E+00	0.12E+01	-2.1	3.0
0.45E+00	0.12E+01	-2.5	3.0
0.50E+00	0.13E+01	-3.1	3.0
0.60E+00	0.14E+01	-3.7	3.0
0.70E+00	0.18E+01	-6.6	3.0
0.80E+00	0.27E+01	-11.8	3.0
0.90E+00	0.50E+01	-26.0	3.0
0.99E+00	0.96E+01	-89.1	3.0
0.12E+01	0.20E+01	-163.8	3.0
0.15E+01	0.75E+00	-173.1	3.0
0.17E+01	0.50E+00	-174.7	3.0
0.20E+01	0.31E+00	-175.2	3.0
0.22E+01	0.26E+00	-174.9	3.0
0.25E+01	0.18E+00	-176.3	3.0
0.30E+01	0.12E+00	-177.2	3.0
0.36E+01	0.82E-01	-178.7	3.0
0.39E+01	0.67E-01	-179.3	3.0

YOU HAVE NOW THE POSSIBILITY TO DO A NEW MEASUREMENT WITH SHORTER INTERVALS BETWEEN THE FREQUENCIES. DO YOU WANT TO DO THIS?Y↓

PLEASE DEFINE A NEW FREQUENCY INTERVAL. LOWER FREQUENCY LIMIT: .75↓

HIGHER FREQUENCY LIMIT: 1.2↓

THE MAXIMUM NUMBER OF FREQUENCIES ARE: 41 HOW MANY DO YOU (APPROXIMATELY) WANT? 10↓

THE ACTUAL NUMBER OF FREQUENCIES ARE: 10 DO YOU WANT TO CHANGE THIS NUMBER?N↓

↓  
>STAR↓

\*\*\*\*\*

FREQUENCY= 0.75E+00 ARGUMENT= -9.2 GAIN= 0.22E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE WHEN IT IS TIME FOR A NEW FREQUENCY. IN THE MEANWHILE: DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .3↓ ; Amplituden hos datorns utsignal.

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER @ (0=60): 6↓

FREQUENCY= 0.78E+00 ARGUMENT= -10.0 GAIN= 0.26E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

; Man har möjlighet att justera antalet frekvenser. Önskat antal överensstämmer nämligen inte alltid med vad man verkligen får.

WHEN IT IS TIME FOR A NEW FREQUENCY,  
IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.81E+00 ARGUMENT= -13.2 GAIN= 0.28E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.84E+00 ARGUMENT= -15.2 GAIN= 0.34E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .2↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.88E+00 ARGUMENT= -22.6 GAIN= 0.44E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .1↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.92E+00 ARGUMENT= -31.4 GAIN= 0.61E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.97E+00 ARGUMENT= -63.3 GAIN= 0.92E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .08↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.10E+01 ARGUMENT=-113.4 GAIN= 0.86E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY,

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.11E+01 ARGUMENT=-147.5 GAIN= 0.47E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?Y↓

AMPLITUDE: .2↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

FREQUENCY= 0.11E+01 ARGUMENT=-159.2 GAIN= 0.31E+01

THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE

WHEN IT IS TIME FOR A NEW FREQUENCY.

IN THE MEANWHILE:

DO YOU WANT TO CHANGE THE AMPLITUDE OF THE OUTPUT SIGNAL?N↓

WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY ANSWER G(G=60):G↓

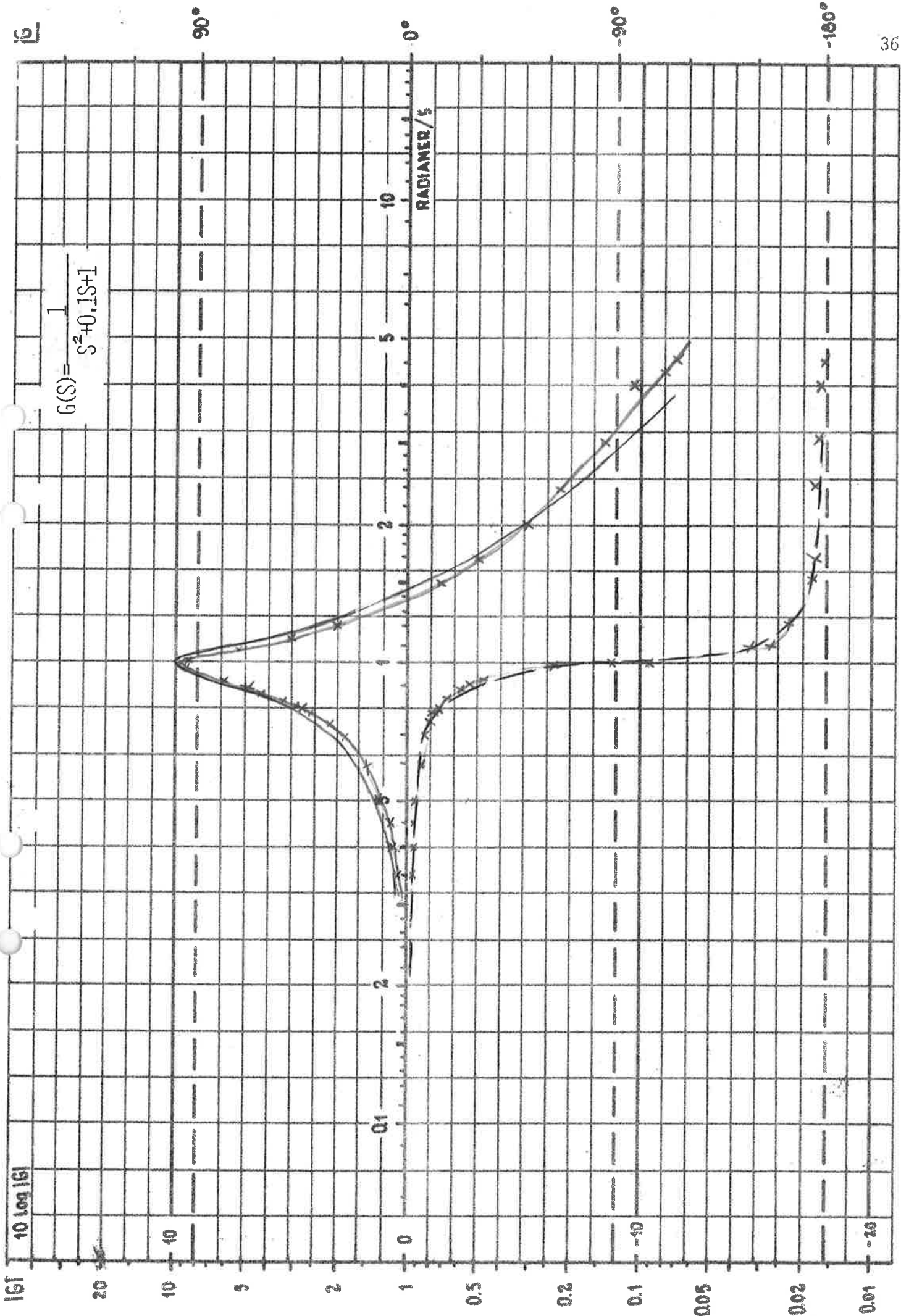
FREQUENCY= 0.12E+01 ARGUMENT=-163.5 GAIN= 0.21E+01

FREQUENCY	GAIN	ARGUMENT (DEG)	NUMBER OF MEASURE PERIODS
0.75E+00	0.22E+01	-9.2	3.0
0.78E+00	0.26E+01	-10.0	3.0
0.81E+00	0.28E+01	-13.2	3.0
0.84E+00	0.34E+01	-15.2	3.0
0.88E+00	0.44E+01	-22.6	3.0
0.92E+00	0.61E+01	-31.4	3.0
0.97E+00	0.92E+01	-63.3	3.0
0.10E+01	0.86E+01	-113.4	3.0
0.11E+01	0.47E+01	-147.5	3.0
0.11E+01	0.31E+01	-159.2	3.0
0.12E+01	0.21E+01	-163.5	3.0

^C

Kurvorna på nästa sida visar dels de korrekta värdena, dels de ur ovanstående analys framkomna värdena.

Avvikelserna från de korrekta värdena härrör till viss del från det faktum att man på analogmaskinen använt en potentiometer vars värde ej kan inställas exakt, varvid man får ett system vars överföringsfunktion skiljer sig något från  $G(s)=1/(s^2+0.1s+1)$ .



### 4.3 Körning 3

Läsning av resultatfilen från körning 2 i kapitel 4.2.

.SET USR NO SWAP↓

.RUN DX1:DUMP↓

GIVE THE NAME OF THE FILE YOU WANT TO READ. FOR EXAMPLE:DX1:MAT.DAT=  
\*DX1:FRE.ANA=↓

ON WHICH DEVICE (TT OR LP) DO YOU WANT THE RESULT?

OUTPUT ON DEVICE:TI↓

; Logisk enhet för utskriften.

MAIN SAMPLE PERIOD WAS 0.080SEC

; Utskrift av resultatfilen.

THE FREQUENCIES ARE GIVEN IN THE UNIT RAD/SEC.

LOWER FREQUENCY LIMIT= 0.40E+00

HIGHER FREQUENCY LIMIT= 0.40E+01

NUMBER OF FREQUENCIES PER DECADE= 16

INTEGRATION TIME (PERIODS)= 3

TRANSIENT TIME (SEC)= 60.0

NO TREND ELIMINATION HAS BEEN DONE.

TIME WINDOW HAS BEEN USED.

THE SYSTEM HAS BEEN RUN MANUALLY.

FREQUENCY	GAIN	ARGUMENT (DEG)	NUMBER OF MEASURE PERIODS
0.35E+00	0.11E+01	-1.7	3
0.40E+00	0.12E+01	-2.1	3
0.45E+00	0.12E+01	-2.5	3
0.50E+00	0.13E+01	-3.1	3
0.60E+00	0.14E+01	-3.7	3
0.70E+00	0.18E+01	-6.6	3
0.80E+00	0.27E+01	-11.8	3
0.90E+00	0.50E+01	-26.0	3
0.99E+00	0.96E+01	-89.1	3
0.12E+01	0.20E+01	-163.8	3
0.15E+01	0.75E+00	-173.1	3

0.17E+01	0.50E+00	-174.7	3
0.20E+01	0.31E+00	-175.2	3
0.22E+01	0.26E+00	-174.9	3
0.25E+01	0.18E+00	-176.3	3
0.30E+01	0.12E+00	-177.2	3
0.36E+01	0.82E-01	-178.7	3
0.39E+01	0.67E-01	-179.3	3

THE RESULT OF A SECOND RUN WITH EXTRA FREQUENCIES:

FREQUENCY	GAIN	ARGUMENT (DEG)	NUMBER OF MEASURE PERIODS
0.75E+00	0.22E+01	-9.2	3
0.78E+00	0.26E+01	-10.0	3
0.81E+00	0.28E+01	-13.2	3
0.84E+00	0.34E+01	-15.2	3
0.88E+00	0.44E+01	-22.6	3
0.92E+00	0.61E+01	-31.4	3
0.97E+00	0.92E+01	-63.3	3
0.10E+01	0.86E+01	-113.4	3
0.11E+01	0.47E+01	-147.5	3
0.11E+01	0.31E+01	-159.2	3
0.12E+01	0.21E+01	-163.5	3

STOP --

#### 4.4 Exempel på trendelimination

Vid en analys av systemet  $1/(s+1)$  har en linjär trend adderats till utsignalen.

Trenden kan beskrivas genom följande ekvation:

$$y=0.22t \quad \text{där } t=\text{tiden.}$$

Nedan visas denna trends inverkan på slutresultatet och hur man kan minska effekterna av denna störning.

Resultat av frekvensanalys av ovanstående system samt teoretiskt framräknade värden:

Frekvens (rad/s)	Utan trendelimination		Med trendelimination		Exakta värden	
	Förstärkning (ggr)	Fas (grader)	Förstärkning	Fas	Förstärkning	Fas
0.80	1.1	-23.8	0.80	-34.9	0.78	-38.7
0.90	1.0	-27.4	0.76	-38.8	0.74	-42.0
0.99	0.97	-29.8	0.72	-41.6	0.71	-44.7
1.20	0.83	-34.6	0.64	-47.1	0.64	-50.2
1.50	0.69	-39.8	0.55	-53.0	0.55	-56.3
1.70	0.61	-43.2	0.50	-56.3	0.51	-59.5
2.00	0.53	-46.0	0.44	-59.5	0.45	-63.4
2.20	0.49	-47.3	0.41	-61.3	0.41	-65.6
2.50	0.43	-49.9	0.36	-64.2	0.37	-68.2
3.00	0.36	-51.8	0.31	-67.5	0.32	-71.6
3.60	0.31	-53.7	0.26	-70.0	0.27	-74.5
3.90	0.28	-54.3	0.24	-69.5	0.25	-75.6

## REFERENSER

- (1) -K. J. Åström  
Rapport 7504  
Lectures on System Identification, Chapter 3  
Frequency Response Analysis
- (2) -Conny Persson  
Rapport 122  
Frekvensanalys
- (3) -T. Paulsson  
L. Rading  
Rapport LUTFD2/(TFRT-5202)/1-174/(1977)  
Program for logging and data Analysis on PDP11/03



## APPENDIX A

Beskrivning av resultatfilen

Efter avslutad behandling av mätvärdena i det önskade frekvensintervall har man möjlighet att lagra resultatet på en fil. Namnet på denna fil bestämmer man själv, varvid man får en möjlighet att skilja på mätvärdena häröfande från olika experiment.

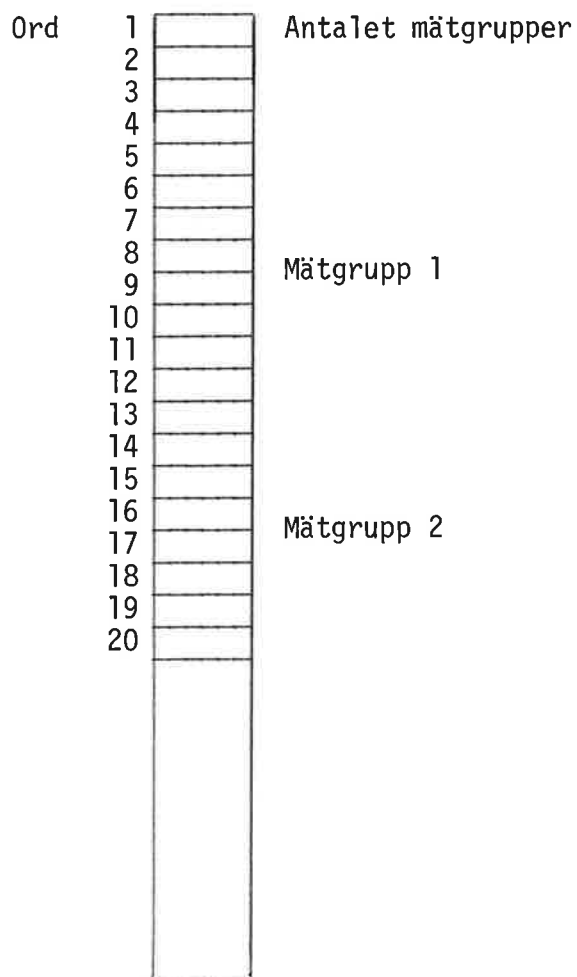
I de första 20 orden i filen lagras för experimentet i fråga karakteristiska data. I ord 21 lagras antalet mätgrupper. Varje mätgrupp omfattar 4 reella tal (8 ord), nämligen frekvens, förstärkning, argument samt det antal perioder över vilka mätningen skett.

Filen är indelad i block om 256 ord vardera. För att ett helt antal mätgrupper skall lagras i första blocket, påbörjas lagringen av mätdata i ord 25. Antalet frekvenser under en körning är maximerat till 50, vilket medför att högst två block av filen tas i anspråk.

Om man efter avslutad ordinarie körning begär körning med tätare frekvenser under ett visst intervall, kommer dessa mätresultat att lagras i samma fil men med början i ett nytt block. Först i detta block lagras antalet nya mätgrupper. I ord 5 och följande lagras därefter mätresultatet på samma sätt som tidigare.

Ord	1		
	2		
	3		
	4	Lägre frekvensgräns	
	5		
	6	Högre frekvensgräns	
	7		
	8	Antal frekvenser per dekad	
	9		
	10	Mättid	
	11		
	12	Transienttid	
	13		
	14	Uppgift om trendelimination	
	15		
	16	Uppgift om tidsfönster	
	17		
	18	Uppgift om automatisk eller manuell körning	
	19		
	20	Huvudsamplingsperiod	
	21	Antal mätgrupper	
	22		
	23		
	24		
	25	Frekvens	
	26		
	27	Förstärkning	
	28		
	29	Argument	Mätgrupp 1
	30		
	31	Antal mätperioder	
	32		
	33		
	34		
	35		
	36		
	37	Mätgrupp 2	
	38		
	39		
	40		

Figur A1. Figuren illustrerar innehållet i resultatfilens första block.



Figur A2. Figuren illustrerar innehållet i det block som bildas då körning med tätare frekvenser begärts.

