

SIMULERING AV PRODUKTIONSSYSTEM

ERIK AXLER

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Maj 1977

Dokumentutgivare
Lund Institute of Technology
Handläggare Dept of Automatic Control
Gustaf Olsson
Författare
Erik Axler

Dokumentnamn
REPORT LUTFD2/(TFRT-5196)/1-67/(1977)
Utgivningsdatum
May 1977
Dokumentbeteckning
Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel

18T0

Simulering av produktionssystem (Simulation of a production system)

Referat (sammandrag)

The purpose of this report has been to find suitable methods and programs for simulation of particularly production flow systems. Only few methods of this subject have been developed, and therefore a considerable time has been used to find basic facts. The problem has been attacked with two different methods. The first one is a simulation program which has all decision rules mathematically defined in the program. The number of decisions increased considerably and troublesome rules in the control of the system resulted. It was difficult to understand the results and the user did not feel that he controlled the system. The other program leaves all the decision rules to the user. When a worker shall be moved, the program writes a report of actual values, from this information the user makes his decision. The two programs are used in different situations. The first one is suitable for testing different control algorithms for a production system. The second one gives the user a better feeling of the system dynamics.

Referat skrivet av

author

Förslag till ytterligare nyckelord

44T0

Klassifikationssystem och -klass(er)

S0T0

Indextermer (ange källa)

S2T0

Omfång

67T0

Övriga bibliografiska uppgifter

56T2

Språk

Swedish

Sekretessuppgifter

60T0

ISSN

60T4

ISBN

60T6

Dokumentet kan erhållas från

Department of Automatic Control
Lund Institute of Technology
P.O. Box 725, S-220 07 LUND 7, Sweden

Mottagarens uppgifter

62T4

Pris

66T0

DOKUMENTATABLAD enligt SIS 62 10 12

SIS-
DB 1

Blankett LU 11:25 1976-07

SIMULERING AV PRODUKTIONSSYSTEM.

Erik Axler

SAMMANFATTNING.

Målsättningen med detta examensarbete har varit att få fram lämpliga metoder och program för att kunna simulera olika produktionssystem, främst flödesorienterade. Detta är ett relativt outforskat område varför avsevärd tid har ägnats åt att få fram grunder.

Problemet har angripits med två olika metoder. Först skrevs ett simuleringsprogram där alla beslut fanns matematiskt definerade i programmet. Antalet beslutsvarianter blev stort vilket ledde till krångliga styrlagar. Resultaten blev svårtolkade och användaren kände inte att det var han som styrde systemet.

Den andra metoden innebär att allt styrande överläts åt programanvändaren. När en operatör skall flyttas ges en rapport ur vars data användaren skall fatta beslut. Metoderna har olika användningsområden. Den första är mest lämplig till att testa ut olika styrlagar medan den andra ger användaren känsla för systemets dynamik.

Ansvarig handledare: Gustav Olsson

SIMULATION OF A PRODUCTION SYSTEM.

Erik Axler

ABSTRACT.

The purpose of this report has been to find suitable methods and programs for simulation of particularly production flow systems.

Only few methods of this subject have been developed, and therefore a considerable time has been used to find basic facts

The problem has been attacked with two different methods. The first one is a simulation program which has all decision rules mathematically defined in the program. The number of decisions increased considerably and troublesome rules in the control of the system resulted. It was difficult to understand the results and the user didn't feel that he controlled the system.

The other program leaves all the decision rules to the user. When a worker shall be moved, the program writes a report of actual values, from this information the user makes his decision.

The two programs are used in different situations. The first one is suitable for testing different control algorithms for a production system. The second one gives the user a better feeling of the system dynamics.

Supervisor: Gustav Olsson

INNEHÅLLSFÖRTECKNING.

1. Inledning	1
2. Problemställning	2
3. Förutsättningar automatiska beslutsregler	3
4. Automatiska beslutsregler	4
5. Programbeskrivning vid automatiska beslutsregler	5
6. Simuleringsresultat vid automatiska beslutsregler	6
7. Slutsatser av automatiska beslutsregler	7
8. Interaktiv simulering	8
9. Vad kan utföras av det interaktiva programmet	9
10. Programbeskrivning interaktivt program	10
11. Resultat av interaktivt program	13
12. Slutsatser	14
13. Referenser	15
Appendix	
A. Program med automatiska beslutsregler	
B. Indata till simulering vid Alfa-Laval	
C. Simuleringsresultat Alfa-Laval	
D. Interaktivt program	
E. Manual till interaktivt program	
F. Körning av interaktivt program	

1. INLEDNING.

Den vanligaste organisationsformen i verkstäder är i dag av funktionell typ d.v.s. likartade maskiner placeras i grupper. Denna typ av produktion är väl utforskad och en rad teorier om t.ex. köpprioritering (Wild) har utvecklats. Dessa verkstäder har en del nackdelar bl.a. binder de stort kapital i produkter i arbete. Vidare har de långa och osäkra genomloppstider.

För att minska dessa nackdelar har en typ av produktion som bygger på så kallade flödesgrupper utvecklats. För att detta skall vara möjligt krävs att produktfloran innehåller gärna många produkter men produkter som är likartade produktionsmässigt.

För att undersöka om produktfloran uppfyller kraven kan ett gruppteknologiskt (Datorstödd produktion) tankesätt vara nyttigt.