## APPENDIX B

Länkning

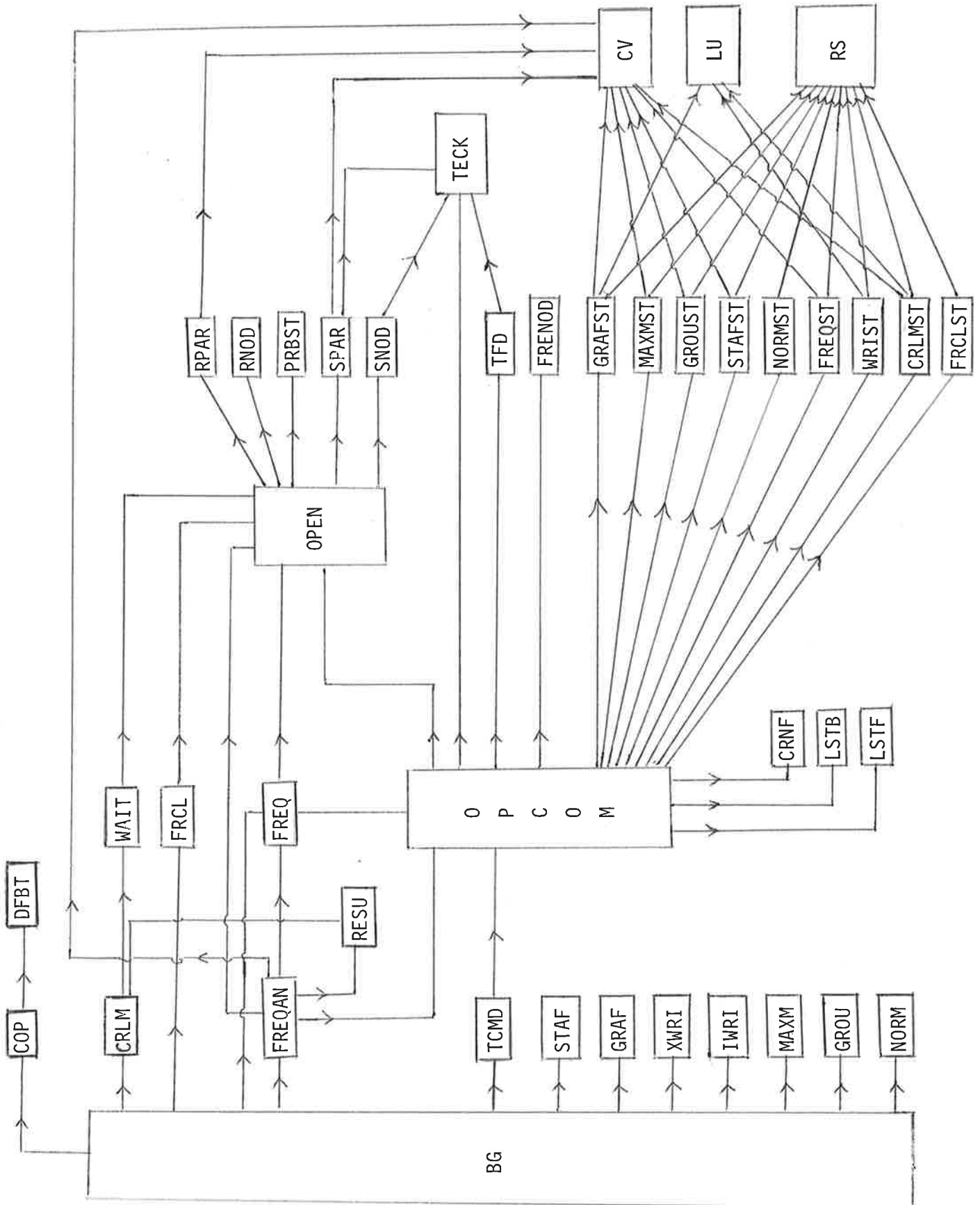
För att primärminnet på 28 k ord skall vara tillräckligt stort, måste bakgrundsprogrammen vid länkningen delas upp i en minnesresident del och ett antal så kallade overlays. Varje overlay kan innehålla fyra programblock vilka delar minnesutrymme med varandra. Program, vilka ligger på en viss overlay-nivå, får ej anropa program som ligger på en nivå med lägre nummer. Naturligtvis får ej heller anrop ske mellan program som ligger på samma overlay-nivå men i olika programblock. Vid indelningen av programmen i olika block och nivåer måste man också vara uppmärksam på att om ett program i en overlay anropar ett minnesresident program, så får detta i sin tur ej anropa ett program som ligger på samma overlay-nivå men i ett annat programblock än det första programmet i anropskedjan.

Vid körning av systemet innehållande rutiner för frekvensanalys har länkningen skett med en indelning i tre overlay-nivåer enligt tabell nedan.

Minnesresidenta rutiner: BG, COP, DFBT, TCMD, OPCOM, CV, WW och BELL

Nivå 1:	FREQAN FREQ	WAIT CRLM	GRAFST WRIST MAXMST GROUST STAFST NORMST	GRAF XWRI IWRI MAXM GROU STAF NORM
Nivå 2:	OPEN SPAR	FREQST	CRLMST	TFD FRCLST
Nivå 3:	RPAR RNOD RESU	SNOD TECK	LU RS PRBST	CRNF LSTB LSTF FRENOD

Figur B1. Figuren visar hur bakgrundsprogrammen anropar varandra.



## APPENDIX C

This appendix contains all new or revised program in alphabetical order.

<u>NAME</u>	<u>SUBTITLE</u>	<u>PAGE</u>
BD	BLOCK DATA	C1
BG	MAIN PROGRAM IN BACKGROUND JOB	C3
CRLM	CALCULATES THE GAIN AND THE ARGUMENT USING THE CORRELATION METHOD	C9
CRLMST	START-UP ROUTINE TO THE ROUTINE CRLM	C14
DUMP	READS A RESULT FILE AND WRITES IT ON SPECIFIED DEVICE	C17
FG	MAIN PROGRAM IN FOREGROUND JOB	C21
FREQ	GENERATES FREQUENCIES AND SETS PARA- METERS IN THREE FOREGROUND NODES	C28
FREQAN	START-UP FOR FREQUENCY ANALYSIS	C33
FREQST	START-UP ROUTINE TO THE ROUTINE FREQ	C37
FRCL	GENERATES MORE FREQUENCIES	C41
FRCLST	START-UP ROUTINE TO THE ROUTINE FRCL	C45
OPCOM	OPERATOR COMMUNICATION ROUTINE	C48
OPEN	SETS A NODE HEAD AND PARAMETERS	C53
RESO	WRITES THE RESULT HEAD	C58
RESU	WRITES THE RESULT HEAD	C60
RFIL	STORES THE RESULT ON A FILE	C62
RNOD	SETS A NODE HEAD	C65
RPAR	SETS PARAMETERS	C67
TIME	STARTS AND STOPS THE SENDING OF DATA TO BACKGROUND JOBS	C69
WAIT	PAUSE IN FREQUENCY ANALYSIS	C71

NAME: BD  
-----

SUBTITLE: BLOCK DATA  
-----

LANGUAGE: FORTRAN + RT-11-FORTRAN  
-----

IMPLEMENTOR: LEIF RADING, TYKE PAULSSON  
-----

DATE: 6-SEP-77  
-----

REVISED BY: CHRISTER LARSSON, HARRIET LINDGREN  
-----

DATE: 28-APR-78  
-----

INSTITUTE:  
-----

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, LUND

PURPOSE  
=====

ESTABLISH AND DEFINE COMMON BLOCKS, ASSIGN VARIABLES  
AND ARRAYS TO THOSE BLOCKS, AND ASSIGN INITIAL VALUES TO  
THE COMPONENTS

USAGE  
=====

COMMON BLOCKS:  
-----

BG        -CONTAINING LIST- AND PARAMETERARRAYS FOR  
          BACKGROUND JOB  
CMD       -CONTAINING 4-CHARACTERS COMMANDNAME AND 2-  
          CHARACTERS PARAMETERNAME  
IM        -CONTAINING 1-CHARACTER CONSTANTS  
RD        -CONTAINING THE NUMBER OF PARAMETERS OF THE  
          NODETYPES IN THE FOREGROUND JOB  
FG        -CONTAINING ARRAYS FOR SENDING AND RECEIVING  
          MESSAGES AND INFORMATION BETWEEN THE JOBS

NOTE:  
-----

SAY THAT A ROUTINE HAS:

1.        TOTALLY N1 PARAMETERS
2.        N2 NOT STATUSPARAMETERS

THEN

IRDC( )=100\*N1+N2

IF THE NODETYPE HAS A START-UP ROUTINE

IRDC( )=-100\*N1+N2

BLOCK DATA

```
C      LOGICAL*1 ICM
C
C      COMMON /BG/ LIST(120)
C      COMMON /CMD/CM1(20),CM2(20),CM3(20),IOP(100)
C      COMMON /IM/ ICM(7)
C      COMMON /RD/ IRD(20)
C
C      DATA LIST /120*-1/
C
C      DATA CM1/4HOPEN,4HACTI,4HSTOP,4HSTOR,4HSTAR,4HSTOC,4HSTAC,
C      *4HCRNF,4HLSTF,4HLSTB,4HCHSP,9*4H&##%/
C
C      DATA CM2/4HGRAF,4HXWRI,4HMAXM,4HIWRI,4HGROU,4HSTAF,4HNORM
C      *,4HFREQ,4HCRLM,4HFRCL,8*4H&##%,4HOUTP,4HINPU/
C
C      DATA CM3/4HPIDR,4HTIME,4HSTEP,4HADDS,4HRAMP,4HSINU,4HPULS,
C      *3*4H&##%,4HFRBG,4HNODI,7*4H&##%,4HIGJN/
C
C      DATA IOP/2HG=,2HTI,2HTD,2HG0,2HY0,2HRI,2HD=,2HY=,
C      *2HTT,2HPS,2HPA,
C      *2HAM,2HX=,2HY=,
C      *2HW1,2HW2,2HW3,2HY=,
C      *2HXK,2HX=,2HY=,
C      *2HAM,2HFR,2HDF,2HX=,2HY=,
C      *2HAM,2HFR,2HPL,2HX=,2HY=,
C      *2HAM,2HY=,
C      *2HXM,2HSG,2HY=,
C      *62*2H ,
C      *2HLI,2HCL/
C
C      DATA ICM/1H.,1H/,1H$,1H!,1H=,1H /
C
C      DATA IRD/805,303,301,403,301,503,503,3*0,-201,-302,7*0,2424/
C
C      END
```



NAME: BG

SUBTITLE: MAINPROGRAM IN BACKGROUND JOB

LANGUAGE: FORTRAN + RT-11-FORTRAN

IMPLEMENTOR: LEIF RADING, TYKE PAULSSON

DATE: 7-SEP-77

REVISED BY: CHRISTER LARSSON, HARRIET LINDGREN

DATE: 29-APR-78

INSTITUTE:

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

PURPOSE AND METHOD

BG STARTS BACKGROUND JOB BY ASKING QUESTIONS AND USING THE OPERATORS ANSWERS, BY ORGANICZING THE COMMUNIZATION WITH MASS STORAGE AND BY ALLOCATING SPACE ON MASS STORAGE. THE PROGRAM ORGANIZES THE TRANSFERING OF THE COLLECTED MEASUREMENT DATA, THE TRANSFERING OF CALCULATED DATA AND THE RUNNING OF CALCULATING PROGRAMS. FURTHER IT ORGANIZES THE TRANSFERING OF AND INFORMATION TO AND FROM FOREGROUND JOB AND THE SUBROUTINE OPCOM AND THE EXECUTION OF COMMANDS FROM OPCOM.

USAGE

COMMON BLOCKS:

- BG            --CONTAINING LIST- AND PARAMETERARRAYS FOR BACKGROUND JOB
- BUFF          --CONTAINING BUFFERS FOR TRANSFERING DATA BETWEEN FOREGROUND JOB, BACKGROUND JOB AND MASS STORAGE.
- LINK          --CONTAINING LINKAGE INFORMATION FOR COMPLETITION ROUTINES.
- PAV,FLA      --CONTAINING FLAGS AND POINTERS WITH INFORMATION ABOUT COLLECTING AND CALCULATING BLOCKS.
- CMD           --CONTAINING 4-CHARACTERS COMMANDNAME AND 2-CHARACTERS PARAMETERNAME.
- IM            --CONTAINING 1-CHARACTER CONSTANTS.
- RD            --CONTAINING THE NUMBER OF PARAMETERS OF THE NODETYPES IN THE FOREGROUND JOB.
- FG            --CONTAINING ARRAYS FOR SENDING AND RECEIVING MESSAGES AND INFORMATION BETWEEN THE JOBS.

WRK        -WORK AREA

NEW COMMON BLOCKS:

FFLAG      -CONTAINING FLAGS.

FRAN       -CONTAINING INFORMATION ABOUT PERIOD, NUMBER OF INPUT-  
AND OUTPUT SIGNALS.

RP         -CONTAINING INFORMATION ABOUT PARAMETERS IN FOREGROUND  
JOBS.

IFAP       -CONTAINING FLAGS.

BUFC       -CONTAINING THE RESULT VECTORS.

CRL        -CONTAINING PARAMETERS AND FLAGS WITH INFORMATION ABOUT  
FREQUENCY ANALYSIS.

NUSA       -CONTAINING INFORMATION ABOUT THE FREQUENCIES.

FRC        -CONTAINING INFORMATION ABOUT INTEGRATION TIME  
AND TRANSIENT TIME.

FILE       -CONTAINING INFORMATION ABOUT THE RESULT FILE.

CHARACTERISTICS

SIZE:

PDP11/03: 632 WORDS

```
*****
*                                     *
*           BACKGROUND PROGRAM       *
*                                     *
*****
```

```
EXTERNAL DFBT, COP
DIMENSION DF( 2 ), IEK( 12 )
DIMENSION X1( 5 )
LOGICAL*1 ICM
LOGICAL*1 II
LOGICAL*1 IVAL
```

```
COMMON /BG/ LIST( 120 ), IPA( 512 )
COMMON /BUFF/ IB1( 100 ), IB2( 512 ), IB4( 256 )
COMMON /LINK/ LI1( 4 ), LI2( 4 )
COMMON /PAV/ I1, I2, I3, I4, I5, I6, I7, I8, IC1, IR1,
*NB1, IR2, NB2, IC2, I9, I10
COMMON /FLA/ IFL1, ICN, NBN, ICO, IFL2
COMMON /CMD/ CM1( 20 ), CM2( 20 ), CM3( 20 ), IOP( 100 )
COMMON /IM/ ICM( 7 )
COMMON /RD/ IRD( 20 )
COMMON /FG/ IV( 100 )
COMMON /WRK/ IW1( 50 ), IW2( 50 ), IW3( 50 ), IW4( 50 )
COMMON/FFLAG/ IFAN, IFG, IBG
COMMON/FRAN/ IPER, IWI( 10 ), IWO( 10 )
```

```

COMMON/RF/ KZ, IWP( 10 ), FIW( 10 )
COMMON/IPAF/ LI, II
COMMON/BUFC/ RES( 200 ), RES1( 10 )
COMMON/CRL/ MPA, MCO, NYF, MR, W, SG, MPE, AM, ARG1
COMMON/NUSA/MP2, MSP, IHR, AN( 50 )
COMMON/FRC/MPI, TT
COMMON/FILE/ IC, NB, IR

```

```

C
EQUIVALENCE ( IEK( 3 ), K1 ), ( IEK( 4 ), K2 ), ( IEK( 5 ), K3 ), ( IEK( 6 ), K4 ),
*( IEK( 7 ), K5 ), ( IEK( 8 ), K6 ), ( IEK( 9 ), K7 ), ( IEK( 10 ), K8 ), ( IEK( 11 ), K9 )

C
DATA DF/6RDX1WOR, 6RK DAT/

C
C
C ALLOCATE TWO RT-11 CHANNELS AND MARKS THEM
C IN USE FOR THE FORTRAN I/O SYSTEM
C
ICN=IGETC( )
ICO=IGETC( )

C
C ALLOCATE SPACE FOR NBN BLOCKS ON FILE NAMED
C DX1:WORK.DAT AND CREATES A TENTATIVE ENTRY
C FOR IT.
C NBN IS EQUAL THE LARGER OF EITHER ONE-HALF THE
C LARGEST EMPTY SEGMENT OR THE ENTIRE SECOND
C LARGEST EMPTY SEGMENT.
C
NBN=IENTER( ICN, DF, 0 )
IF( NBN, LE, 0 ) STOP' ENTER FAILURE'
NBN=NBN-1

C
C ADD AVAILABLE ELEMENTS TO THE QUEUE.
C
IF( IQSET( 10 ), NE, 0 ) STOP' NOT ENOUGH FREE SPACE FOR
* QUEUE ELEMENTS'

C
C READ AND SEND INFORMATION ABOUT MAIN SAMPLE PERIOD,
C NUMBER OF INPUT SIGNALS AND OUTPUT SIGNALS.
C
IB1( 1 )=13
IB1( 2 )=0
IFB=0
IBG=0

C
600 WRITE( 7, 601 )
601 FORMAT( 1X, $, ' DO YOU WANT FREQUENCY ANALYSIS?( Y=YES, N=NO)' )
READ( 5, 602 ) IVAL
602 FORMAT( A1 )
IF( IVAL, EQ, 89, OR, IVAL, EQ, 78 ) GO TO 401
WRITE( 7, 502 )
GO TO 600

C
C
401 IF( IVAL, EQ, 78 ) GO TO 1
WRITE( 7, 402 )
402 FORMAT( 1X, ' *****' /
* 1X, ' * WELLCOME TO FREQUENCY ANALYSIS *' /
* 1X, ' *****' // )
WRITE( 7, 403 )
403 FORMAT( 1X, ' SHORT DESCRIPTION:' /
* 1X, ' THE COMPUTER HAS TWO OUTPUT SIGNALS AND NEEDS THREE' /
* 1X, ' INPUT SIGNALS.' )
WRITE( 7, 404 )
404 FORMAT( 1X, ' ON OUTPUT 1(A) YOU HAVE A SINUS SIGNAL, WHICH IS THE' /
* 1X, ' INPUT SIGNAL TO THE SYSTEM YOU WANT TO ANALYSE.' )