I en flödesgrupp utförs arbetet av ett arbetslag, som tillsammans utför ett antal arbetsoperationer på produkten. Gruppen styrs ej av någon förman utan ansvarar själva för produktionen. För att kunna utföra ett effektivt arbete krävs att arbetsledning och gruppmedlemmar är väl insatta i problemen vid tillverkningen.

2. PROBLEMSTÄLLNINGEN.

Som beskrivs i inledningen är produktion med flödesgrupper en relativt ny företeelse. Detta innebär vid planerandet och införandet av nya grupper att det kan vara av stort värde att genom simuleringar få kännedom om sådana saker som lämpligt antal operatörer, förväntad produktion och systemets stabilitet.

Målet är att skapa ett program för simulering av flödesgrupper som är möjligt att använda och som kan ge utbyte även till den som ej är programmerare.

Vad som skall simuleras är en allmän produktionslinje där produkter-~~na~~na vandrar från första till sista maskin i turordning. Produkterna skall ~~ej behöva~~ bearbetas i alla stationerna. De stationer där de ej bearbetas skall de bara passera förbi, men med bibehållen produktionsordning.

Stationerna kan betjäna av ett antal operatörer. Vilka antal opera-~~torer~~törer som är tillåtet bestäms av produkten.

Produktionstiderna kommer att variera en hel del. Beroende t.ex. på operatörernas individuella skicklighet kan slumpmässiga variationer i tillverkningstid förekomma. Andra störningar som t.ex. förekommer också.

3. FÖRUTSÄTTNINGAR AUTOMATISKA BESLUTSREGLER.

För att ett produktionssimuleringsprogram skall kunna exekveras helt utan att användaren ingriper måste det finnas automatiska beslutsregler för hur operatörerna skall flyttas mellan maskinerna. Vid skrivandet av det första simuleringsprogrammet som härafter kallas Alfa-1 gjordes vissa förutsättningar.

Maskinerna kan rent generellt ha från noll till sex operatörer. Vad som är tillåtet vid en viss produkt framgår av behandlingstiderna. Det antal operatörer för vilka tiderna är större än noll är tillåtna. Se t.ex. behandlingstiderna i app. B.

En annan förutsättning som antogs var att hoppet mellan tillåtet antal operatörer ej var större än två. Med detta menas, att kunde det finnas två operatörer vid en maskin fick nästa tillåtna åtna antal ej vara fem utan tre eller fyra. Utan denna begränsning hade styrlagarna blivit mycket mer komplexa.

Vidare antogs att alla operatörer kunde arbeta vid alla maskinerna och att vid en flyttning ingen tid gick åt till att repetera kunskaperna.

Körlängden antogs vara den styrvariabel som styrde omfördelningen av operatörer.

Deterministiska bearbetningstider antogs vidare vilket är en klar förenkling av verkligheten. Det går lätt att låta bearbetningstiderna variera stokastiskt i simuleringarna. Utvärderingarna av simuleringarna är dock besvärliga att göra enkelt och entydigt. Det är dessutom troligt att inverkan på produktionen är ganska liten eftersom störningarna jämnar ut sig. Intressantare är då att t.ex. anta att tredje maskins tider är satta tio procent för långsamma och titta på vilken effekt detta har.

4. AUTOMATISKA BESLUTSREGLER.

Det primära för om en maskin kan arbeta är:

1. Att den har något att göra.
2. Att den har ett tillåtet antal operatörer.
3. Att efterföljande maskins inkö ej är för lång.

Är dessa villkor ej uppfyllda måste programmet styra så att de blir uppfyllda eller får maskinen stanna. I Alfa-1 används två procedurer. Den ena används när en maskin vill skaffa operatörer och den andra när en maskin gör sig av med operatörer.

Proceduren som skaffar operatörer (Getholdof) lyckas inte jämnt. Resultatet beror på hur gärna maskinen vill ha operatörerna. Däremot är den andra proceduren (Getridof) tvingande. Lyckas den ej fördela operatörerna bland maskinerna får de stå överksamma.

Den variabel som styr operatörernas fördelning är kölängderna framför maskinerna. För att kunna prioritera vissa maskiner multipliceras kölängden med en maskinprioritering.

När en maskin vill bli av med en operatör sker följande. Man söker igenom hela kedjan av maskiner, förutom den som vill bli av med operatören. Operatören tilldelas den maskin som har högsta värdet på maskinprioritering gånger kölängden. Dock måste man se till att efterföljande maskin inte har för lång inkö samtidigt som en extra operatör tillåtes.

Om man skall fördela två operatörer uppstår problemet om båda skall ges till en maskin eller om de skall delas upp på två maskiner. Detta löses genom att summa prioritet av att ge två en vägs via en konstant mot att ge en två.

När en maskin söker operatörer uppstår ännu ett problem. Hur gärna vill han ha operatörerna? Detta fås av kölängd gånger en ny maskinberoende konstant. Sedan genomsöks de övriga maskinerna på den som lättast kan bli av med en operatör d.v.s. har lägst värde på kölängden gånger en tredje maskinberoende konstant. Dessutom sänks värdet ytterliggare om maskinen har svårt att arbeta p.g.a. efterföljande maskin.

Slutligen jämförs hur mycket den sökande maskinen önskar operatören jämfört hur lätt den andra kan avvara operatören. Är den förstnämnda störst får han operatören av den andra.

Svårare blir det när man söker två. Skall man då ta båda från samma eller en från två? Detta vägs med konstanter och slutligen jämförs den lägsta mot den sökande.

5. PROGRAMBESKRIVNING VID AUTOMATISKA BESLUTSREGLER.

Programmet Alfa-1 är skrivet i Simula. Simula är speciellt lämpat för händelsestyrda simuleringar då det i sig innehåller rutiner för köhantering. Programmet är händelsestyrt. Med händelsestyrt menas att tiden rör sig i steg framåt. Stegen inträffar i Alfa-1 när en maskin blir klar med en produkt. Då aktiveras maskinen och kan genom styrlagarna i sin tur aktivera andra maskiner. Som ovan antytts innehåller Alfa-1 en produktionslinje i vilken det står ett inlager följt av en rad maskiner. Framför varje maskin finns det en inkö för produkter. Produkterna passerar varje maskin i turordning. Alla produkter har inte behandlingstid i alla stationer.

För att en station skall kunna arbeta krävs som redovisats i kap. 4 tre saker. Något att göra, rätt antal operatörer och att nästa maskins inkö ej hindrar.

När ej önskvärda tillstånd dyker upp försöker styrprocedurerna rätta till detta. Lyckas detta ej måste maskinen stanna.

När den sedan skall starta igen uppstår problemet vem som skall starta den. Närmast efterföljande eftersom det kanske är dess kö som har stoppat? Vem startar sista maskin då? Det kan t.ex. näst sista maskin men om både sista och näst sista står stilla vem startar då? I programmet startas en viss maskin av den som ligger närmast före i sekvens. Vidare startar näst sista maskin i produktionslinjen sista maskin. Om de två nästkommande maskinerna står stilla startar under vissa villkor framförvarande maskin efterföljande.

Programmet ger vidare en beskrivning av systemet vid simuleringens början en kontinuerlig utmatning av data under exekveringen plus en slutrapport. Programmet finns app. A

6. SIMULERINGSRESULTAT VID AUTOMATISKA BESLUTSREGLER.

Resultaten som här presenteras är samtliga körningar på en flödesorienterad grupp på Alfa-Laval i Lund.

Simuleringen sträcker sig över en treveckors period. Produktionsprogrammet är detsamma i alla simuleringarna. Konstanter som hålls lika under alla simuleringarna och produktionsprogrammet finns i app. B. Alla simuleringsresultat finns i app. C.

I den första simuleringen har alla maskiner samma värden på konstanter som ångar hur lätt en maskin får och ger operatörer.

Detta ger som resultat att 65 produkter passerade systemet med en genomsnittlig genomloppstid av 50 timmar. 92 överksamma timmar uppstod p.g.a att programmet ej lyckades fördela alla operatörerna. Den andra simuleringen presenterar ett försök till god viktning av maskinerna. Vikterna har provats ut genom ett antal simuleringar.

Här har det passerat 70 produkter med en genomloppstid på 41 timmar. Överksamma mantimmar är 17. En klar förbättring alltså. 17 timmar av totalt 1200 är en förlust på 1,5 procent.

Vilket är då lämpligt antal operatörer? Här ovan har använts tio. Hur fungerar det med nio eller elva?

Om vi tittar på procenten överksamma mantimmar är den för nio operatörer 1,1% för tio 1,4% och för elva 11,2%.

Som synes en obetydlig skillnad mellan nio och tio medan för elva sker det en kraftig stegring. Första maskinen är här styrande och det finns inte tillräckligt arbetsinnehåll i följande operationer för att sysselsätta elva operatörer.

För att ytterligare minska antalet förlorade mantimmar skrevs en procedur som när en maskin blev stoppad av efterföljande inte stannade direkt utan sökte igenom den egna kön för att se om det fanns några produkter som bara skulle behandlas i denna stationen och sedan var klar. Dessa, om det fanns några, behandlades först och när de var klara lämnade de systemet. P.

Programmet blev effektivare se app. C men sådana omplaceringar av produkterna går ej att genomföra i verkligheten då systemet på Alfa-Laval är spårbundet.

7. SLUTSATSER AV AUTOMATISKA BESLUTSREGLER.

Som har framgått ur tidigare kapitel blir antalet konstanter plus antalet varianter på förflyttningar som måste täckas upp stort. Även om det i Alfa-1 kanske används onödigt många maskinprioriterande konstanter kvarstår problemet, hur man väljer lämpliga värden på dessa. Att ur nuvarande utstrifter se slutet app. C för typexempel dra några logiska slutsatser är svårt. Önskvärt vore att man ur en körning där alla prioriteras lika enkelt kunde få fram hyfsade värden på konstanterna.

Den simulering jag kallat optimal ger bra värden men har tagit lång tid och ett antal körningar att få fram.

Vidare är det olyckligt med alla varianter där man tar och ger olika antal operatörer. Användaren känner sig styrd av systemet i stället för tvärtom. En lämpligare metod är antagligen att systemet fattar rutinbesluten medan användaren går in i de besvärligare situationerna. Detta kräver interaktiv körning på terminal. En annan sak som vållade stora problem vem som startar en stoppad maskin. Först användes efterföljande maskin men efter ett tag upptäcktes att ingen startade sista maskin om den stannat. Då fick näst sista maskin starta sista. Stod både sista och näst sista stilla fungerade det ändå inte. Då ändrades programmet så att om de två nästföljande maskinerna till en maskin stod stilla startade denna maskinen under vissa vilkor efterföljande maskin.