```

```

WRITE(7,405)
405  FORMAT(1X,'THIS SIGNAL SHALL ALSO BE DIRECTLY COUPLED TO THE'//
      *1X,'SECOND INPUT ON THE INTERFACE.')
WRITE(7,406)
406  FORMAT(1X,'ON OUTPUT 2(B) YOU HAVE A COSINUS SIGNAL.THIS SIGNAL'//
      *1X,'SHALL BE DIRECTLY COUPLED TO THE THIRD INPUT.')
WRITE(7,407)
407  FORMAT(1X,'FINALLY YOU PUT THE OUTPUT SIGNAL FROM THE'//
      *1X,'SYSTEM ON INPUT 1.'//)
WRITE(7,408)
408  FORMAT(1X,'THE FOLLOWING QUESTION SHALL BE ANSWERED IN THE UNIT:'//
      *1X,' "NUMBER OF TICKS".1 TICK=0.020 SEC.')

C
C
1  WRITE(7,501)
501  FORMAT(1X,*, 'MAIN SAMPLE PERIOD:')
      READ(5,5) IB1(3)
      MSP=IB1(3)
      IF (IB1(3).GE.2.AND. IB1(3).LE.32767) GO TO 2

C
      WRITE(7,502)
502  FORMAT(1X,'?ILL CMD?')
      GO TO 1

C
2  WRITE(7,503)
503  FORMAT(1X,*, 'NUMBER OF INPUT SIGNALS:')
      READ(5,5) IB1(4)
      IF (IB1(4).GE.1.AND. IB1(4).LE.8) GO TO 3
      WRITE(7,502)
      GO TO 2

C
3  WRITE(7,504)
504  FORMAT(1X,*, 'NUMBER OF OUTPUT SIGNALS:')
      READ(5,5) IB1(5)
      IF (IB1(5).GE.1.AND. IB1(5).LE.4) GO TO 4
      WRITE(7,502)
      GO TO 3

C
4  CALL ISDAT(IB1,5)
5  FORMAT(I10)

C
C  CONVERT A 2-WORD INTERNAL TIME FORMAT TO INTEGER*4 FORMAT
C  AND THEN CONVERT IT TO REAL*4 VALUE
C
      CALL JJCVT(IB1)
      TSM=AJFLT(IT1)*0.02

C
C
      I1=0
      NI=IB1(4)*2+2
      NU=IB1(5)*2
      I2=NI+NU
      NU=NU+18
      I3=256/I2
      I4=0
      I5=0
      I6=1
      I7=1
      I10=0
      IR1=-1
      IR2=-1
      NB1=NBN
      NB2=NBN
      IC1=ICN

```

IC2=ICN  
IFL1=0  
MR=0  
ARG1=0.

```

C
C
C   ENTER THE SUBROUTINE DFBT WHEN A MESSAGE FROM
C   FOREGROUND JOB IS RECEIVED.
C
C   CALL IRCVDF( IB1, 56, LI1, DFBT )
C
C
C   FREQUENCY ANALYSIS?
C
C   IF( IVAL. NE. 89 ) GO TO 10
C   CALL FREQAN
C
C   SET FLAG NO DATA AVAILABLE.
C
10  I9=0
C
C   MESSAGE FROM CONSOLE TERMINAL?
C
C   CALL TCMD
C
C   SCHEDULE THE SUBROUTINE COP TO BE RUN AS
C   AN ASYNCHRONOUS ROUTINE.
C
C   CALL ITIMER( 0, 0, 0, 0, LI2, ID, COP )
C
C   WAIT UNTIL DATA IS AVAILABLE.
C
15  IF( I9 ) 10, 15, 16
C
C   SUSPEND EXECUTION UNTIL ALL INPUT/OUTPUT ON CHANNEL IC2
C   ARE COMPLETE.
C
16  CALL IWAIT( IC2 )
C   DO 100 I=17, 18
C
C   MESSAGE FROM CONSOLE TERMINAL?
C
C   CALL TCMD
C
C   COPY ONE SAMPLE BLOCK FROM VECTOR IB4 TO VECTOR IPA
C
C   J1=( I-1 ) * I2
C   DO 17 K=1, NI
C   J1=J1+1
17  IPA( K ) = IB4( J1 )
C
C   DO 18 K=19, NU
C   J1=J1+1
18  IPA( K ) = IB4( J1 )
C
C   RUN ACTIVATED NODS.
C
C   DO 100 K=1, 120, 12
C   IF( LIST( K ) ) 100, 200, 300
200  J=LIST( K+2 ) / 1000
C
C   DO 210 J2=3, 11
210  IEK( J2 ) = LIST( K+J2 )
C   LIST( K ) = LIST( K+1 )
C

```

GO TO ( 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,  
\*230, 230, 230, 230, 230, 230, 230, 230, 230, 230 ), J

C

```
211 CALL GRAF( IPA(K6 ), IPA(K1 ), IPA(K2 ), IPA(K3 ), IPA(K4 ),  
*IPA(K5 ), IW1 )  
GO TO 300  
212 CALL XWRI( IPA(K6 ), IPA(K7 ), IPA(K8 ), IPA(K9 ), IPA(K1 ), CM1,  
*IPA(K5 ), IPA, IW1 )  
GO TO 300  
213 CALL MAXM( IPA(K6 ), IPA(K2 ), IPA(K4 ), IPA(K1 ), IPA(K3 ), IPA(K5 ))  
GO TO 300  
214 CALL IWRI( IPA(K6 ), IPA(K7 ), IPA(K8 ), IPA(K9 ), IPA(K1 ), CM1,  
*IPA(K5 ), IPA, IW1 )  
GO TO 300  
215 CALL GROU( IPA(K6 ), IPA(K2 ), IPA(K3 ), IPA(K5 ), IPA(K1 ))  
GO TO 300  
216 CALL STAF( IPA(K6 ), IPA(K1 ), IPA(K2 ), IPA(K3 ), IPA(K4 ),  
*IPA(K5-2 ), IPA(K5 ))  
GO TO 300  
217 CALL NORM( IPA(K6 ), IPA(K7 ), IPA(K8 ), IPA(K1 ))  
GO TO 300  
218 IF( NYF.NE.1 ) GO TO 300  
CALL FREQ( IPA(K1 ), IPA(K2 ),  
*IPA(K5-4 ), IPA(K3 ), IPA(K5-3 ), IPA(K5-2 ), IPA(K5-1 ), IPA(K5 ), IPA(K4 ))  
IR2=-1  
I10=0  
GO TO 300  
219 CALL CRLM( IPA(K6 ), IPA(K7 ), IPA(K8 ), IPA(K3 ),  
*IPA(K1 ), IPA(K2 ), IPA(K4 ), IPA(K5 ), IPA(K5-1 ),  
*IPA(K5-2 ), IPA(K5-3 ), IPA(K5-4 ))  
GO TO 300  
220 IF( NYF.NE.1 ) GO TO 300  
CALL FRCL( IPA(K1 ), IPA(K2 ), IPA(K5 ))  
IR2=-1  
I10=0  
GO TO 300  
230 CONTINUE  
300 LIST(K)=LIST(K)-1  
100 CONTINUE  
C  
I7=I8+1  
IF( I8.EQ. I3 ) I7=1  
C  
C SET FLAG NO DATA AVAILABLE.  
C  
I9=0  
GO TO 10  
C  
STOP  
END
```

NAME: CRLM

SUBTITLE: CALCULATES THE GAIN AND THE ARGUMENT USING THE  
CORRELATION METHOD

LANGUAGE: FORTRAN IV

KEYWORDS: CORRELATION METHOD

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDBREN      DATE: 26-APR-78

INSTITUTE:

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:

VERSION:

PURPOSE:  
=====

THE ROUTINE:

1. CALCULATES THE GAIN AND ARGUMENT USING THE CORRELATION METHOD.
2. DELETES THE OLD WORK FILE AND CREATES A NEW.

USAGE:  
=====

PROGRAM TYPE: SUBROUTINE

ARGUMENTS:  
-----

CRLM( Y, SI, CO, T, SYS, SYC, ITW, ITE, IAU, LUN, IFST, IFIL )

- Y            -INPUT VALUE 1
- SI           -INPUT VALUE 2 FROM THE SINUSGENERATOR
- CO           -INPUT VALUE 3 FROM THE COSINUSGENERATOR
- T            -THE TIME
- SYS, SYC    -SUMS IN THE CORRELATION METHOD
- ITW          -INFORMATION ABOUT TIME WINDOW
- ITE          -INFORMATION ABOUT TREND ELIMINATION
- IAU          -INFORMATION ABOUT AUTOMATIC OR MANUAL RUN
- LUN          -LOGICAL UNIT
- IFST        -FLAG
- IFIL        -INFORMATION ABOUT IF A FIL IS REQUESTED

COMMON BLOCKS:  
-----

```

C      BUFC      -CONTAINING THE RESULTVECTORS
C
C      CRL       -CONTAINING PARAMETERS AND FLAGS WITH INFORMATION
C                  ABOUT FREQUENCY ANALYSIS
C
C      PAV       -CONTAINING INFORMATION ABOUT THE FREQUENCIES
C
C      PAV,FLA   -CONTAINING FLAGS AND POINTERS WITH INFORMATION
C                  ABOUT COLLECTING AND CALCULATING BLOCKS.
C
C      WRK       -WORK AREA
C
C      FFLAG     -CONTAINING FLAGS WITH INFORMATION ABOUT OPENING
C                  OF NODES
C
C      FILE      -CONTAINING INFORMATION BOUT THE RESULT FILE
C
C      CHARACTERISTICS
C      =====
C
C      SUBROUTINES REQUIRED: WAIT, RESU, OPCOM, RFIL
C      -----
C      FROM THE SYSTEM LIBRARY: IDELET, IFREE, PURGE, IBETC, IENTER
C
C      SIZE:
C      -----
C      PDP11/03: 1042

```

```

C
C      PROGRAM:
C      =====

```

```

C      SUBROUTINE CRLM(Y, SI, CO, T, SYS, SYC, ITW, ITE, IAU, LUN, IFST, IFIL)

```

```

C      DIMENSION FN(2)

```

```

C      LOGICAL*1 IVAL

```

```

C      COMMON/BUFC/ RES(200), RES1(10)
C      COMMON/CRL/ MPA, MCO, NYF, MR, W, SG, MPE, AM, ARG1
C      COMMON/NUSA/ MP2, MSP
C      COMMON/FLA/ IFL1, ICN, NBN, ICO, IFL2
C      COMMON/PAV/ I1, I2, I3, I4, I5, I6, I7, I8, IC1, IR1,
C      *NB1, IR2, NB2, IC2, I9, I10
C      COMMON/WRK/ IW1(50), IW2(50), IW3(50), IW4(50)
C      COMMON/FFLAG/ IFAN, IF6, IB6
C      COMMON/FILE/ IC, NB, IR

```

```

C      DATA PI/3.1415926/
C      DATA FN/6RDX1WOR, 6RK DAT/

```

```

C      DT=FLOAT(MSP)*0.02

```

```

C
C      SI2=SI
C      CO1=CO
C      MPA=MPA-1
C      PE=FLOAT(MPE)
C      TW=FLOAT(ITW)

```



```

C      INTEGRATION TIME.
C
C      TI=PE*2.*PI/W
C      TP=PI/W
C      T=T+DT
C
C      TIME WINDOW.
C
C      WI=1.-TW*COS(T*W/PE)
C      SI=SI*DT*WI/AM
C      CO=CO*DT*WI
C
C      TREND ELIMINATION?
C
C      IF(ITE.EQ.0) GO TO 101
C      TM=AMOD(T,3*TP)
C      TR=0.75
C      IF(TM.GE.TP.AND.TM.LE.2*TP) TR=1.5
C      SI=SI*TR
C
C      THE CORRELATION CHANNELS.
C
C      101  SYS=SYS+Y*SI
C          SYC=SYC+Y*CO
C
C      OVERFLOW IN THE CHANNELS?
C
C      IF(SYS.LT.10000.) GO TO 1
C      WRITE(7,100) Y,SI2
C      100  FORMAT(1X,'SYS OVERFLOW,Y=' ,E10.2,' SIN=' ,E10.2)
C          1  IF(SYC.LT.10000.) GO TO 2
C          WRITE(7,110) Y,CO1
C          110  FORMAT(1X,'SYC OVERFLOW,Y=' ,E10.2,' COS=' ,E10.2)
C
C      LAST SAMPLE?
C
C      2  IF(MPA.NE.0) RETURN
C
C      CALCULATE GAIN AND ARGUMENT OF THE ANALYSED SYSTEM.
C
C      GAIN=SQRT(4*(SYS**2+SYC**2))/(TI*AM)
C      ARG=ATAN2(SYC,SYS)
C      MR=MR+1
C
C      GAIN AND ARG ARE STORED IN THE RESULTVECTOR.
C
C      RES(4*MCO+2)=GAIN
C
C      ARE THERE ANY DISCONTINUITIES IN THE PHASE.
C
C      ARG=ARG*180./3.141596536
C      ARG0=0.
C      IF(ABS(ARG1).LT..00001) GO TO 500
C      IF(ARG-ARG1.GT.180.) ARG0=-360.
C      IF(ARG-ARG1.LT.-180.) ARG0=360.
C      ARG=ARG+ARG0
C      500  ARG1=ARG
C          RES(4*MCO+3)=ARG
C
C      WRITE(7,700) RES(4*MCO+1),ARG,GAIN
C      700  FORMAT(1X,' FREQUENCY=' ,E9.2,2X,' ARGUMENT=' ,F6.1,2X,' GAIN=' ,
C          *E9.2/)
C          SYS=0.
C          SYC=0.

```

```

IF(S6.LT.0.5) NYF=1
T=0.

C
C
C
DELETE THE FILE WORK.DAT.

C
C
C
IK=IGETC( )
CALL IDELET(IK, FN)
CALL IFREEC(IK)
CALL PURGE(IC0)
CALL PURGE(ICN)
CALL IFREEC(IC0)
CALL IFREEC(ICN)

C
C
C
CREATE A NEW FILE NAMED WORK.DAT.

C
C
C
ICN=IGETC( )
IC0=IGETC( )
NBN=IENTER(ICN, FN, 0)
IF(NBN.GT.0) GO TO 3
WRITE(7, 120)
120  FORMAT(1X, 'ENTER FAILURE IN CRLM' )
RETURN

C
C
C
SET THE POINTERS WITH INFORMATION ABOUT CALCULATING AND COLLECTING
BLOCKS.

C
C
C
3  NBN=NBN-1
   NB1=NBN
   NB2=NBN
   IC1=ICN
   IC2=ICN
   IR1=-1

C
C
C
AUTOMATIC OR MANUAL RUN?

C
C
C
IF(IAU.EQ.0.OR.S6.GT.0.5) GO TO 4
CALL WAIT

C
C
C
LAST FREQUENCY?

C
C
C
4  IF(S6.LT.0.5) RETURN

C
C
C
STOP COLLECTING.

C
C
C
IW1(1)=1
CALL ISDAT(IW1, 1)

C
C
C
FIRST OR SECOND RUN?

C
C
C
IF(IFST.EQ.0) GO TO 5
IF(IFIL.EQ.0) GO TO 17
CALL RFIL(MR, IFST)
CALL CLOSEC(IC)
GO TO 17

C
C
C
5  WRITE(7, 130)
130  FORMAT(1X, *, 'DO YOU WANT TO SAVE THE RESULT ON A FILE?' )
     READ(5, 210) IVAL
     IF(IVAL.EQ.89.OR.IVAL.EQ.78) GO TO 6
     WRITE(7, 220)
     GO TO 5

C
C
C
WAS A FILE WANTED?