Detta träcklande är mycket olyckligt. Här borde något med överblick för hela systemet gripa in istället för att maskinerna startar varandra kors och tvärs

Slutligen är det en risk att programmet är lite väl anpassat till en specifik situation och inte tillräckligt flexibelt så att det klarar andra produktionslinjer.

8. INTERAKTIV SIMULERING.

Ur erfarenheterna av programmet med automatiska beslutsregler utvecklades ett frågesvar program avsett att köras på terminal. Programmeringsspråket är också här Simula.

Programmet med de automatiska beslutsreglerna gav användaren ingen känsla av att vara den som styr simuleringen.

Programmet hade en snårskog av viktningar och konstanter. Efter en simulering kommer det ut sida upp och ner med data som ingen orkar sätta sig in i. Så när man ändrar viktningarna sker det inte på grundval av logiska beslut.

Det interaktiva programmet är motsatsen till detta. Stor möda har lagts ner på att få programmet användarvänligt. Användaren känner hela tiden att det är han som styr.

Vidare är programmet tänkt att vara generellt i det att nya element och styrmetoder "lätt" skall kunna infogas. Hittills finns den rent manuella styrningen programmerad. I denna flyttas operatörerna av användaren rent manuellt. Detta ger användaren en fin känsla för systemet. Man ser lätt vilka maskiner som hela tiden måste vara igång och vilka som ej är så känsliga.

Programmet är en kombination av händelse- och tidsstyrning.

Användaren får en utskrift av läget plus en möjlighet att förändra det, dels med ett tidsintervall valt av honom själv, dels genom att programmet själv "kör fast" och begär åtgärder av användaren

9. VAD KAN UTFÖRAS AV DET INTERAKTIVA PROGRAMMET.

Programmet är utformat för att via dialog kunna bygga upp och simulera produktionslinjer. Linjen består inte bara av maskiner utan även av inlager och färdiglager.

Ett element som lägger upp parallella flöden skall förhoppningsvis också läggas in i framtiden.

Varje maskin läggs upp ett fält för behandlingstider.

Slutligen läggs produktionsprogrammet in och då är programmet klart för exekvering.

Under exekveringen fattar användaren alla beslut om personal förflyttningar. Han får utskrift om att åtgärder bör vidtagas av systemet, men måste själv fatta besluten. Efter det att han fattat besluten och flyttat operatörerna tar programmet hand om sånt som att ändra bearbetningstider, stanna maskiner där det ej finns några operatörer, osv.

10. PROGRAMBESKRIVNING AV INTERAKTIVT PROGRAM.

Detta kapitel ägnas åt en ingående beskrivning av det interaktiva programmet. Programspråket är Simula men kapitlet är skrivet så att även den utan Simula kunskaper skall ha utbyte av det. Vid noggrannare studier av programmet som finns i app. D hänvisas till boken Simula Begin.

10.1 Programstomme.

I ett försök att göra programmet generellt och utvecklingsbart består varje station av ett "huvud" (rad 466-475) som är lika för alla stationer. Kroppen under huvudet är sedan olika p.g.a. dess funktion. Det finns inlager, färdiglager och maskin. Huvudet innehåller ett namn på stationen, stationens ordningsnummer i linjen och en kö att ha produkterna i. Vidare ett heltal som anger antalet operatörer i stationen samt en pekare på den produkt som just nu behandlas.

10.1.1 Stationen maskin.

Stationen maskin (rad 477-705) är den bearbetande delen av systemet. För varje produkt genomlöper den en cykel enligt följande. Den undersöker att det finns något att göra. Vidare att produkten skall behandlas i denna stationen, om inte skickas den vidare till nästa station. Vidare undersöks om det går att arbeta med det antal operatörer som finns vid maskinen och att nästa maskins inkö ej är för lång. Uppfylls inte dessa villkor anropas procedurer som beskrivs senare.

Är det däremot okey bestäms behandlingstider. Först undersöks hur maskinen är riggad. Är den rätt riggad tilldelas den riggtiden noll annars får den rigga om och den tiden fås ur ett fält som funktion av produktens sort och antalet operatörer.

Bearbetningstiden söks på motsvarande sätt i ett fält som funktion av produktens sort och antalet operatörer. Behandlingstiden är summan av riggtiden och bearbetningstiden.

Därefter börjar maskinen bearbeta produkten vilket i Simula innebär att den ställs i en tidskö och aktiveras när tiden blivit starttiden plus behandlingstiden.

Om den aktiveras innan den tiden p.g.a. styrprocedurerna som behandlas i 10.3 måste vissa saker ske. Först måste det räknas ut hur stor del av produkten som är klar. Sedan testas att antalet operatörer ej förändrats så att jag inte kan arbeta vidare. Därefter beräknas nya behandlingstider då ju antalet operatörer har förändrats. Nu startar behandlingen av resterande delen av produkten.

När produkten är färdigbehandlad skickas den vidare till nästa station, men först görs en utskrift på Printfil4. Detta görs för att man efter simuleringen detaljerat skall kunna analysera vad som hänt.

10.1.2 Stationen inlager.

Inlagret (rad 707-720) är den enhet som levererar produkter till första maskin. Med ett av användaren bestämt tidsintervall undersöker inlagret längden på efterföljande maskins inkö är den mindre än maxkön levereras produkter tills maxkön uppnås.

10.1.3 Stationen färdiglager.

Färdiglager (rad 722-741) tar emot de klara produkterna och placerar dem i sin egen kö. Vidare anges tidpunkten när produkten blir klar.

10.1.4 Produkten.

Produktens (rad 458-464) struktur är enkel i programmet. Produkten har ett namn, som kommer i utskrifterna, vidare en typ som anger vilken riggning som krävs. Riggtiden är alltså noll mellan produkter med samma typ. Sort är en tredje variabel som är specifik för produkten och med vars hjälp behandlingstider beräknas. En produkt består av ett antal detaljer. Bunt anger hur många enheter den specifika produkten består av.

10.2 Initialisering.

I programmet finns ett antal procedurer som handhållningskapandet och länkanudet av produktionslinjen. Dessa procedurer använder ett antal filer benämnda Hjälpfil1-3. Detta är temporärt tillordnade filer. Byggs produktionslinjen upp för första gången asignar programmet själv dessa filer. Vid användning av ett redan inlagt produktionssystem måste användaren själv asignera och flytta ut data till dessa filer. För den fysiska inplaceringen av stationerna och för att ge dem rätt värden på konstanterna svarar Skapasyst (rad 230-274). Skapasyst är interaktivt uppbyggt och svaren läggs ut på Hjälpfil1. Länkasyst (rad 276-319) tar in uppgifter från Hjälpfil1 och utför själva skapandet av systemet. Denna uppdelning har gjorts för att om man vill simulera ett system flera gånger man bara skall behöva bygga upp en gång. Bearbetnings och riggningstider för maskinerna handhas av Skapatid (rad 321-361) och Lankatid (rad 363-392). De fungerar på motsvarande sätt som ovan, d.v.s. man kan spara uppgifterna om man vill. Vid lite större system blir arbetet orimligt stort om man ej sparar uppgifterna utan lägger in dem varje gång. Tiderna finns på Hjälpfil2. Uppgörandet av produktionsprogrammet tillgår så att önskat antal produkter placeras i maskinernas inköer. Skapaprogram (rad 394-427) begär uppgifterna och lägger ut dem på Hjälpfil3. Varifrån Lankaprogram (rad 429-448) begär uppgifterna och utför inplaceringen i köerna. Uppdelningen i så många delar har gjorts därför att till samma produktionslinje vill man ofta ha flera produktionsprogram.

10.3 Styrning av simuleringen.

När inte alla villkor är uppfyllda för att en maskin skall kunna producera måste olika styrlagar gripa in. Här är programmet tänkt att möjliggöra olika alternativ.

Är kön tom anropas en procedur (Haringetattgora)¹ som, beroende på vilken styrlag som gäller, (detta är ej inlagt ännu) anropar olika procedurer. Vid den manuella styrlagen händer inget om maskinen ej betjänas av några operatörer. Annars ges en utskrift och en begäran om att man flytta operatörerna från denna maskin, då de ju ej har något att göra.

Sedan undersöks om maskinen kan arbeta (Kanjarbeta rad 487-512).

Vad som begränsar i det manuella fallet är nästa maskins inkö.

Dessutom måste det finnas tillåtet antal operatörer vid maskinen.

Uppfylls ej detta anropas en procedur (Gorasigavmedop rad 530-542).

Nästa maskins inkö kan ej styras från föregående maskin men är antalet operatörer fel går det att åtgärda

10.4 Rapporter.

Programmet innehåller tre typer av rapporter. Den första (Indatarapport rad 136-190) ger en beskrivning av läget vid simuleringens början. Se app. F för exempel. Att dokumentera bra är viktigt för annars är det omöjligt att upprepa simuleringen.

En rapport (Rapport rad 12-74) se app. F skrivs ut i samband med att användaren skall fatta beslut om omfördelning av operatörer. Rapporten innehåller väsentliga data i en komprimerad form då den ju skrivs ut många gånger.

Den tredje rapporten (Slutrapport rad 81-134) innehåller utnyttjandeprocent av maskinerna och köinnehåll vid simuleringens slut.

1/ rad 514-528

11. RESULTAT AV INTERAKTIV SIMULERING.

Vid interaktiv simulering beror resultaten av en kombination av användarens skicklighet och lämpligheten i programutformningen. Programmet måste på ett överskådligt sätt presentera informationen ur vilket beslutet skall fattas, samtidigt som det tillåter så många och så realistiska beslut som möjligt. Samtidigt får ej antalet frågor vid varje beslutstillfälle bli för stort. I app. F presenteras en komplett körning. I denna körning sker en uppbyggnad av hela produktionslinjen. Detta medför att in-datamängden blir rätt stor vilket i sin tur innebär att en hel del arbete på att samla in information måste ha utförts innan simuleringen påbörjats.

Indatan har stor betydelse och även relativt små fel på behandlingstider kan få stora konsekvenser.

Vad är då ett önskat resultat av simuleringen?

Detta är svårt att säga då ett dåligt resultat kan bero av tre saker. I det första fallet har användaren ej fattat de rätta besluten. Han kan t.ex. ha låtit en kö bli för stor så att styrande maskin har stannats. För det andra begränsar programmet. Tidssteget mellan rapporterna gör att det hinner hända en del som kanske borde följts upp tidigare. För det tredje och kanske viktigaste, när är det själva systemet som begränsar. Detta fallet vill man kunna spåra direkt särskilt om man simulerar en ännu inte byggd linje.

12. SLUTSATSER.

Den interaktiva simuleringen har visatsig ha stora fördelar jämfört med Alfa-1. Den största är den att det är användaren som känner att det är han som styr simuleringen. Han får ut data och med hjälp av dessa skall han fatta beslut. Detta ger en stor processkänedom.