```

```

6      IF(IVAL.EQ.78) GO TO 16
      IFIL=1
      CALL RFIL(MR,IFST)

C
C      WRITE THE RESULTHEAD ON SPECIFIED DEVICE.
C
16     IR=IR+1
      IF(MR.GT.29) IR=IR+1
      WRITE(LUN,170) DT
170    FORMAT(1X,'MAIN SAMPLE PERIOD= ',F5.3,' SEC' /)
      CALL RESU(LUN,RES1)

C
C      WRITE THE RESULT ON SPECIFIED DEVICE.
C
17     WRITE(LUN,180)
      DO 18 I=1,MR
      KR=4*(I-1)
18     WRITE(LUN,190) (RES(KR+K),K=1,4)
180    FORMAT(1X,'FREQUENCY',7X,'GAIN',6X,'ARGUMENT',5X,'NUMBER OF' /
      *27X,'(DEG)',7X,'MEASURE PERIODS' //)
190    FORMAT(1X,E9.2,2X,E9.2,2X,F9.1,14X,F4.1)
      CALL CLOSE(LUN)

C
C      IF(IFST.EQ.1) RETURN
C
19     WRITE(7,200)
200    FORMAT(1X,'YOU HAVE NOW THE POSSIBILITY TO DO A NEW MEASUREMENT' /
      *1X,'WITH SHORTER INTERVALS BETWEEN THE FREQUENCIES.' /
      *1X,?, 'DO YOU WANT TO DO THIS?')
210    READ(7,210) IVAL
      FORMAT(A1)
      IF(IVAL.EQ.89.OR,IVAL.EQ.78) GO TO 20
      WRITE(7,220)
220    FORMAT(1X,'?ILL CMD?')
      GO TO 19
20     IF(IVAL.EQ.78) GO TO 23
      IFST=1
      WRITE(7,230)
230    FORMAT(1X,'PLEASE DEFINE A NEW FREQUENCY INTERVAL.')
```

C

```

      DO 21 I=1,10
21     IW4(I)=0

C
C      ACTI FRCL.
C
      IW4(1)=2
      IW4(2)=2
      IW4(5)=30
      IW4(6)=1
      IBG=1
      CALL OPCOM

C
C      IBG=0
      MR=MR*4
      DO 22 I=1,MR
22     RES(I)=0
      MR=0
      RETURN

C
C      CLOSE THE FILE.
C
23     IF(IFIL.EQ.1) CALL CLOSEC(IC)
      RETURN
      END
```

NAME: CRLMST

SUBTITLE: START-UP ROUTINE TO THE ROUTINE CRLM

LANGUAGE: FORTRAN IV

KEYWORDS: START-UP

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN

DATE: 26-APR-78

INSTITUTE:

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:

VERSION:

PURPOSE AND METHOD

THE ROUTINE:

1. ALLOCATES SPACE IN THE PARAMETERAREA
2. SETS THE ADDRESS PART IN THE NODEHEAD, ADDRESSES TO THE PARAMETERAREA
3. ASKS FOR AND SETS SOME PARAMETERVALUES

USAGE:

=====

PROGRAM TYPE: SUBROUTINE

ARGUMENTS:

CRLMST(LIST, IL, IPA, IW)

IL            -POINTER TO THE FIRST WORD IN THE ACTUAL NODE, (I)  
IPA           -VECTOR CONTAINING THE PARAMETERS IN THE BACKGROUND JOB, (I)  
IW            -WORKAREA

COMMON BLOCKS:

NUSA          -CONTAINING INFORMATION ABOUT THE FREQUENCIES  
BUFS          -CONTAINING THE RESULTVECTORS

CHARACTERISTICS:

=====

SUBROUTINES REQUIRED: RS, CV, LU

```
C
C
C      SIZE:
C
C      PDF11/03: 654
C
C-----
C
C
C      PROGRAM:
C      =====
C
C      SUBROUTINE CRLMST(LIST,IL,IPA,IW)
C
C      DIMENSION LIST(1),IPA(1),IW(1)
C
C      COMMON/NUSA/MP2,MSP
C      COMMON/BUFC/RES(200),RES1(10)
C
C      LOGICAL*1 I
C
C      ALLOCATE SPACE IN THE VECTOR IPA FOR 12 WORDS.
C
C      CALL RS(IL,12,LIST,IW)
C
C      ENOUGH FREE SPACE?
C
C      IF(IL.LT.0) RETURN
C
C      ITE=0
C      ITW=0
C      IAU=0
C      LUN=0
C
C      SET ADDRESSES IN THE NODE.
C
C      K1=LIST(IL+3)
C      LIST(IL+4)=K1+2
C      LIST(IL+5)=K1+4
C      LIST(IL+6)=K1+6
C      K5=LIST(IL+7)
C
C      DO 1 J=K1,K5
C 1      IPA(J)=0
C
C 10     WRITE(7,2)
C 2     FORMAT(1X,'IF YOU ANSWER YES ON THE FOLLOWING QUESTION' /
C          *1X,'YOU HAVE TO CHOOSE AN INTEGRATION TIME OF 3 PERIODS.' /
C          *1X,$,'DO YOU WANT TREND ELIMINATION?')
C      READ(5,3) I
C 3     FORMAT(A1)
C      IF(I.EQ.89.OR.I.EQ.78) GO TO 12
C      WRITE(7,11)
C 11    FORMAT(1X,'?ILL CMD?')
C      GO TO 10
C 12    IF(I.EQ.89) ITE=1
C      RES1(7)=FLOAT(ITE)
C
C 13    WRITE(7,4)
C 4     FORMAT(1X,$,'DO YOU WANT TIME WINDOW?')
C      READ(5,3) I
```

```
IF(I.EQ.89.OR.I.EQ.78) GO TO 14
WRITE(7,11)
GO TO 13
14 IF(I.EQ.89) ITW=1
RES1(8)=FLOAT(ITW)
C
15 WRITE(7,16)
16 FORMAT(1X,'YOU HAVE NOW THE POSSIBILITY TO DECIDE IF THE SYSTEM' /
*1X,' IS TO BE RUN AUTOMATICALLY OR NOT.' /
*1X,' IF NOT,THE RUNNING OF THE ANALYSIS IS INTERRUPTED BETWEEN' /
*1X,' TWO FREQUENCIES,AND YOU CAN DECIDE YOURSELF WHEN IT IS TIME' /
*1X,' FOR A NEW FREQUENCY.YOU CAN ALSO AT THAT OCCASION CHANGE THE' /
*1X,'AMPLITUDE OF THE OUTPUT SIGNAL OF THE COMPUTER.' //
*1X,%,'DO YOU WANT AUTOMATIC FREQUENCY ANALYSIS:' )
C
READ(5,3) I
IF(I.EQ.89.OR.I.EQ.78) GO TO 17
WRITE(7,11)
GO TO 15
17 IF(I.EQ.78) IAU=1
RES1(9)=FLOAT(IAU)
C
C STORE PARAMETERS IN PARAMETERAREA.
C
IPA(K5)=ITE
IPA(K1+6)=ITW
IPA(K5-1)=IAU
C
RES1(10)=MSP
C
WRITE(7,18)
18 FORMAT(1X,'ON WHICH DEVICE DO YOU WANT THE FINAL RESULT?' /
*1X,'(ANSWER TT OR LP ON THE FOLLOWING QUESTION!)' )
CALL LUX(LUN)
IPA(K5-2)=LUN
RETURN
END
```



```

C
1  WRITE(7,100)
100  FORMAT(1X,'GIVE THE NAME OF THE FILE YOU WANT TO READ,'
      *1X,'FOR EXAMPLE:DX1:MAT.DAT=#'/)

C
C \  PARSE A COMMAND STRING AND RETURN A FILE DESCRIPTION TO
C   THE PROGRAM.
C
      JJ=ICSI(IW,EXT,,,0)
      IF(JJ.EQ.0) GO TO 2
      WRITE(7,200)
200   FORMAT(1X,'?ILL CMD?')
      GO TO 1

C
      LEGAL DEVICE?
C
C
2   DO 3 I=1,3
      IF(IDEV(I).EQ.IW(1)) GO TO 4
3   CONTINUE
      WRITE(7,300)
300  FORMAT(1X,'?ILL DEV?')
      GO TO 1

C
      ALLOCATE AN RT-11 CHANNEL AND MARK IT IN USE FOR THE I/O SYSTEM.
C
C
4   IK=IGETC( )

C
      OPEN THE FILE.
C
      NK=LOOKUP(IK,IW)
      IF(NK.EQ.0) GO TO 15
      IF(NK.EQ.-2) WRITE(7,450)
      IF(NK.EQ.-1) WRITE(7,500)
450  FORMAT(1X,'FILE NOT FOUND')
500  FORMAT(1X,'AN ERROR OCCURED DURING OPENING THE FILE.')
```

STOP

```

C
15  WRITE(7,400)
400  FORMAT(1X,'ON WHICH DEVICE (TT OR LP) DO YOU WANT THE RESULT?'/)
      CALL LU(LUN)

C
      NK=NK-1
      MR1=29
      IFT=1
      IST=1

C
      BEGIN TO WRITE ON A NEW SIDE.
C
      WRITE(LUN,550)
550  FORMAT(1H1)

C
      READ A BLOCK ON THE FILE.
C
5   ICODE=IREADW(256,IO,IR,IK)
      IF(ICODE+1) 14,14,6

C
6   IF(IFT.NE.1) GO TO 8

C
      CONVERT TWO CONTINUOUS INTEGER*2 NUMBER TO A REAL*4 NUMBER.
C
      DO 7 I=1,10
      II=2*(I-1)+1
7   CALL CV(RES1(I),IO(II))

C

```



```

DT=RES1(10)*0.02
WRITE(LUN,600) DT
600  FORMAT(1X,'MAIN SAMPLE PERIOD WAS ',F5.3,'SEC'//)
C
C  WRITE THE RESULTHEAD.
C
C  CALL RES0(LUN,RES1)
C
C  NUMBER OF RESULTS.
C
C  MR=IO(21)
NR=MR
C
C  MORE THAN ONE BLOCK TO READ?
C
C  IF(MR.GT.29) NR=29
NST=25
MR1=29
C
C  8  GO TO (9,10,20,10) IST
C
C  20  MR=IO(1)
NR=MR
IF(MR.GT.31) NR=31
WRITE(LUN,640)
640  FORMAT(1X,'THE RESULT OF A SECOND RUN WITH EXTRA FREQUENCIES:'//)
C
C  9  WRITE(LUN,650)
650  FORMAT(1X,'FREQUENCY',7X,'GAIN',6X,'ARGUMENT',5X,'NUMBER OF'//
*27X,'(DEG)',7X,'MEASURE PERIODS'//)
C
C  CONVERT TWO CONTINUOUS INTEGER*2 NUMBER TO A REAL*4 NUMBER.
C
C  10  NR1=4*NR
DO 11 I=1,NR1
II=2*(I-1)+NST
01  CALL CV(RES(I),IO(II))
C
C  WRITE THE RESULT.
C
C  DO 12 I=1,NR
II=4*(I-1)
MP=IFIX(RES(II+4))
12  WRITE(LUN,700) RES(II+1),RES(II+2),RES(II+3),MP
700  FORMAT(1X,E9.2,4X,E9.2,4X,F9.1,11X,I2)
C
C  IFT=IFT+1
0  IST=IST+1
GO TO (16,16,13,16,13) IST
C
C  MORE RESULTS TO READ?
C
C  16  IF(MR.LE.MR1) GO TO 13
NR=MR-NR
NST=1
IR=IR+1
GO TO 5
C
C  ARE THERE RESULTS FROM A SECOND RUN TO READ?
C
C  13  IR=IR+1
IF(IR.GT.NK) STOP
C
C  IST=IST+1

```

```
MR1=31
NST=5
WRITE(LUN,710)
FORMAT(1X,///)
GO TO 5

C
14 WRITE(7,750)
750 FORMAT(1X,'FATAL READ')
STOP
END
```

NAME: FG  
-----

SUBTITLE: MAINPROGRAM IN FOREGROUND JOB  
-----

LANGUAGE: FORTRAN + RT-11-FORTRAN  
-----

IMPLEMENTOR: LEIF RADING, TYKE PAULSSON  
-----

DATE: 7-SEP-77  
-----

REVISED BY: CHRISTER LARSSON, HARRIET LINDGREN  
-----

DATE: 28-APR-78  
-----

INSTITUTE:  
-----

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

PURPOSE AND METHOD  
-----

FG STARTS FOREGROUND JOB WITH INFORMATION RECEIVED FROM  
BACKGROUND JOB.  
THE PROGRAM ORGANIZES COLLECTING OF MEASUREMENT DATA, RUNNING  
OF CONTROLLING PROGRAMS AND SENDING OF DATA TO BACKGROUND JOB.  
FURTHER IT TAKES CARE OF INFORMATION AND COMMANDS FROM BACK-  
GROUND JOB, EXECUTES THE COMMAND AND SENDS POSSIBLE REQUESTED  
INFORMATION.

USAGE  
=====

COMMON BLOCKS:  
-----

FL -

NEW COMMON BLOCKS:  
-----

FORG -CONTAINING THE NODE HEAD,  
COLL -CONTAINING PARAMETERS WHICH CHECK IF THE MAIN SAMPLE  
PERIOD HAS BEEN TO LONG

CHRACTERISTICS  
=====

SIZE:  
-----

PDP11/03: 1776 WORDS

\*\*\*\*\*  
\*  
\* FOREGROUND PROGRAM \*  
\*  
\*\*\*\*\*

```

C      EXTERNAL FLAG6,FLAGR
C      DIMENSION IV(60),IB(60),IT1(2),IT2(2)
C      *,LINK1(4),LINK2(4),IPF(400)
C      COMMON /FL/I1,KFL,ISC
C      COMMON /FORG/LF(240)
C      COMMON/COLL/ JT1,JT2,LSI
C      DATA IPF/400*0/

C
C      SET ALL ZERO LEVELS EQUAL 0.0 AND SLOPES EQUAL 1.0
C
C      DO 5 I=29,75,4
5      IPF(I)=16512

C
C      SET ALL NODES INACTIVE
C
C      DO 10 I=1,240,12
10     LF(I)=-1
        LF(I+1)=0

C
C      ISC=0
C      LSI=0
C      KFL=0
C      JT1=0
C      JT2=0
C      JT3=0
C      ADD AVAILABLE ELEMENTS TO THE QUEUE
C
C      CALL IQSET(6)
C
C      RECEIVE DATA FROM BACKGROUND JOB
C
C      CALL IRCVDW(IB,59)
C
C      SETS INITIALLY THAT ONE MESSAGE FROM BACKGROUND JOB ARE
C      WAITING TO BE EXECUTED.
C
C      I1=0
C
C      INCREMENT SUSPENSION COUNTER TWICE.
C
C      CALL RESUME
C      CALL RESUME
C
C      SET INITIALLY NO REGULATION AND NO DATA COLLECTION.
C
C      I2=1
C
C      WHAT TO DO?
C
C      15  GO TO(20,21,22,23,24,25,26,29,31,33,35,36,39,38)IB(2)
C      STOP REGULATION.
C
C      20  IF(I2.EQ.1.OR.I2.EQ.3) GO TO 50
C      I2=I2-1
C      GO TO 50
C
C      START REGULATION.
C
C      21  IF(I2.EQ.4.OR.I2.EQ.2) GO TO 50
C      I2=I2+1
C      GO TO 50
C
C      STOP COLLECTION.

```

```
C
22  IF(I2.EQ.1.OR.I2.EQ.2) GO TO 50
    I2=I2-2
    GO TO 50
C
C    START COLLECTION.
C
23  IF(I2.EQ.3.OR.I2.EQ.4) GO TO 50
    I2=I2+2
    GO TO 50
C
C    SEND PARAMETERS FROM VECTOR LF TO BACKGROUND JOB.
C
24  J1=IB(3)
C
C    IB(4) NOT ALLOWED TO BE EQUAL NS.
C
    IF(IB(4).EQ.NS) IB(4)=IB(4)+1
    CALL ISDAT(LF(J1),IB(4))
    GO TO 50
C
C    SEND PARAMETERS FROM VECTOR IPF TO BACKGROUND JOB.
C
25  J1=IB(3)
C
C    IB(4) NOT ALLOWED TO BE EQUAL NS.
C
    IF(IB(4).EQ.NS) IB(4)=IB(4)+1
    CALL ISDAT(IPF(J1),IB(4))
    GO TO 50
C
C    CHANGE PARAMETERS IN VECTOR LF.
C
26  J1=IB(3)
    J2=IB(4)
    I=4
    DO 27 J=J1,J2
    I=I+1
    IF(I.LT.13.OR.IB(I).GE.0) GO TO 27
    JJ=-IB(I)*12-7
    IB(I)=LF(JJ)-2
27  LF(J)=IB(I)
    GO TO 50
C
C    CHANGE PARAMETERS IN VECTOR IPF.
C
29  J1=IB(3)
    J2=IB(4)
    I=4
    DO 30 J=J1,J2
    I=I+1
30  IPF(J)=IB(I)
    GO TO 50
C
C    ACTIVATE NODE(S).
C
31  J1=IB(1)+1
    DO 32 I=3,J1
    J2=(IB(I)-1)*12+1
    LF(J2)=0
32  LF(J2+1)=IABS(LF(J2+1))
    GO TO 50
C
C    DEACTIVATE NODE(S).
```

```

C
33  J1=IB(1)+1
    DO 34 I=3, J1
    J2=(IB(I)-1)*12+1
    LF(J2)=-1
34  LF(J2+1)=0
    GO TO 50

C
C    AVAILABLE SPACE IN VECTOR IPF?
C
35  J1=IB(3)
    CALL RS1(IL, J1, LF, IV)

C
C    SEND THE RESULT OF THE SERCH TO THE BACKGROUND JOB.
C
    IV(1)=IL
    CALL ISDAT(IV, 1)
    GO TO 50

C
C    SEND INFORMATION ABOUT ACTIVE NODES TO THE BACKGROUND JOB.
C
36  J2=1
    DO 37 J1=2, 240, 12
    IF(LF(J1).EQ.0) GO TO 37
    J2=J2+1
    IV(J2)=LF(J1+1)
    J2=J2+1
    IV(J2)=J1/12+1
    J2=J2+1
    IV(J2)=LF(J1-1)
    J2=J2+1
    IV(J2)=LF(J1)
37  CONTINUE
    IV(1)=J2

C
C    J2 NOT ALLOWED TO BE EQUAL NS.
C
    IF(J2.EQ.NS) J2=J2+1
    CALL ISDAT(IV, J2)
    GO TO 50

C
C    CONTROL OF THE MAIN SAMPLE PERIOD.
C
38  IV(1)=LSI
    CALL ISDAT(IV, 1)
    LSI=0
    GO TO 50

C
C    RECEIVED INFORMATION ABOUT, NUMBER OF INPUT SIGNALS,
C    OUTPUT SIGNALS AND MAIN SAMPLE PERIOD.
C
39  IT1(1)=IB(3)
    IT1(2)=IB(4)
    ISAMPL=IB(4)
    NI=IB(5)
    NU=IB(6)
    NS=(NI+NU+1)*2

C
C    CONVERT MAIN SAMPLE PERIOD FROM 2-WORD INTERNAL
C    FORMAT TIME TO HOURS, MINUTES, SECONDS AND TICKS.
C
    CALL CVTTIM(IT1, IH, IM, IS, IT)

C
C    CONVERT A 2-WORD INTERNAL TIME FORMAT TO INTEGER*4 FORMAT

```

```

C      AND THEN CONVERT IT TO REAL*4 VALUE.
C
C      CALL JJCVT(IT1)
C      TSM=AJFLT(IT1)*0.02
C
C
C
50    I1=I1-1
C
C      QUEUED A RECEIVE DATA REQUEST AND CONTINUE EXECUTION OF
C      ISSUING JOB UNTIL A NEW MESSAGE IS RECEIVED, THEN ENTER
C      THE SUBROUTINE FLAGR.
C
C      CALL IRCVDF(IB,59,LINK1,FLAGR)
C
C      DECREMENT SUSPENSION COUNTER TWICE.
C
60    CALL SUSPND
C      CALL SUSPND
C
C      WHEN FREQUENCY ANALYSIS A FLAG IS SET THAT DATA ONLY SHOULD
C      BE SEND DURING THE MEASUREPERIODS
C
C      IF(ISC.EQ.1) I2=4
C      IF(ISC.EQ.2) I2=2
C      IF(ISC.EQ.1.OR.ISC.EQ.2) I1=-1
C      ISC=0
C
C
C      MESSAGE FROM BACKGROUND?
C      IF(I1.GE.0) GO TO 15
C
C      SCHEDULE THE SUBROUTINE FLAG TO BE RUN AFTER
C      THE SAMPLE PERIOD HAS ELAPSED
C
C      CALL ITIMER(IH,IM,IS,IT,LINK2,LD,FLAG)
C
C      GET THE CURRENT TIME OF DAY
C
C      CALL GTIM(IT2)
C
C      CHECK IF THE SAMPLE PERIOD HAS BEEN LONGER THAN IT SHOULD.
C
C      IPF(1)=IT2(1)
C      IPF(2)=IT2(2)
C      IF(JT1.NE.0.OR.JT2.NE.0) GO TO 370
C      JT1=IT2(2)
C      GO TO 376
370   JT2=IT2(2)
C      IF(JT1.GE.0.AND.JT2.GT.0.OR.JT1.LT.0.AND.JT2.GE.0) GO TO 371
C      IF(JT1.LT.0.AND.JT2.LT.0) GO TO 371
C      IF(JT1.GT.0.AND.JT2.LT.0) GO TO 372
C      IF(JT1.GT.0.AND.JT2.EQ.0) GO TO 373
C      IF(JT1.EQ.0.AND.JT2.LT.0) GO TO 374
371   JT3=JT2-JT1
C      GO TO 375
372   JT3=16000-JT1+16768+JT2+16000+16768
C      GO TO 375
373   JT3=16768-JT1+16000
C      GO TO 375
374   JT3=16768+JT2+16000
375   IF(JT3.GT.ISAMPL) LSI=LSI+1
C      IF(LSI.GT.32700) LSI=32700
C      JT1=JT2

```

```
C
C      READ INPUT MEASUREMENT AND CONVERT TO PHYSICAL VALUE
C
376  C      CALL LOGIN(IPF,NI)
C
      GO TO(350,90,330,90),I2
C
C      RUN ACTIVATED PROGRAMS.
C
90   C      DO 320 IP=1,240,12
      IF(LF(IP)) 320,100,310
100  C      TS=TSM*FLOAT(LF(IP+1))
      K1=LF(IP+3)
      K2=LF(IP+4)
      K3=LF(IP+5)
      K4=LF(IP+6)
      K5=LF(IP+7)
      K6=LF(IP+8)
      K7=LF(IP+9)
      K8=LF(IP+10)
      K9=LF(IP+11)
      GO TO(101,102,103,104,105,106,107,108,109,110,
      *111,112,113,114,115,116,117,118,119,120) LF(IP+2)
C
101  C      CALL PIDR(IPF(K2-2),IPF(K6),IPF(K7),IPF(K1+8),IPF(K1+10),
      *IPF(K1+12),IPF(K1),IPF(K1+2),IPF(K1+4),IPF(K1+6),TS)
      GO TO 290
C
102  C      CALL TIME(IPF(K1),IPF(K1+2),IPF(K1+4))
      GO TO 290
C
103  C      CALL STEP(IPF(K1+2),IPF(K1),IPF(K2-2))
      GO TO 290
C
104  C      CALL ADDS(IPF(K6),IPF(K7),IPF(K8),IPF(K1),IPF(K1+2),
      *IPF(K1+4),IPF(K2-2))
      GO TO 290
C
105  C      CALL RAMP(IPF(K1+2),IPF(K1),TS,IPF(K2-2))
      GO TO 290
C
106  C      CALL SINU(IPF(K1+6),IPF(K1),IPF(K1+2),TS,IPF(K1+4),IPF(K2-2))
      GO TO 290
C
107  C      CALL PULS(IPF(K1+6),IPF(K1),IPF(K1+2),TS,IPF(K1+4),
      *IPF(K2-2))
      GO TO 290
C
108  C      CONTINUE
      GO TO 290
C
109  C      CONTINUE
      GO TO 290
C
110  C      CONTINUE
      GO TO 290
C
111  C      CALL PRB(IPF(K2),IPF(K3),IPF(K2-2),IPF(K5),IPF(K1))
      GO TO 290
C
112  C      CALL NODI(IPF(K1),IPF(K1+2),IPF(K2-2),IPF(K3),IPF(K3+1))
      GO TO 290
113  C      CONTINUE
      GO TO 290
```



```
C
114  CONTINUE
      GO TO 290
C
115  CONTINUE
      GO TO 290
C
116  CONTINUE
      GO TO 290
C
117  CONTINUE
      GO TO 290
C
118  CONTINUE
      GO TO 290
C
119  CONTINUE
      GO TO 290
C
120  CONTINUE
C
290  IF(K9.NE.0) IPF(K9)=IPF(K2-2)
C
300  LF(IP)=LF(IP+1)
310  LF(IP)=LF(IP)-1
320  CONTINUE
C
C
C      SEND OUTPUT SIGNALS TO D/A-CONVERTER.
C
C      CALL LOGOUT(IPF,NU)
330  GO TO(350,350,340,340),I2
C
C      QUEUED A SEND DATA REQUEST AND CONTINUE EXECUTION OF ISSURING
C      JOB UNTIL THE BACKGROUND JOB HAS RECEIVE THE MESSAGE.
C
340  J1=(NI-8)*2
      J2=NU*2+18
      DO 339 I=19,J2
339  IPF(I+J1)=IPF(I)
      IF(ISDATF(IPF,NS,LINK2,FLAG).EQ.0) GO TO 360
      I2=I2-2
C
C      INCREMENT SUSPENSION COUNTER.
C
350  CALL RESUME
C
C      IF IT IS NOT FREQUENCY ANALYSIS,GIVE BACKGROUND JOB SOME TIME.
C
360  IF(KFL.NE.1) CALL ISLEEP(0,0,0,2)
      GO TO 60
      STOP
      END
```



```

C      B6          -CONTAINING LIST AND PARAMETER ARRAYS FOR B6 JOB
C      BUFC       -CONTAINING THE RESULT VECTORS
C      CRL        -CONTAINING PARAMETERS AND FLAGS WITH INFORMATION C29
C                  ABOUT FREQUENCY ANALYSIS
C      NUSA       -CONTAINING THE ARRAY WITH THE STANDARD
C                  FREQUENCIES
C
C
C      CHARACTERISTICS
C      -----
C      SUBROUTINES REQUIRED:
C      -----
C      OPEN, OPCOM
C
C      SIZE:
C      -----
C      PDP11/03: 594
C
C-----
C
C      PROGRAM:
C      =====
C
C      SUBROUTINE FREQ(F1, F2, MPI, TT, MP, NR, NEX, N2, M2)
C
C      LOGICAL*1 II
C
C      COMMON/IPAP/ LI, II
C      COMMON/FFLAG/ IFAN, IFG, IBG
C      COMMON/RP/ KA, IWP(10), FIW(10)
C      COMMON/WRK/ IW1(50), IW2(50), IW3(50), IW4(50)
C      COMMON/FAN/IPER, IWI(10), IWO(10)
C      COMMON/CHECK/ICH
C      COMMON/B6/LIST(120)
C      COMMON/BUFC/RES(200)
C      COMMON/CRL/MPA, MCO, NYF, MR, W, SG, MPE, AM
C      COMMON/NUSA/MP2, MSP, IHR, A(50)
C
C      DIMENSION F(2), T(2), N1(2), M1(2)
C
C      SG=-1.
C      SP=FLOAT(MSP)
C
C      FIRST TIME?
C
C      IF(II.GT.0) GO TO 8
C      MCO=-1
C      F(1)=F1
C      F(2)=F2
C
C      THE FREQUENCY LIMITS ARE FITTED TO STANDARD FREQUENCIES
C
C      DO 1 I=1,2
C      T(I)=50./SP/F(I)
C      N1(I)=0
C      DO 7 J=1,2
C      IF(T(J).LE.A(MP+1)) GO TO 3
C      IF(T(J).LE.A(1)) GO TO 5
C      T(J)=T(J)/10.
C      N1(J)=N1(J)+1
C      GO TO 2
C      MP1=MP2-MP

```

```

DO 4 K=1,MP1
  KX=MP2-K+1
  IF(A(KX).GE.T(J)) GO TO 7
4  CONTINUE
5  DO 6 K=1,MP
  KX=MP-K+1
  IF(A(KX).GE.T(J)) GO TO 7
6  CONTINUE
7  M1(J)=KX
  N2=N1(2)
  M2=M1(2)
  NR=M1(1)
  NEX=N1(1)
  GO TO 29

C
C  NEXT FREQUENCY.
C
8  NR=NR+1
  IF(NR.LE.MP) GO TO 28
  IF(NEX.EQ.0) GO TO 28
  NR=1
  NEX=NEX-1

C
C  LAST FREQUENCY?
C
28 IF(NEX.GT.N2) GO TO 29
  IF(NR.LT.M2) GO TO 29
  NEX=N2
  NR=M2
  WRITE(7,100)
100 FORMAT(1X,'LAST FREQUENCY')
  SG=1.

C
C  DEACTIVATE FREQ1
C
  LIST(LI-2)=-1

C
C  CALCULATE THE FREQUENCY.
C
29 F1=50./SP/(A(NR)*10.**NEX)
  W=F1*6.28318531
  MCO=MCO+1

C
C  THE FREQUENCY IS STORED IN VECTOR RES
C
  RES(4*MCO+1)=F1
  IF(IHR.EQ.0) RES(4*MCO+1)=6.28318531*RES(4*MCO+1)

C
C  NUMBER OF MEASURE PERIODS ARE SET
C
  MPE=MPI
  IF(MPE.GT.0) GO TO 30
  MPE=1
  IF(F1.GE.0.1) MPE=3
  IF(F1.GE.0.2) MPE=5
  IF(F1.GE.0.5) MPE=10
  IF(F1.GE.1.) MPE=20

C
C  THE NUMBER OF MEASURE PERIODS IS STORED IN VECTOR RES.
C
30 RES(4*MCO+4)=FLOAT(MPE)

C
C  THE NUMBER OF MEASURE PERIODS IS TRANSLATED TO A NUMBER OF
C  MAIN SAMPLE PERIODS.

```

```

C      IA=IFIX(A(NR))
      MPS=IA*MPE*10**NEX
      MPA=MPS

C
C      THE TRANSIENT TIME IS TRANSLATED TO A NUMBER OF MAIN SAMPLE PERIODS.
C
      TT1=50./SP*TT
      NT=IFIX(TT1)
      L=0
31     MTS=IA*L*10**NEX
      IF(MTS.GE.NT) GO TO 32
      L=L+1
      GO TO 31

C
32     DO 33 I=1,50
33     IW4(I)=0

C
C      OPEN SINU1 AND SET THE FREQUENCY.
C
      IW4(1)=2
      IW4(2)=1
      IW4(5)=46
      IW4(6)=1
      IFAN=1
      IPER=1
      IWI(1)=0
      IWO(1)=1
      IWO(2)=1
      IWP(2)=1
      IWP(1)=0
      KA=2
      FIW(2)=F1
      CALL OPEN

C
C      OPEN SINU2 AND SET THE FREQUENCY AND THE PHASE.
C
      IW4(5)=46
      IW4(6)=2
      IWO(2)=2
34     DO 34 I=1,3
      IWP(I)=1
      KA=3
      FIW(1)=1.
      FIW(2)=F1
      FIW(3)=1.57079633
      CALL OPEN

C
C      OPEN TIME3
C
      IW4(5)=42
      IW4(6)=3
      IPER=MTS
      IF(MTS.EQ.0) IPER=1
      IWO(1)=0
      IWO(2)=0
      IWP(4)=1
      KA=4
      FIW(1)=FLOAT(MTS)
      FIW(2)=FLOAT(MPS)
      FIW(3)=-1
      CALL OPEN

C
C      ACTI SINU1,SINU2,TIME3

```

```
C  
IFG=1  
IW4( 1 )=4  
IW4( 2 )=2  
IW4( 5 )=46  
IW4( 6 )=1  
IW4( 8 )=46  
IW4( 9 )=2  
IW4( 11 )=42  
IW4( 12 )=3  
CALL OPCOM
```

```
C  
C  
C  
RESET THE FLAGS.
```

```
IFAN=0  
IFG=0  
ICH=0  
NYF=0  
RETURN  
END
```



C CHARACTERISTICS:

C =====

C SUBROUTINES REQUIRED: FREQ,OPCOM,OPEN

C -----

C SIZE:

C -----

C PDP11/03: 487

C PROGRAM:

C =====

C SUBROUTINE FREQAN

C LOGICAL\*1 II  
C LOGICAL\*1 IVAL

C COMMON/BUFC/ RES(200),RES1(10)  
C COMMON/BG/LIST(120),IPA(512)  
C COMMON/FRAN/IPER,IWI(10),IWO(10)  
C COMMON/FFLAG/ IFAN,IFG,IBG  
C COMMON/RP/K,IWP(10),FIW(10)  
C COMMON/WRK/IW1(50),IW2(50),IW3(50),IW4(50)  
C COMMON/IFAP/LI,II  
C COMMON/CRL/ MPA,MCO,NYF,MR,W,SG,MPE,AM

C IFG=0

C IBG=0

10 DO 20 I=1,50

20 IW4(I)=0

C WRITE(7,100)

100 FORMAT(1X,'PLEASE DEFINE ON WHICH INPUTS YOU HAVE THE DIFFERENT'  
\*1X,'SIGNALS.THE FIRST SIGNAL IS THE OUTPUT SIGNAL FROM THE '  
\*1X,'ANALYSED SYSTEM.THE SECOND SIGNAL IS THE INPUT SINUS SIGNAL'  
\*1X,'TO THIS SYSTEM.THE THIRD SIGNAL IS THE COSINUS SIGNAL.'  
\*1X,'THE ANSWER SHALL IN GENERAL BE: INPU1,INPU2,INPU3')

C ACTI CRLM1

C IW4(1)=2

C IW4(2)=2

C IW4(5)=29

C IW4(6)=1

C IBG=1

C CALL OPCOM

C ACTI FREQ1

C IW4(5)=28

C IW4(6)=1

C CALL OPCOM

C RESET THE FLAG.

C IBG=0

C



```

WRITE(7,200)
200  FORMAT(1X,%, 'AMPLITUDE OF THE OUTPUT SIGNAL:')
READ(5,300) AM
300  FORMAT(F10.0)
C
C   WRITE THE RESULTHEAD.
C
CALL RESU(7,RES1)
WRITE(7,350) AM
350  FORMAT(1X, 'AMPLITUDE OF THE OUTPUT SIGNAL:',F6.2)
C
30  WRITE(7,400)
400 @ FORMAT(1X,%, 'DO YOU WANT TO CHANGE ANY PARAMETERS?')
READ(5,500) IVAL
500  FORMAT(A1)
IF(IVAL.EQ.89.OR.IVAL.EQ.78) GO TO 40
WRITE(7,600)
600  FORMAT(1X, ' ?ILL CMD?')
GO TO 30
40  IF(IVAL.EQ.78) GO TO 50
C
C   STOP CRLM1,FREQ1
C
IBG=1
IW4(1)=2
IW4(2)=3
IW4(5)=29
IW4(6)=1
CALL OPCOM
IW4(5)=28
CALL OPCOM
C
C   RESET THE FLAG.
C
IBG=0
GO TO 10
C
C   SEARCH FOR THE NODE FREQ1.
C
50  DO 60 I=3,120,12
60  IF(LIST(I).EQ.8001) LI=I
K1=LIST(LI+1)
K5=LIST(LI+5)
C
C   FIRST FREQUENCY.
C
@II=0
CALL FREQ(IPA(K1), IPA(K1+2), IPA(K5-4), IPA(K1+4), IPA(K5-3),
*IPA(K5-2), IPA(K5-1), IPA(K5), IPA(K1+6))
C
C   CONVERT TWO CONTINUOUS INTEGER*2 NUMBERS TO A REAL*4 NUMBER.
C
CALL CV(F1, IPA(K1))
C
II=1
C
C   OPEN SINU1 AND SET THE AMPLITUDE.
C
IW4(1)=2
IW4(2)=1
IW4(5)=46
IW4(6)=1
IFAN=1

```

```
IPER=1  
IWI( 1 )=0  
IWO( 1 )=1  
IWO( 2 )=1  
K=1  
IWP( 1 )=1  
FIW( 1 )=AM  
CALL OPEN
```

```
C  
C  
C
```

```
RESET THE FLAG.
```

```
IFAN=0  
RETURN  
END
```

NAME: FREQST  
 -----

SUBTITLE: START-UP ROUTINE TO THE ROUTINE FREQ  
 -----

LANGUAGE: FORTRAN IV  
 -----

KEYWORDS: START UP  
 -----

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN      DATE: 25-APR-78  
 -----

INSTITUTE:  
 -----

DEPARTMENT OF AUTOMATIC CONTROL  
 LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:  
 -----

VERSION:  
 -----

PURPOSE AND METHOD  
 =====

THE ROUTINE:

1. ALLOCATES SPACE IN THE PARAMETERAREA.
2. SETS THE ADDRESS-PART IN THE NODEHEAD, ADDRESSES TO THE PARAMETERAREA.
3. ASKS FOR AND SETS SOME PARAMETERVALUES.