Programmet har utformats så att det skall vara användarvänligt. Detta är dock svårt. För en van användare vill ha helt annan information än den som använder programmet för första gången. Inläringen sker dock snabbt och p.g.a. den relativt låga skrivhastigheten på skrivmaskinterminalerna bör utmatningen hållas liten speciellt i satser som skrivs ut många gånger under en simulering. Det är bättre att ha en utförlig manual istället.

Att köra på terminal är dock rätt dyrt.

På programmets grundstomme är det lätt att bygga vidare. Tänkbara utvidgningar beskrivs nedan.

Stokastiska och/eller systematiska störningar kan i vissa lägen ha stort intresse. Andra typer av stationer t.ex. för att parallellkoppla flöden är en annan naturlig utveckling.

Någon kombination av automatisk och manuell styrning är väl dock det som verkar mest aktuellt. Med detta menas att programmet så länge allt går bra sköter förflyttningen av operatörer men överlåter i de kniviga situationerna besluten åt användaren.

Produktionssimulering är ett mycket stort område som är relativt lite utforskat. En del företag har utvecklat egna simuleringsmetoder t.ex. Alfa-Laval i Lund men något mer systematiskt angrepp har inte gjorts.

Vilken av de två angreppsmetoder som använts i detta examensarbete är då lämpligast? Det går ej generellt att säga. Den första har reglertekniskt större intresse då den ger en möjlighet att analysera och utvärdera olika styrlagar. Konstanterna kan tänkas som regulatorer med vars hjälp man försöker optimera. Här kan olika styr- och målvariabler användas.

Den andra metoden riktar sig till programanvändaren och försöker ge honom de kunskaper och insikter om systemet som kan vara nyttiga att känna till.

Idealet vore att först simulera manuellt så användaren fick kunskap om systemet. Med hjälp av denna kunskap skulle sedan en lämplig stylag väljas och lämpliga värden på konstanterna simuleras fram

REFERENSER.

Graham M Birtwistle, Ole-Johan Dahl, Björn Myhrhaug and Kirsten Nygaard (1973): Simula Begin. Studentlitteratur

Ray Wild (1971): The techniques of production management. Holt Rinehart Winston Ltd

Gunnar Kullberg (1976): Samordnad datorstyrd produktion. Ingenjörslitteratur

Projektarbete i systemteknik. Rapport 7632(C) Inst. för Regler-
teknik Lunds Tekniska Högskola

APPENDIX A

Simuleringsprogram med automatiska beslutsregler.

```

*I.M,INN SYMB,RELOC
RUEGIAN COMPUTING CENTER - EXEC8 SIMULA 76-09-29 A20 0 72
E COMPILATION STARTED ON 77-04-27 AT 15:13:22. ( / 0)
1 BEGIN
2 SIMULATION CLASS ALFA(NROFSTATION, RU, SORT);
3 INTEGER NROFSTATION, RU, SORT;
4 BEGIN
5 COMMENT***NROFSTATION ANGER ANTALET STATIONER DET FINNS I
6 ***PRODUKTIONSLINJEN***;
7 COMMENT***VARJE MASKIN FÅR FRÅN BÖRJAN EN OPERATÖR RU ANGER HUR MÅNGA
8 *** YTTRELLIGARE OPERATÖRER DET FINNS, SEDAN ANGER DET ANTALET
9 *** OVERKSAMMA OPERATÖRER***;
10 COMMENT***SORT ANGER ANTALET OLIKA PRODUKTER***;
11 REAL PASSING, SIMPERIOD, OVERKTID, OVERKSAM;
12 REAL KONST1, KONST2, KONST3, KONST4;
13 REAL ARRAY O, W, VIKT(1:NROFSTATION);
14 REAL ARRAY R, WO(1: SORT, 1:NROFSTATION, 1:6);
15 INTEGER ARRAY ANTBUNT(1:NROFSTATION);
16 INTEGER ANTALOP;
17 INTEGER UTSKRIFT;
18 INTEGER I, L, K, ANTB, ANTPASSERA;
19 REF(MASKIN)ARRAY STATION(1:NROFSTATION);
20 PROCEDURE REPORT;
21 BEGIN
22 COMMENT***PROCEDURE REPORT SKRIVER UT RESULTAT OCH MASKINLAGER VID
23 *** SIMULERINGENS SLUT***;
24 INTEGER II, JJ, KK;
25 OUTIMAGE;
26 OUTTEXT("ANTALET OPERATÖRER");
27 OUTINT(ANTALOP, 3);
28 OUTIMAGE;
29 OUTTEXT("ANTAL BUNTAR GENOM SYSTEMET");
30 OUTINT(ANTPASSERA, 6);
31 OUTIMAGE;
32 OUTTEXT("GENOMSNITTLIG GENOMLOPPSTID");
33 IF ANTB=0 THEN
34 OUTTEXT(" INGEN HAR GENOMLÖPT")
35 ELSE BEGIN
36 I:=PASSING/ANTB;
37 OUTFIX(I, 2, 6);
38 END;
39 OUTIMAGE;
40 OUTTEXT("OVERKSAMMA MANTIMMAR");
41 OVERKTID:=OVERKTID+(TIME-OVERKSAM)*RU;
42 OUTFIX(OVERKTID, 2, 6);
43 OUTIMAGE;
44 FOR I:=1 STEP 1 UNTIL NROFSTATION DO
45 BEGIN
46 INSPECT STATION(I) DO BEGIN
47 OUTTEXT(NAMN);
48 OUTIMAGE;
49 OUTTEXT("STATION NR.");
50 OUTINT(P, 1);
51 OUTIMAGE;
52 OUTTEXT("TIDSUTNYTTJANDEPROCENT=");
53 IF WORK THEN UTNYTPRO:=UTNYTPRO+TIME-STARTTIME;
54 UTNYTPRO:=UTNYTPRO/SIMPERIOD*100;
55 OUTFIX(UTNYTPRO, 2, 6);
56 OUTIMAGE;
57 OUTTEXT("KAPACITETSUTNYTTJANDEPROCENT=");
58 IF WORK THEN KAPUTNYTPRO:=KAPUTNYTPRO+(U/MAXU*(TIME-STARTTIME));
59 KAPUTNYTPRO:=KAPUTNYTPRO/SIMPERIOD*100;
60 OUTFIX(KAPUTNYTPRO, 2, 6);
61 OUTIMAGE;

```

```

62 JJ:=QUEUE.CARDINAL;
63 IF JJ=0 THEN GO TO INGEN;
64 FOR KK:=1 STEP 1 UNTIL JJ DO
65 BEGIN
66 SERVED:=QUEUE.FIRST;
67 OUTTEXT(SERVED.NAMN);
68 OUTINT(SERVED.TYPE,5);
69 OUTINT(SERVED.BUNT,5);
70 OUTIMAGE;
71 SERVED.OUTPUT;
72 END;
73 OUTIMAGE;
74 INGEN:END;END;
75 END***REPORT***;
76
77 COMMENT*****;
78
79 PROCESS CLASS PLATTA(NAMN,TYPE,RIGG,BUNT);
80 TEXT NAMN;INTEGER TYPE,RIGG,BUNT;
81 BEGIN REAL STARTTIME;
82 COMMENT***NAMN AR EN TEXTVARIABEL FOR PRODUKTENS NAMN (20 POS.)***;
83 COMMENT***BUNT ANGER ANTALET PLATTOR SOM INGAR I BUNTEN***;
84 COMMENT***TYP TAL SOM KNYTER PLATTSORT OCH BEHANDLINGSTID***;
85 COMMENT***STARTTIME=TIDEN NAR BUNTEN LAGGS I FORSTE MASKINS INKO***;
86 END***PLATTA***;
87
88 COMMENT*****;
89
90 PROCESS CLASS LAGER(NUMBER,ANT);
91 INTEGER ANT,NUMBER;
92 COMMENT***ANT ANGER ANTALET BUNTAR SOM FINNS I LAGRET VID SIMULERINGENS
93 ***BORJAN***;
94 COMMENT***NUMBER ANGER VID VILKEN STORLEK MAN FORSOKER HALLA FORSTA
95 *** MASKINS INLAGER***;
96 BEGIN INTEGER I;
97 REF(HEAD)QUEUE;
98 REF(PLATTA)SERVED;
99 QUEUE:=NEW HEAD;
100 FOR I:=1 STEP 1 UNTIL ANT DO BEGIN
101 INIMAGE;
102 NEW PLATTA(COPY(INTEXT(20)),ININT,ININT,ININT).INTO(QUEUE);
103 END;
104 OUTTEXT("LAGRETS STORLEK VID SIMULERINGENS BORJAN");
105 I:=QUEUE.CARDINAL;
106 OUTINT(I,5);
107 OUTIMAGE;
108 IF QUEUE.EMPTY THEN GO TO INGEN;
109 SERVED:=QUEUE.FIRST;
110 IGEN:
111 OUTTEXT(SERVED.NAMN);
112 OUTINT(SERVED.BUNT,7);
113 OUTIMAGE;
114 SERVED:=SERVED.SUC;
115 IF NOT SERVED==NONE THEN GO TO IGEN;
116 INGEN:
117 WHILE TIME<SIMPERIOD DO BEGIN
118 IF QUEUE.EMPTY THEN GOTO SLUT;
119 IF STATION(1).QUEUE.CARDINAL<NUMBER THEN BEGIN
120 SERVED:=QUEUE.FIRST;
121 SERVED.STARTTIME:=TIME;
122 SERVED.INTO(STATION(1).QUEUE);
123 END;
124 HOLD(0.5);
125 END;