USAGE:  
 =====

PROGRAM TYPE: SUBROUTINE  
 -----

ARGUMENTS:  
 -----

FREQST(LIST, IL, IPA, IW)

IL            -POINTER TO THE FIRST WORD IN THE ACTUAL  
                   NODE.(I)  
 IPA            -VECTOR CONTAINING THE PARAMETERS IN BACK-  
                   GROUND JOB.(I)  
 IW            -WORKAREA

COMMON BLOCKS:  
 -----

BUFC            -CONTAINING THE RESULTVECTORS.  
 NUSA            -CONTAINING INFORMATION ABOUT THE FREQUENCIES.  
 FRC            -CONTAINING INFORMATION ABOUT TRANSIENT TIME

C  
C  
C  
C CHARACTERISTICS:  
C =====  
C  
CC  
C SUBROUTINES REQUIRED: RS,CV  
C -----  
C  
CC  
C SIZE:  
C -----  
C  
CC  
C PDP11/03: 1066  
C  
C  
C-----C  
C  
C  
CC  
C PROGRAM:  
C =====  
C  
C  
CC  
C SUBROUTINE FREQST(LIST,IL,IPA,IW)  
C  
CC  
C LOGICAL\*1 IU  
C  
CC  
C DIMENSION LIST(1),IPA(1),IW(1),F(16)  
C COMMON/BUFC/ RES(200),RES1(10)  
C COMMON/NUSA/ MF2,MSP,IHR,AN(50)  
C COMMON/FRC/ MPF,TF  
C  
CC  
C DATA F/0.010,0.012,0.015,0.017,0.020,0.022,0.025,0.030,  
C \*0.035,0.040,0.045,0.050,0.060,0.070,0.080,0.090/  
C \*B/0.628318531/  
C  
CC  
C ALLOCATE SPACE IN THE VECTOR IPA FOR 12 WORDS.  
C  
CC  
C CALL RS(IL,12,LIST,IW)  
C  
CC  
C ENOUGH FREE SPACE?  
C  
CC  
C IF(IL.LT.0) RETURN  
C  
CC  
C SET ADDRESSES IN NODE.  
C  
CC  
C K1=LIST(IL+3)  
C LIST(IL+4)=K1+2  
C LIST(IL+5)=K1+4  
C LIST(IL+6)=K1+6  
C K5=LIST(IL+7)  
C  
CC  
C DO 10 I=K1,K5  
C 10 IPA(I)=0  
C  
CC  
C 11 WRITE(7,100)  
C 100 FORMAT(1X,'PLEASE DEFINE THE FREQUENCY LIMITS,IF YOU  
C \* WANT TO WORK'/1X,' IN THE UNIT HZ ANSWER H AND IF YOU WANT RAD/S  
C \* ANSWER R, '//1X,\*,' WHICH UNIT DO YOU WANT?')  
C READ(5,200) IU  
C 200 FORMAT(A1)  
C IF(IU.EQ.72.OR.IU.EQ.82) GO TO 12  
C WRITE(7,300)  
C GO TO 11  
C 300 FORMAT(1X,' ?ILL CMD?')  
C

```

C
C      IF THE UNIT IS HZ IHR=0 ELSE IHR=1.
C
C      12      IHR=0
C              IF(IU.EQ.72) IHR=1
C              RES1(1)=FLOAT(IHR)
C
C      13      WRITE(7,400)
C      400     FORMAT(1X,*, 'LOWER FREQUENCY LIMIT:')
C              READ(5,500) F1
C      500     FORMAT(F10.0)
C              RES1(2)=F1
C
C      600     WRITE(7,600)
C              FORMAT(1X,*, 'HIGHER FREQUENCY LIMIT:')
C              READ(5,500) F2
C              RES1(3)=F2
C              IF(F2.GE.F1) GO TO 20
C              WRITE(7,300)
C              GO TO 13
C
C      IF THE FREQUENCIES ARE GIVEN IN RAD/SEC, THEY ARE 6
C      TRANSLATED TO THE UNIT HZ.
C
C      20      IF(IU.EQ.72) GO TO 14
C              F1=F1/(B*10.)
C              F2=F2/(B*10.)
C
C      14      WRITE(7,700)
C      700     FORMAT(1X, 'HOW MANY FREQUENCIES PER DECADE DO YOU WANT' /1X, *,
C              * '(POSSIBLE VALUES ARE 1,2,4,8 AND 16)?' )
C              READ(5,800) MP
C      800     FORMAT(I3)
C              RES1(4)=FLOAT(MP)
C              IF(MOD(16,MP).EQ.0) GO TO 21
C              WRITE(7,300)
C              GO TO 14
C
C      21      WRITE(7,80)
C      80      FORMAT(1X, 'PLEASE CHOOSE OVER HOW MANY PERIODS
C              * OF THE' /1X, 'FREQUENCY THAT THE INTEGRATION IN THE CORRELATION'
C              */1X, 'METHOD WILL LAST.')
C              WRITE(7,85)
C      85      FORMAT(1X, 'IF YOU ANSWER 0 THE NUMBER WILL BE SET DEPENDING
C              * ON THE' /1X, 'FREQUENCY AS FOLLOWS:')
C              WRITE(7,81)
C      81      FORMAT(5X, '1 IF F<0.1HZ(0.6284 RAD/SEC)' /
C              *5X, '3 IF 0.1<F<0.2HZ(1.2568 RAD/SEC)' /
C              *5X, '5 IF 0.2<F<0.5HZ(3.1420 RAD/SEC)' /
C              *4X, '10 IF 0.5<F<1HZ(6.2840 RAD/SEC)' /
C              *4X, '20 IF F>1HZ' /)
C
C      82      WRITE(7,82)
C              FORMAT(1X, *, 'INTEGRATION TIME(NUMBER OF PERIODS):')
C              READ(5,800) MPI
C              RES1(5)=FLOAT(MPI)
C              MPF=MPI
C      50      WRITE(7,83)
C      83      FORMAT(1X, *, 'TRANSIENT TIME(SEC):')
C              READ(5,500) TT
C
C      IS THE TRANSIENT TIME SO LONG THAT IT WILL BE INTEGER OVERFLOW
C      IN THE ROUTINE FREQ?
C
C

```

```

PT=1./F1
IF(IHR.EQ.0) PT=PT*10.*B
SPT=PT
30 IF(TT.LE.SPT) GO TO 31
SPT=SPT+PT
31 OF=SPT/(FLOAT(MSP)*0.02)
IF(OF.LT.32700.) GO TO 84
WRITE(7,84)
84 FORMAT(1X,'WITH THE ACTUAL MAIN SAMPLE PERIOD IS'
*1X,'THE TRANSIENT TIME TOO LONG!')
GO TO 50
86 RES1(6)=TT
TF=TT

C
C STORE PATAMETERS IN THE PARAMETERAREA.
C
CALL CV(IPA(K1),F1)
CALL CV(IPA(K1+2),F2)
IPA(K5-4)=MPI
CALL CV(IPA(K1+4),TT)
IPA(K5-3)=MP

C
C CALCULATE THE STANDARD FREQUENCIES.
C
C=1.
IF(IHR.EQ.0) C=B
KA=16/MP
DT=1./50.*FLOAT(MSP)
K=1
DO 15 I=1,MP
KVOT=INT(C/(F(K)*DT)+0.5)
AN(I)=FLOAT(KVOT)
15 K=K+KA
L=1
K=1
DO 17 I=1,MP
NI=L*MP+I
KVOT=INT(C/(F(K)*DT*10**L)+0.5)
AN(NI)=FLOAT(KVOT)
K=K+KA
IF(AN(NI).LE.8.) GO TO 18
17 CONTINUE
L=L+1
GO TO 16
18 MP2=NI
IF(AN(NI).LT.8.) MP2=NI-1
RETURN
END

```

NAME: FRCL  
-----

SUBTITLE: GENERATES MORE FREQUENCIES  
-----

LANGUAGE: FORTRAN IV  
-----

KEYWORDS: MORE FREQUENCIES  
-----

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN      DATE: 27-APR-78  
-----

INSTITUTE:  
-----

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:  
-----

VERSION:  
-----

PURPOSE:  
=====

THE ROUTINE:

1. CALCULATES A NEW FREQUENCY
2. CALCULATES THE NUMBER OF SAMPLES DURING THE TRANSIENT-  
    AND INTEGRATION TIME
3. SEND THE PARAMETERS TO FOREGROUND NODES

USAGE:  
=====

PROGRAM TYPE: SUBROUTINE  
-----

ARGUMENTS:  
-----

FRCL(NF1, NF2, NF3)

NF1	-NUMBER OF SAMPLES DURING ONE PERIOD, WHICH CORRESPONDS TO THE LOWER FREQUENCY LIMIT
NF2	-NUMBER OF SAMPLES DURING ONE PERIOD, WHICH CORRESPONDS TO THE HIGHER FREQUENCY LIMIT
NF3	-THE LENGHT BETWEEN TWO FREQUENCIES IN THE UNIT: NUMBER OF SAMPLES

COMMON BLOCKS:  
-----

FRCL	-CONTAINING INFORMATION ABOUT INTEGRATION TIME AND TRANSIENT TIME
NUSA	-CONTAINING INFORMATION ABOUT THE FREQUENCIES
CRL	-CONTAINING INFORMATION ABOUT PARAMETERS IN FOREGROUND NODES
WRK	-WORK AREA
FRAN	-CONTAINING INFORMATION ABOUT PERIOD, NUMBER

```

C           OF IN AND OUTPUT SIGNALS
C   FFLAG  -CONTAINING FLAGS
C   RP      -CONTAINING INFORMATION ABOUT PARAMETERS
C           IN FOREGROUND NODES
C   BUFC    -CONTAINING THE RESULT VECTORS

```

```

C   CHARACTERISTICS:
C   =====

```

```

C   SUBROUTINES REQUIRED: OPEN,OPCOM
C   -----

```

```

C   SIZE:
C   -----

```

```

C   PDP11/03: 347

```

```

C   PROGRAM:
C   =====

```

```

C   SUBROUTINE FRCL(NF1,NF2,NF3)

```

```

C   COMMON/FRC/ MPI,TT
C   COMMON/NUSA/ MP2,MSP,IHR
C   COMMON/CRL/ MPA,MCO,NYF,MR,W,SG,MPE,AM
C   COMMON/WRK/ IW1(50),IW2(50),IW3(50),IW4(50)
C   COMMON/FRAN/ IPER,IWI(10),IWO(10)
C   COMMON/FFLAG/ IFAN,IFG,IBG
C   COMMON/RP/ KA,IWP(10),FIW(10)
C   COMMON/BUFC/ RES(200)

```

```

C   DATA BA/6.28318531/

```

```

C   CALCULATE A NEW FREQUENCY,

```

```

C   SP=FLOAT(MSP)
C   SG=-1.
C   F1=50./SP/FLOAT(NF1)
C   W=F1*BA
C   MCO=MCO+1
C   RES(4*MCO+1)=F1
C   IF(IHR.EQ.0) RES(4*MCO+1)=W

```

```

C   INTEGRATION TIME,

```

```

C   MPE=MPI
C   IF(MPE.GT.0) GO TO 11

```

```

C   SET THE INTEGRATION TIME DEPENDING ON THE FREQUENCY,

```

```

C   MPE=1
C   IF(F1.GE.0.1) MPE=3
C   IF(F1.GE.0.2) MPE=5
C   IF(F1.GE.0.5) MPE=10
C   IF(F1.GE.1.) MPE=20

```

```

C   11 RES(4*MCO+4)=FLOAT(MPE)

```

```

C   CONVERT THE NUMBER OF INTEGRATION PERIODS INTO A

```



```

C      NUMBER OF SAMPLES.
C
C      MPA=NF1*MPPE
C
C      CONVERT THE TRANSIENT TIME INTO A NUMBER OF SAMPLES.
C
C      TT1=50./SP*TT
C      NT=IFIX(TT1)
C      L=0
12     MTS=NF1*L
C      IF(MTS.GE.NT) GO TO 13
C      L=L+1
C      GO TO 12
C
C      13     DO 14 I=1,50
C      14     IW4(I)=0
C
C      OPEN SINU1 AND SET THE FREQUENCY.
C
C      IW4(1)=2
C      IW4(2)=1
C      IW4(5)=46
C      IW4(6)=1
C      IFAN=1
C      IPER=1
C      IWI(1)=0
C      IWO(1)=1
C      IWO(2)=1
C      IWP(2)=1
C      IWF(1)=0
C      KA=2
C      FIW(2)=F1
C      CALL OPEN
C
C      OPEN SINU2 AND SET THE FREQUENCY AND THE PHASE.
C
C      IW4(5)=46
C      IW4(6)=2
C      IWO(2)=2
C      DO 15 I=1,3
15     IWP(I)=1
C      KA=3
C      FIW(1)=1.
C      FIW(2)=F1
C      FIW(3)=1.57079633
C      CALL OPEN
C
C      OPEN TIME3 AND SET TRANSIENT TIME, INTEGRATION TIME AND A COUNTER.
C
C      IW4(5)=42
C      IW4(6)=3
C      IPER=MTS
C      IF(MTS.EQ.0) IPER=1
C      IWO(1)=0
C      IWO(2)=0
C      IWP(4)=1
C      KA=4
C      FIW(1)=FLOAT(MTS)
C      FIW(2)=FLOAT(MPA)
C      FIW(3)=-1
C      CALL OPEN
C
C      ACT1 SINU1, SINU2, TIME3
C

```

```
IFG=1
IW4( 1 )=4
IW4( 2 )=2
IW4( 5 )=46
IW4( 6 )=1
IW4( 8 )=46
@ IW4( 9 )=2
IW4( 11 )=42
IW4( 12 )=3
CALL OPCOM
```

```
C
C
C
```

```
RESET THE FLAGS.
```

```
IFAN=0
IFG=0
NYF=0
```

```
C
C
C
```

```
NEXT FREQUENCY.
```

```
NF1=NF1-NF3
IF(NF1.LT.NF2) SG=1.
```

```
C
C
```

```
RETURN
END
```

NAME: FRCLST

SUBTITLE: START-UP ROUTINE TO THE ROUTINE FRCL

LANGUAGE: FORTRAN IV

KEYWORDS: START-UP

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN      DATE: 26-APR-78

INSTITUTE:

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:

VERSION:

PURPOSE AND METHOD:

THE ROUTINE:

1. ALLOCATES SPACE IN THE PARAMETERAREA.
2. SETS THE ADDRESSPART IN THE NODE HEAD, ADDRESSES TO THE PARAMETERAREA.
3. ASKS FOR AND SETS SOME PARAMETER VALUES.

USAGE:

PROGRAM TYPE: SUBROUTINE

ARGUMENTS:

FRCLST(LIST, IL, IPA, IW)

IL	--POINTER TO THE FIRST WORD IN THE ACTUAL NODE
IPA	--VECTOR CONTAINING THE PARAMETERS IN BACKGROUND JOB
IW	--WORKAREA

COMMON BLOCKS:

CRL	--CONTAINING DIFFERENT TYPES OF PARAMETERS
NUSA	--CONTAINING INFORMATION ABOUT THE FREQUENCIES

CHARACTERISTICS:

SUBROUTINES REQUIRED: RS, CV

SIZE:

PDP11/03: 580 WORDS

C45

PROGRAM:

=====

SUBROUTINE FRCLST(LIST, IL, IPA, IW)

DIMENSION LIST(1), IPA(1), IW(1)

LOGICAL\*1 IVAL

COMMON/CRL/ MPA, MCO, NYF, MR, W, SG, MPE, AM, ARG1  
COMMON/NUSA/ MP2, MSP, IHR

DATA BA/6.28318531/

ALLOCATE SPACE IN THE VECTOR IPA FOR 3 WORDS.

CALL RS(IL, 3, LIST, IW)  
IF(IL, LT, 0) RETURN

NYF=1  
MCO=-1  
ARG1=0.

1 WRITE(7, 2)  
2 FORMAT(1X, #, 'LOWER FREQUENCY LIMIT:')  
READ(5, 3) F1  
3 FORMAT(F10.0)

WRITE(7, 4)  
4 FORMAT(1X, #, 'HIGHER FREQUENCY LIMIT:')  
READ(5, 3) F2

IF(F2.GT.F1) GO TO 6  
WRITE(7, 5)  
5 FORMAT(1X, ' ?ILL CMD?' )  
GO TO 1

6 DT=0.02\*FLOAT(MSP)  
CA=1.  
IF(IHR.EQ.0) CA=BA

HIGHER AND LOWER FREQUENCY LIMIT CONVERTED INTO NUMBER  
OF SAMPLES PER PERIOD.

NF1=INT(CA/(DT\*F1)+0.5)  
NF2=INT(CA/(DT\*F2)+0.5)  
IS=NF1-NF2+1  
IF(IS.GT.0) GO TO 8

WRITE(7, 7)  
7 FORMAT(1X, 'THE FREQUENCY INTERVAL IS NOT WIDE ENOUGH. TRY AGAIN!')  
GO TO 1

8 IR=IS

```

      IF(IR.GT.50) IR=50
      IF(IR.LE.50.OR.IR.GE.75) GO TO 50
      IR=INT(FLOAT(IR)/2.+0.5)
50    WRITE(7,9) IR
      9    FORMAT(1X,'THE MAXIMUM NUMBER OF FREQUENCIES ARE:',I5)
C
10    WRITE(7,11)
11    FORMAT(1X,*, 'HOW MANY DO YOU (APPROXIMATELY) WANT?')
      READ(5,12) IT
12    FORMAT(I5)
      IF(IT.LE.IR) GO TO 13
      WRITE(7,5)
      GO TO 10
C
      THE INTERVAL BETWEEN TWO FREQUENCIES.
C
13    NF3=INT(FLOAT(IS)/FLOAT(IT)+0.5)
      NF4=INT(FLOAT(IS)/FLOAT(NF3)+0.5)
14    WRITE(7,15) NF4
15    FORMAT(1X,'THE ACTUAL NUMBER OF FREQUENCIES ARE:',I5/
      *1X,*, 'DO YOU WANT TO CHANGE THIS NUMBER?')
      READ(5,16) IVAL
16    FORMAT(A1)
      IF(IVAL.EQ.89.OR.IVAL.EQ.78) GO TO 17
      WRITE(7,5)
      GO TO 14
17    IF(IVAL.EQ.89) GO TO 1
C
      SET ADDRESSES IN THE NODE HEAD.
      K1=LIST(IL+3)
      LIST(IL+4)=K1+1
      K5=LIST(IL+7)
      DO 18 I=K1,K5
18    IPA(I)=0
C
      SET PARAMETERS IN THE PARAMETERAREA.
C
      IPA(K1)=NF1
      IPA(K1+1)=NF2
      IPA(K5)=NF3
C
      RETURN
      END

```

C  
C  
C NAME: OPCOM  
C  
C  
C

C  
C SUBTITLE: OPERATOR COMMUNICATION ROUTINE  
C  
C  
C

C  
C LANGUAGE: FORTRAN + RT-11-FORTRAN  
C  
C  
C

C  
C IMPLEMENTOR: LEIF RADING, TYKE PAULSSON  
C  
C  
C

DATE: 7-SEP-77  
C  
C

C  
C REVISED BY: CHRISTER LARSSON, HARRIET LINDGREN  
C  
C  
C

DATE: 28-APR-78  
C  
C

C  
C INSTITUTE:  
C  
C  
C

C  
C DEPARTMENT OF AUTOMATIC CONTROL  
C  
C LUND INSTITUTE OF TECHNOLOGY, SWEDEN  
C  
C

C  
C  
C PURPOSE AND METHOD  
C  
C  
C

=====

C  
C OPCOM TAKES CARE OF THE OPERATOR COMMUNICATION.  
C  
C WHEN THE PROGRAM PRINTS ">" IT IS READY TO RECEIVE A COMMAND,  
C  
C AFTER READING AND IDENTIFYING THE COMMAND THE PROGRAM EITHER  
C  
C CALLS A SUBROUTINE THAT TAKES CARE OF THE COMMAND OR EXECUTES  
C  
C THE COMMAND ITSELF.  
C  
C

C  
C USAGE  
C  
C  
C

C  
C COMMON BLOCKS:  
C  
C  
C

C  
C BG --CONTAINING LIST- AND PARAMETERARRAYS FOR BACKGROUND  
C  
C JOB  
C  
C FLA --CONTAINING FLAGS AND POINTERS WITH INFORMATION  
C  
C ABOUT COLLECTING AND CALCULATING BLOCKS.  
C  
C CMD --CONTAINING 4-CHARACTERS COMMANDNAME AND 2-CHARACTERS  
C  
C PARAMETERNAME.  
C  
C WRK --WORK AREA  
C  
C FG --CONTAINING ARRAYS FOR SENDING AND RECIEVING MESSAGES  
C  
C AND INFORMATION BETWEEN THE JOBS.  
C  
C

C  
C NEW COMMON BLOCKS:  
C  
C  
C

C  
C FFLAG --CONTAINING FLAGS.  
C  
C  
C

C  
C CHARACTERISTICS  
C  
C  
C

C  
C SIZE:  
C  
C  
C

C  
C PDF11/03: 936 WORDS  
C  
C  
C

C  
C SUBROUTINE REQUIRED: CRNF, FRENOD, GRAFST, GROUST, LSTB, LSTF, MAXMST  
C  
C NORMST, OPEN, STAFST, TECK, TFD, WRIST, CRLMST, FRCLST,  
C

FROM SYSTEM LIBRARY: IPEEK, IPOKE, ISDAT

SUBROUTINE OPCOM

LOGICAL\*1 ICM

COMMON /BG/LIST(120), IPA(512)  
 COMMON /FLA/IFL1, ICN, NBN, ICO, IFL2  
 COMMON /CMD/ CM1(20), CM2(20), CM3(20)  
 COMMON /FG/ IV(100)  
 COMMON /WRK/ IW1(50), IW2(50), IW3(50), IW4(50)  
 COMMON/FFLAG/ IFAN, IFG, IBG

ACTIVATE FOREGROUND OR BACKGROUND NODE IN FREQUENCY ANALYSIS?

IF(IFG.EQ.1) GO TO 200  
 IF(IBG.EQ.1) GO TO 350

CALL IPOKE('44', '177677', AND, IPEEK('44'))

500 WRITE(7, 500)  
 FORMAT(1X, '#, / >')

READ THE COMMAND LINE.

10 READ(5, 10) (IW1(I), I=1, 25)  
 FORMAT(25A2)

IDENTIFY THE CHARACTERSTRING.

CALL TECK(IW1, IW4, 1, 60, 4, 50)

TEST COMMAND LINE.

15 IF(IW4(1).GT.0.AND.IW4(2).GT.0.AND.IW4(2).LE.20) GO TO 20  
 501 WRITE(7, 501)  
 FORMAT(1X, '?ILL CMD?')  
 GO TO 600

20 GO TO (30, 40, 40, 301, 301, 301, 301, 305, 306, 307,  
 \*308, 307, 307, 307, 307, 307, 307, 307, 307), IW4(2)

FOREGROUND NODE?

30 IF(IW4(5)-40) 15, 15, 35

SET A NODE HEAD AND PARAMETERS (FOREGROUND NODE)

35 CALL OPEN  
 GO TO 600

40 IF(IW4(5).GE.41) GO TO 200  
 IF(IW4(5).LE.20.OR.IW4(5).GE.39)GO TO 15

ACTIVATE A BACKGROUND NODE.

350 NN=IW4(6)

LEGAL NODE NUMBER?

IF(NN.GT.0.AND.NN.LT.10) GO TO 45  
 WRITE(7, 502)

```

502  FORMAT(1X,'?ILL NODE NUMBER?')
      GO TO 600
C
45   ID=IW4(5)-20
      ITY=1000*ID+NN
C
      DO 50 I=3,120,12
      IF(LIST(I).EQ.ITY) GO TO 55
50   CONTINUE
C
      IF(IW4(2).EQ.2) GO TO 70
      WRITE(7,503)
503  FORMAT(1X,'NOT FOUND')
      GO TO 600
C
55   IF(IW4(2).EQ.2) GO TO 65
C
C     DEACTIVATE ONE NODE.
C
60   IF(LIST(I-2).LT.0) WRITE(7,504) CM2(ID),NN
504  FORMAT(1X,A4,I1,1X,'NOT ACTIVE')
      LIST(I-2)=-1
      LIST(I-1)=-1
      GO TO 600
C
65   IF(LIST(I-2).LT.0) GO TO 70
      WRITE(7,505) CM2(ID),NN
505  FORMAT(1X,A4,I1,1X,'ALREADY ACTIVE')
      GO TO 600
C
C     ACTIVATE ONE NODE.
C
70   CALL FRENOD(LIST,IL)
      IF(IL.LT.0) GO TO 600
C
C     READ SAMPLE PERIOD.
C     WHEN FREQUENCY ANALYSIS, SAMPLE PERIOD=1.
C
      IF(IB6.NE.1) GO TO 81
      LIST(IL+1)=1
      GO TO 82
81   WRITE(7,506)
506  FORMAT(1X,*, 'SAMPLE PERIOD:')
      READ(5,80) LIST(IL+1)
80   FORMAT(I10)
C
82   LIST(IL+2)=ITY
C
C     IF THE NODE IS OF TYPE 'FREQ' OR 'FRCL', NO INSIGNALS ARE TO BE
C     RECIEVED.
C
      LIST(IL)=0
      IF(ITY.EQ.8001.OR.ITY.EQ.10001) GO TO 84
      SET INFORMATION ABOUT WERE TO RECEIVE INPUT SIGNALS FROM,
C
83   CALL TFD(CM1,LIST,IL,IW1,IW2)
C
84   GO TO (100,101,102,103,104,105,106,107,108,109,
      *110,110,110,110,110,110,110,110,110,110) ID
C
100  CALL GRAFST(LIST,IL,IPA,IW1)
      GO TO 600
C
101  CALL WRIST(LIST,IL,IPA,IW1,IW2)

```



```

C          GO TO 600
C 102     CALL MAXMST(LIST, IL, IPA, IW1, IW2)
          GO TO 600
C
C 103     CALL WRIST(LIST, IL, IPA, IW1, IW2)
          GO TO 600
C
C 104     CALL GROUST(LIST, IL, IPA, IW1)
          GO TO 600
C 105     CALL STAFST(LIST, IL, IPA, IW1)
          GO TO 600
C 106     CALL NORMST(LIST, IL, IPA, IW1)
          GO TO 600
C 107     CALL FREQST(LIST, IL, IPA, IW1)
          GO TO 600
C 108     CALL CRLMST(LIST, IL, IPA, IW1)
          GO TO 600
C 109     CALL FRCLST(LIST, IL, IPA, IW1)
          GO TO 600
C 110     CONTINUE
          GO TO 600
C
C
C 200     ACTIVATE FOREGROUND NODES.
          IV(1)=12
          IFL2=0
C
C RECEIVE INFORMATION ABOUT ACTIVE NODES AND WAIT FOR COMPLETION.
C
C CALL ISDAT(IV, 1)
C 210     IF(IFL2) 220, 210, 220
C
C ACTIVATE OR DEACTIVATE ?
C
C 220     IW1(1)=9
          IF(IW4(2).EQ.3) IW1(1)=10
C
C DO 250 I=2, IW4(1)
          IF(IW4(3*I), LE.0, OR, IW4(3*I), GE.21) GO TO 270
          DO 230 I1=1, IV(4)
C 230     IF(IV(I1*4+2), EQ, IW4(I*3), AND, IV(I1*4+1), EQ, IW4(I*3-1)
          *-40) GO TO 250
          WRITE(7, 245)
C 245     FORMAT(1X, 'THE NODE IS NOT ACTIVATED OR OCCUPIED')
          GO TO 600
C 250     IW1(I)=IW4(I*3)
          DO 260 I=1, IW4(1)
C 260     IV(I)=IW1(I)
          CALL ISDAT(IV, IW4(1))
          GO TO 600
C 270     WRITE(7, 502)
          GO TO 600
C
C STOP REGULATION, START REGULATION, STOP COLLECTION, START COLLECTION.
C
C 301     IW1(1)=IW4(2)-3
          CALL ISDAT(IW1, 1)
          GO TO 600
C
C OPEN NEW FILE.
C
C 305     CALL CRNF(ICN, ICO, NBN, IFL1, IW1)
          GO TO 600

```

```
C
C      LIST ACTIVE NODES. ( FOREGROUND )
C
306   I=7
      IF( IW4( 4 ), EQ, 6 ) I=IW4( 4 )
      CALL LSTF( IV, IFL2, CM3, I )
      GO TO 600

C
C      LIST ACTIVE NODES. ( BACKGROUND )
C
307   I=7
      IF( IW4( 4 ), EQ, 6 ) I=IW4( 4 )
      CALL LSTB( LIST, CM2, I )
      GO TO 600

C
C      CHECK IF THE LENGTH OF THE MAIN SAMPLE PERIOD HAS BEEN RIGHT.
C
308   IW1( 1 )=IW4( 2 )+3
      IFL2=0
      CALL ISDAT( IW1, 1 )
309   IF( IFL2 ) 310, 309, 310
310   WRITE( 7, 311 ) IV( 4 )
311   FORMAT( 1X, ' THE MAIN SAMPLE PERIOD HAS BEEN TO LONG', I5, ' TIMES.' /
*1X, ' THE COUNTER WILL NOW BE SET ZERO.' )
      GO TO 600

C
C      WHEN FREQUENCY ANALYSIS, NO ##### ARE WANTED
C
600   IF( IF6, EQ, 1 ) RETURN
      IF( IF6, EQ, 1 ) RETURN
      WRITE( 7, 507 )
507   FORMAT( 1X, ' #####' )
      RETURN
      END
```

C  
C  
C NAME: OPEN  
C  
C  
C

C  
C SUBTITLE: SETS A NODEHEAD AND PARAMETERS  
C  
C  
C

C  
C LANGUAGE: FORTRAN IV + RT-11-FORTRAN  
C  
C  
C

C  
C IMPLEMENTOR: LEIF RADING, TYKE PAULSSON  
C  
C  
C

DATE: 2-SEP-77  
C  
C

C  
C REVISED BY: CHRISTER LARSSON, HARRIET LINDGREN  
C  
C  
C

DATE: 28-APR-78  
C  
C

C  
C INSTITUTE:  
C  
C  
C

C  
C DEPARTMENT OF AUTOMATIC CONTROL  
C  
C LUND INSTITUTE OF TECHNOLOGY, SWEDEN  
C  
C

C  
C  
C PURPOSE AND METHOD  
C  
C  
C

=====

C  
C IF THE NODE "N1" IS IN FOREGROUND'S NODELIST IT IS  
C  
C FETCHED, ELSE THE STARTUPROUTINE TO THE NODETYPE,  
C  
C IF ANY, IS RUN. THE NODEHEAD ARE SET BY THE SUBROUTINE  
C  
C SNOD. IF THE NODE IS ACTIVE, THE PARAMETERS IS FETCHED  
C  
C ELSE A REQUEST IS MADE TO FOREGROUND JOB TO SEE IF ENOUGH  
C  
C ROOM IS AVAILABLE IN FOREGROUND'S PARAMETERLIST FOR THE  
C  
C NEW NODE. FINALLY THE PARAMETERS ARE SET OR CHANGED BY  
C  
C THE SUBROUTINE SPAR.

C  
C NOTE: THE ENGINEERNODE HAS NO NODEHEAD.  
C  
C

C  
C USAGE  
C  
C  
C

=====

C  
C ARGUMENTS: NONE  
C  
C  
C

C  
C COMMON BLOCKS  
C  
C  
C

C  
C FLA --CONTAINING FLAGS AND POINTERS WITH INFORMATION  
C  
C ABOUT COLLECTING AND CALCULATING BLOCKS  
C  
C CMD --CONTAINING 4-CHARACTERS COMMANDNAME AND 2-  
C  
C CHARACTERS PARAMETERNAME  
C  
C IM --CONTAINING 1-CHARACTER CONSTANTS  
C  
C RD --CONTAINING INFORMATION ABOUT TYPE OF STARTUPROUTINE,  
C  
C (IF ANY) AND NUMBER OF PARAMETERS FOR EACH TYPE OF NODE  
C  
C FB --CONTAINING ARRAYS FOR SENDING AND RECEIVING MESSAGES  
C  
C AND INFORMATION BETWEEN THE JOBS  
C  
C WRK --WORK AREA  
C  
C

C  
C NEW COMMON BLOCKS:  
C  
C  
C

C  
C FFLAG --CONTAINING FLAGS  
C  
C  
C

C  
C CHARACTERISTICS  
C  
C  
C

=====

C  
C SIZE:

C  
C -----  
C FDP11/03: 551 WORDS

C  
C SUBROUTINES REQUIRED: TECK, SNOD, SPAR, RNOD, RPAR

C  
C -----  
C FROM SYSTEM LIBRARY: ISDAT  
C  
C-----

C  
C SUBROUTINE OPEN

C  
C  
C COMMON /FLA/IFL1, ICN, NBN, ICO, IFL2  
C COMMON /CMD/CM1( 20 ), CM2( 20 ), CM3( 20 ), IOP( 100 )  
C COMMON /RD/ IRD( 20 )  
C COMMON /FG/IV( 100 )  
C COMMON /WRK/IW1( 50 ), IW2( 50 ), IW3( 50 ), IW4( 50 )  
C COMMON /FFLAG/ IFAN, IFG

C  
C NODENUMBER

C  
C N1=IW4( 6 )

C  
C TYPE OF NODE

C  
C ID=IW4( 5 )-40

C  
C IF IRD( ) < 0 : THE ROUTINE HAS A START-UP ROUTINE  
C ABS( IRD( ) ) =NUMBER OF PARAMETERS  
C ABS( IRD( ) )/100 =NUMBER OF NOT STATUSPARAMETERS  
C  
C

C  
C NS=IABS( IRD( ID ) )  
C NS1=NS/100  
C NS2=MOD( NS, 100 )  
C NS=2\*NS2

C  
C ENGINEER-NOD?

C  
C IF( ID.EQ.20 ) GO TO 300

C  
C LEGAL NODENUMBER?

C  
C IF( N1.GE.1.AND.N1.LE.20 ) GO TO 10

C  
C WRITE( 7, 5 )  
C 5 FORMAT( 1X, ' ?ILL NODNUMBER?' )  
C RETURN

C  
C SEND FOR A NODE AND WAIT FOR COMPLETION

C  
C 10 IV( 1 )=5  
C IV( 2 )=( N1-1 ) \* 12 + 1  
C IV( 3 )=12  
C IFL2=0  
C CALL ISDAT( IV, 3 )

C  
C 20 IF( IFL2 ) 30, 20, 30

C  
C THE NODE IS IN IV  
C IS THE NODE ACTIVE?

C  
C 30 IF( IV( 5 ), EQ. 0 ) GO TO 50

```

C
C      IS THE NODE OF THE SAME TYPE AS THE REQUESTED?
C
C      IF(IV(6).EQ.ID) GO TO 310
C
C      WRITE(7,45)
45     FORMAT(1X,'THE NODE IS ALREADY ACTIVE AS ANOTHER TYPE')
      RETURN
C
C      50     NS=2*NS1
C
C      HAS THE NODE A STARTUP-ROUTINE?
C
C      IF(IRD(ID).GT.0) GO TO 210
      IP5=0
      IP6=0
C
C      GO TO(110,120,120,120,120) ID-10
C
C      START UP THE FRB-GENERATOR.
C
C      110    WRITE(7,508)
508    FORMAT(1X,#,'NUMBER OF BITS:')
      READ(5,509) NA
509    FORMAT(I15)
      IF(NA.GE.3.AND.NA.LE.15) GO TO 115
      WRITE(7,510)
510    FORMAT(1X,'NOT IN RANGE')
      GO TO 110
115    NS=NS+NA*2+1
      CALL PRBST(IW3,IW3(NA+1),NA)
      IP5=NA
      IW3(NA*2+1)=NA
      GO TO 210
C
C      120    NS=NS+2
      IW3(1)=0
      IW3(2)=0
      GO TO 210
C
C
C
C      ENOUGH ROOM IN FORGROUND'S PARAMETERVECTOR?
C      ASK AND WAIT FOR COMPLETION
C
C      210    IV(1)=11
      IV(2)=NS
      IFL2=0
      CALL ISDAT(IV,2)
220    IF(IFL2) 230,220,230
C
C      ENOUGH ROOM?
C
C      230    IF(IV(4).GE.0) GO TO 270
C
C      WRITE(7,235)
235    FORMAT(1X,'NOT ENOUGH ROOM IN PARAMETERVECTOR')
      RETURN
C
C      SET A NODE IN IV
C
C      270    IP2=IV(4)
      IV(8)=NS1*2+IP2
      IV(9)=IP5+IV(8)

```

```

IV(10)=IP6+IV(8)
IP3=IP2+NS-1
C
C      WHEN FREQUENCY ANALYSIS, CALL 'RNOD' INSTEAD OF 'SNOD'.
C
      IF(IFAN.NE.1) GO TO 400
      CALL RNOD(ID,N1,IP2,IP3)
      GO TO 401
400    CALL SNOD(ID,N1,IP2,IP3)
C
C      SEND THE NODE TO FOREGROUND
C
401    CALL ISDAT(IV,12)
C
      DO 290 I=4,48
290    IV(I)=0
C
      ADD THE PARAMETERS FROM THE STARTUP-ROUTINE
C
      J1=NS1*2+3
      J=NS-J1+3
C
      DO 350 I=1,J
350    IV(J1+I)=IW3(I)
      GO TO 340
C
      SEND FOR THE PARAMETERS AND WAIT FOR COMPLETION
C
300    IP2=27
      IP3=74
      GO TO 320
C
310    IP2=IV(7)
      IP3=IV(8)
C
      IF FREQUENCY ANALYSIS NO TEXT IS WANTED.
C
      IF(IFAN.EQ.1) GO TO 320
      WRITE(7,315)
315    FORMAT(1X,'THE NODE IS ALREADY ACTIVE')
320    IV(1)=6
      IV(2)=IP2
      IV(3)=IP3-IP2+1
      IFL2=0
      CALL ISDAT(IV,3)
330    IF(IFL2) 340,330,340
C
      THE PARAMETERS ARE IN IV
      SET OR CHANGE THE PARAMETERS IN IV.
      WHEN FREQUENCY ANALYSIS, CALL RPAR INSTEAD OF SPAR.
C
340    IF(IFAN.NE.1) GO TO 342
      CALL RPAR(IP2,NS)
      GO TO 341
C
342    CALL SPAR(ID,IP2,NS)
C
C      SEND THE PARAMETERS TO FOREGROUND JOB
C
341    CALL ISDAT(IV,NS+3)
C

```

RETURN  
END





```
C      =====
C
C      SUBROUTINE RESO(LUN,RES1)
C
C      DIMENSION RES1(1)
C
C      IF(IFIX(RES1(1)).EQ.0) GO TO 1
C      WRITE(LUN,100)
100    FORMAT(1X,'THE FREQUENCIES ARE GIVEN IN THE UNIT HZ./')
C      GO TO 2
C
C      1    WRITE(LUN,200)
200    FORMAT(1X,'THE FREQUENCIES ARE GIVEN IN THE UNIT RAD/SEC./')
C
C      2    WRITE(LUN,300) RES1(2),RES1(3)
300    FORMAT(1X,'LOWER FREQUENCY LIMIT=',E10.2//1X,'HIGHER FREQUENCY
C      *LIMIT=',E10.2/)
C
C      MP=IFIX(RES1(4))
C      WRITE(LUN,400) MP
400    FORMAT(1X,'NUMBER OF FREQUENCIES PER DECADE=',I3/)
C
C      MPI=IFIX(RES1(5))
C      IF(MPI.EQ.0) GO TO 10
C      WRITE(LUN,500) MPI
500    FORMAT(1X,'INTEGRATION TIME (PERIODS)=' ,I3/)
C      GO TO 11
C
C      10   WRITE(LUN,750)
750    FORMAT(1X,'THE INTEGRATION TIME IS DEPENDING ON THE FREQUENCY./')
C
C      11   WRITE(LUN,600) RES1(6)
600    FORMAT(1X,'TRANSIENT TIME (SEC)=' ,F6.1/)
C
C      IF(IFIX(RES1(7)).EQ.1) GO TO 3
C      WRITE(LUN,700)
700    FORMAT(1X,'NO TREND ELIMINATION HAS BEEN DONE./')
C      GO TO 4
C
C      3    WRITE(LUN,800)
800    FORMAT(1X,'TREND ELIMINATION HAS BEEN DONE./')
C
C      4    IF(IFIX(RES1(8)).EQ.1) GO TO 5
C      WRITE(LUN,900)
900    FORMAT(1X,'NO TIME WINDOW HAS BEEN USED./')
C      GO TO 6
C
C      5    WRITE(LUN,910)
910    FORMAT(1X,'TIME WINDOW HAS BEEN USED./')
C
C      6    IF(IFIX(RES1(9)).EQ.1) GO TO 7
C      WRITE(LUN,920)
920    FORMAT(1X,'THE SYSTEM HAS BEEN RUN AUTOMATICALLY./')
C      RETURN
C
C      7    WRITE(LUN,930)
930    FORMAT(1X,'THE SYSTEM HAS BEEN RUN MANUALLY./')
C      RETURN
C      END
```

NAME: RESU  
-----

SUBTITLE: WRITES THE RESULTHEAD  
-----

LANGUAGE: FORTRAN IV  
-----

KEYWORDS: RESULTHEAD  
-----

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN  
-----

DATE: 28-APR-78  
-----

INSTITUTE:  
-----

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:  
-----

VERSION:  
-----

PURPOSE:  
=====

THE ROUTINE WRITES THE RESULTHEAD ON SPECIFIED DEVICE.

USAGE:  
=====

PROGRAM TYPE: SUBROUTINE  
-----

ARGUMENTS:  
-----

RESU(LUN, RES1)

LUN            --LOGICAL UNIT

RES1           --PARAMETERVECTOR

CHARACTERISTICS:  
=====

SUBROUTINES REQUIRED: NONE  
-----

SIZE:  
-----

PDP11/03: 408

PROGRAM:

```
C      =====
C
C      SUBROUTINE RESU(LUN,RES1)
C
C      DIMENSION RES1(1)
C
C      IF(IFIX(RES1(1)),EQ.0) GO TO 1
100    WRITE(LUN,100)
      FORMAT(1X,'THE FREQUENCIES ARE GIVEN IN THE UNIT HZ./')
      GO TO 2
C
C      1  WRITE(LUN,200)
200    FORMAT(1X,'THE FREQUENCIES ARE GIVEN IN THE UNIT RAD/SEC./')
C
C      2  WRITE(LUN,300) RES1(2),RES1(3)
300    FORMAT(1X,'LOWER FREQUENCY LIMIT=',E10.2//1X,'HIGHER FREQUENCY
      *LIMIT=',E10.2/)
C
C      MP=IFIX(RES1(4))
      WRITE(LUN,400) MP
400    FORMAT(1X,'NUMBER OF FREQUENCIES PER DECADE=',I3/)
C
C      MPI=IFIX(RES1(5))
      WRITE(LUN,500) MPI
500    FORMAT(1X,'INTEGRATION TIME (PERIODS)=',I3/)
C
C      WRITE(LUN,600) RES1(6)
600    FORMAT(1X,'TRANSIENT TIME (SEC)=',F6.1/)
C
C      IF(IFIX(RES1(7)),EQ.1) GO TO 3
      WRITE(LUN,700)
700    FORMAT(1X,'NO TREND ELIMINATION WILL BE DONE./')
      GO TO 4
C
C      3  WRITE(LUN,800)
800    FORMAT(1X,'TREND ELIMINATION WILL BE DONE./')
C
C      4  IF(IFIX(RES1(8)),EQ.1) GO TO 5
      WRITE(LUN,900)
900    FORMAT(1X,'NO TIME WINDOW WILL BE USED./')
      GO TO 6
C
C      5  WRITE(LUN,910)
910    FORMAT(1X,'TIME WINDOW WILL BE USED./')
C
C      6  IF(IFIX(RES1(9)),EQ.1) GO TO 7
      WRITE(LUN,920)
920    FORMAT(1X,'THE SYSTEM IS TO BE RUN AUTOMATICALLY./')
      RETURN
C
C      7  WRITE(LUN,930)
930    FORMAT(1X,'THE SYSTEM IS TO BE RUN MANUALLY./')
      RETURN
      END
```



```

C      SUBROUTINES REQUIRED: CV
C      -----
C      FROM THE SYSTEM LIBRARY: ICSI,IGETC,IENTER,IWRITW
C
C      SIZE:
C      -----
C      PDP11/03: 341
C
C-----
C
C      PROGRAM:
C      =====
C
C      SUBROUTINE RFILE(MR,IFST)
C
C      DIMENSION EXT(2),IDEV(3)
C
C      COMMON/BUFC/ RES(200),RES1(10)
C      COMMON/WRK/ IW1(50),IW2(50),IW3(50),IW4(50)
C      COMMON/FILE/ IC,NB,IR
C      COMMON/BUFF/ IB1(100),IB2(512),IB4(256)
C
C      DATA EXT/6RDATDAT,6RDATDAT/
C      DATA IDEV/3RDX ,3RDX0,3RDX1/
C
C      IF(IFST.EQ.1) GO TO 7
C      IC=IGETC( )
C      1  WRITE(7,100)
C      100 FORMAT(1X,'GIVE YOUR FILE A NAME.'/
C      *1X,'FOR EXAMPLE:DX1:MAT.DAT='/)
C
C      PARSE A COMMANDSTRING AND RETURN A FILE DESCRIPTION TO THE PROGRAM.
C
C      IN=ICSI(IW2,EXT,,0)
C      IF(IN.EQ.0) GO TO 2
C      WRITE(7,110)
C      110 FORMAT(1X,'?ILL CMD?')
C      GO TO 1
C
C      LEGAL DEVICE?
C
C      2  DO 3 I=1,3
C      IF(IDEV(I).EQ.IW2(1)) GO TO 4
C      3  CONTINUE
C      WRITE(7,120)
C      120 FORMAT(1X,'?ILL DEV?')
C      GO TO 1
C
C      CREATE A FILE FOR THE RESULT.
C
C      4  NB=IENTER(IC,IW2,6)
C      IF(NB.GT.0) GO TO 5
C      WRITE(7,130)
C      130 FORMAT(1X,'NOT ENOUGH ROOM ON DEVICE FOR FILE SIZE REQUESTED.')

```

```
        CALL CV(IB2(II),RES1(I))
6      CONTINUE
C
      IB2(21)=MR
      N0=21
      NST=25
      MR1=29
      GO TO 8
C
C      SECOND RUN.
C
7      IB2(1)=MR
      N0=1
      NST=5
      MR1=31
C
8      DO 9 I=1,3
      IJ=N0+I
9      IB2(IJ)=0
      NR1=4*MR
C
C      CONVERT TO A REAL*4 NUMBER.
C
      DO 10 I=1,NR1
      II=NST+2*(I-1)
      CALL CV(IB2(II),RES(I))
10     CONTINUE
C
C      IS THERE MORE THAN ONE BLOCK TO WRITE?
C
      NOW=256
      IF(MR.GT.MR1) NOW=512
C
C      WRITE ON THE FILE.
C
      IL=IWRTW(NOW,IB2,IR,IC)
      IF(IL.GE.0) RETURN
      WRITE(7,140)
140    FORMAT(1X,'AN ERROR OCCURED DURING WRITING ON FILE')
      RETURN
      END
```

NAME: RNOD

SUBTITLE: SETS A NODE HEAD

LANGUAGE: FORTRAN IV

IMPLEMENTOR: CHRISTER LARSSON, HARRIET LINDGREN

DATE: 28-APR-78

INSTITUTE:

DEPARTMENT OF AUTOMATIC CONTROL  
LUND INSTITUTE OF TECHNOLOGY, SWEDEN

ACCEPTED:

VERSION:

PURPOSE:

=====

A NODE HEAD IS SET IN IN THE VECTOR IV, BY INFORMATION ABOUT PERIOD,  
INPUT- AND OUTPUT SIGNALS.

USAGE:

=====

PROGRAM TYPE: SUBROUTINE

ARGUMENTS:

RNOD( ID, N, IP1, IP2 )

ID            --NODE TYPE

N             --NODE NUMBER

IP1           --POINTER TO THE FIRST WORD OF THE NODE'S PARAMETERAREA  
              IN VECTOR IPF IN FOREGROUND JOB

IP2           --POINTER TO THE LAST WORD OF THE NODE'S PARAMETERAREA

COMMON BLOCKS:

FG            --CONTAINING ARRAYS FOR SENDING AND RECIEVING MESSAGES  
              BETWEEN THE JOBS

FRAN          --CONTAINING INFORMATION ABOUT PERIOD, NUMBER OF INPUT-  
              AND OUTPUT SIGNALS

CHARACTERISTICS:

=====

SIZE:

-----

PDP11/03: 86











```
C
C
C
C
C      PROGRAM
C
C
C      SUBROUTINE TIME(TT,PS,PA)
C
C      COMMON/FORE/ LF(240)
C      COMMON/FL/  I1,KFL,ISC
C
C      DIMENSION IV(10)
C
C      KFL=1
C      PA=PA+1.
C
C      TRANSIENT TIME EQUAL ZERO?
C
C      IF(TT.LT..5.AND.PA.LT..5) GO TO 5
C      IF(PA.GT.0.5) GO TO 1
C      LF(26)=IFIX(TT)-1
C      RETURN
1     IF(PA.GT.1.5) GO TO 2
C      LF(26)=1
C      RETURN
2     IF(PA.GT.2.5) GO TO 3
C
C      START COLLECTION
C
C      LF(26)=IFIX(PS)-1
C      ISC=1
C      RETURN
3     IF(PA.GT.3.5) GO TO 4
C
C      DEACTIVATION OF FOREGROUND NODES.
C
C      LF(1)=-1
C      LF(2)=-LF(2)
C      LF(14)=-LF(14)
C      LF(13)=-1
C      LF(26)=1
C      RETURN
C
C      DEACTIVATION OF TIME3.
C
4     LF(25)=-1
C      LF(26)=-LF(26)
C
C      STOP COLLECTION.
C
C      ISC=2
C      RETURN
5     LF(26)=IFIX(PS)-1
C      PA=PA+2.
C
C      START COLLECTION.
C
C      ISC=1
C      RETURN
C      END
```



C  
C SIZE:  
C -----  
C PDP11/03: 268  
C  
C

C  
C  
C PROGRAM:  
C =====

C  
C SUBROUTINE WAIT

C  
C LOGICAL\*1 NEW

C  
C COMMON/CRL/ MPA, MCO, NYF, MR, W, SG, MPE, AM  
C COMMON/FFLAG/ IFAN  
C COMMON/FRAN/IPER, IWI(10), IWO(10)  
C COMMON/RP/ KA, IWP(10), FIW(10)  
C COMMON/WRK/ IW1(50), IW2(50), IW3(50), IW4(50)

C  
C  
C WRITE(7,1)  
1 FORMAT(1X,'THE SYSTEM IS NOW WAITING FOR YOU TO DECIDE '/1X,'WHEN  
\* IT IS TIME FOR A NEW FREQUENCY,'/1X,'IN THE MEANWHILE:')

C  
C  
C 2 WRITE(7,3)  
3 FORMAT(1X,'DO YOU WANT TO CHANGE THE AMPLITUDE  
\* OF THE OUTPUT SIGNAL?')

C  
C  
C 4 READ(5,4) NEW  
FORMAT(A1)

C  
C  
C IF(NEW.NE.89) GO TO 7  
C WRITE(7,5)  
5 FORMAT(1X,'AMPLITUDE:')  
C READ(5,6) AM  
6 FORMAT(F10.0)  
GO TO 9

C  
C  
C 7 IF(NEW.EQ.78) GO TO 10  
C WRITE(7,8)  
8 FORMAT(1X,'?ILL CMD?')  
GO TO 2

C  
C  
C SET THE AMPLITUDE OF THE SINUS SIGNAL.

C  
C  
C 9 IW4(1)=2  
IW4(2)=1  
IW4(5)=46  
IW4(6)=1  
IFAN=1  
IPER=1  
IWI(1)=0  
IWO(1)=1  
IWO(2)=1  
KA=1  
IWP(1)=1  
FIW(1)=AM  
CALL OPEN

C  
C  
C RESET THE FLAG.

```
C      IFAN=0
C
10     WRITE(7,11)
11     FORMAT(1X,*, 'WHEN YOU THINK IT IS TIME FOR A NEW FREQUENCY
      * ANSWER G(G=60):' )
      READ(5,4) NEW
      IF(NEW.EQ.71) GO TO 12
      WRITE(7,8)
      GO TO 10
12     CONTINUE
      RETURN
      END
```