```

```

126 SLUT:OUTTEXT("*****LAGRET SLUT*****");
127 OUTFIX(TIME/2,6);
128 OUTIMAGE;
129 END***LAGER***;
130
131 COMMENT*****;
132
133 PROCESS CLASS MASKIN(NAMN,P,QUEU,STOPPEDSTART,STOP,U,WANT,DANGER);
134 REF(HEAD)QUEU;
135 TEXT NAMN;
136 INTEGER P,STOPPEDSTART,STOP,U,DANGER,WANT;
137 BEGIN
138 COMMENT***NAMN ANGER TILLVERKNINGSSTATIONENS NAMN (20 POS)***;
139 INTEGER MAXU;
140 INTEGER I,LAST,SLASK,SLAS1;
141 INTEGER IBEFORE,IAFTER;
142 REAL STARTTIME,FINISHTIME,WORKINGTIME,FAKTOR,WORKINGT;
143 REAL UTNYTPRO,KAPUTNYTPRO;
144 REAL PASSINGTIME;
145 BOOLEAN ARRAY B(-2:8);
146 BOOLEAN WORK,DONTWANT,AFT;
147 REF(PLATTA)SERVED;
148 REF(PLATTA)SMITA;
149 REF(PLATTA)PEKA;
150
151 COMMENT*****;
152
153 BOOLEAN PROCEDURE GETHOLDOF(R,OP);
154 INTEGER R,OP;
155 BEGIN
156 COMMENT***GETHOLDOF ANROPAS HAR EN MASKIN VILL ÖKA ANTALET
157     ***OPERATÖRER***;
158 COMMENT***R ANGER VILKEN MASKIN SOM ANROPAR***;
159 COMMENT***OP ANGER HUR MÅNGA OPERATÖRER SOM SÖKES***;
160 INTEGER J,NR1,NP2,CMR;
161 REAL GETHOLD,LOSE1,LOSE2,WANT,CLOSE;
162 REAL LOSE;
163 BOOLEAN FALSK;
164 GETHOLD:=Q(R)*STATION(R).QUEU.CARDINAL;
165 OVERKTID:=OVERKTID+(TIME-OVERKSAM)*RU;
166 IF OP=2 THEN GO TO TWO;
167 IF RU>=1 THEN BEGIN
168     RU:=RU-1;
169     GETHOLDOF:=TRUE;
170     GO TO SLUT;
171 END;
172 LOSE1:=10000;
173 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
174     INSPECT STATION(J) DO BEGIN
175         IF NOT P=R AND B(U-1) AND DONTWANT THEN
176             WANT:=VIKT(J)*QUEU.CARDINAL*KONST2
177         ELSE IF NOT P=R AND B(U-1) THEN
178             WANT:=VIKT(J)*QUEU.CARDINAL ELSE WANT:=10001,0;
179         IF WANT<LOSE1 THEN BEGIN
180             LOSE1:=WANT;
181             NR1:=J;
182         END;
183     END; END;
184     IF LOSE1<GETHOLD THEN BEGIN
185         GETHOLDOF:=TRUE;
186         STATION(NR1).U:=STATION(NR1).U-1;
187         REACTIVATE STATION(NR1) AFTER STATION(R);
188     GO TO SLUT;
189 END;

```

```

190 LOSE1:=10000;
191 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
192 INSPECT STATION(J) DO BEGIN
193 IF NOT P=R AND B(U-2) AND NOT B(U-1) AND DONTWANT THEN
194 WANT:=VIKT(J)*QUEU,CARDINAL*KONST2
195 ELSE IF NOT P=R AND B(U-2) AND NOT B(U-1) THEN
196 WANT:=VIKT(J)*QUEU,CARDINAL
197 ELSE WANT:=10001,0;
198 IF WANT<LOSE1 THEN BEGIN
199 LOSE1:=WANT;
200 NR1:=J;
201 END;
202 END; END;
203 LOSE1:=LOSE1*KONST4;
204 IF LOSE1<GETHOLD THEN BEGIN
205 GETHOLDOF:=TRUE;
206 STATION(NR1),U:=STATION(NR1),U-2;
207 REACTIVATE STATION(NR1) AFTER STATION(R);
208 GETRIDOF(R,1);
209 GO TO SLUT;
210 END;
211 GETHOLDOF:=FALSE;
212 FALSK:=TRUE;
213 GO TO SLUT;
214 TWO:IF RU>=2 THEN BEGIN
215 RU:=RU-2;
216 GETHOLDOF:=TRUE;
217 GO TO SLUT;
218 END;
219 IF RU=1 THEN BEGIN
220 RU:=RU-1;
221 IF GETHOLDOF(R,1) THEN BEGIN
222 GETHOLDOF:=TRUE;
223 GO TO SLUT;
224 END
225 ELSE RU:=RU+1;
226 END;
227 CLOSE:=10000;
228 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
229 INSPECT STATION(J) DO BEGIN
230 IF NOT P=R AND B(U-2) AND NOT B(U-1) AND DONTWANT THEN
231 WANT:=VIKT(J)*QUEU,CARDINAL*KONST2 ELSE
232 IF NOT P=R AND B(U-2) THEN
233 WANT:=VIKT(J)*QUEU,CARDINAL ELSE WANT:=10001,0;
234 IF WANT<CLOSE THEN BEGIN
235 CLOSE:=WANT;
236 CNR:=J;
237 END;
238 END;
239 END;
240 LOSE1:=10000;
241 LOSE2:=10000;
242 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
243 INSPECT STATION(J) DO BEGIN
244 IF NOT P=R AND B(U-1) AND DONTWANT THEN
245 WANT:=VIKT(J)*QUEU,CARDINAL*KONST2 ELSE
246 IF NOT P=R AND B(U-1) THEN WANT:=VIKT(J)*QUEU,CARDINAL ELSE WANT:=10001;
247 IF WANT<LOSE2 THEN BEGIN
248 LOSE1:=LOSE2;
249 LOSE2:=WANT;
250 NR1:=NR2;
251 NR2:=J;
252 END
253 ELSE IF WANT<LOSE1 THEN BEGIN

```



```

254 LOSE1:=WANT;
255 NR1:=J;
256 END;
257 END;
258 END;
259 LOSE:=(LOSE1+LOSE2)*KONST3;
260 IF CLOSE<LOSE AND CLOSE<GETHOLD THEN BEGIN
261 STATION(CNR),U:=STATION(CNR),U-2;
262 REACTIVATE STATION(CNR) AFTER STATION(R);
263 GETHOLDOF:=TRUE;
264 GO TO SLUT;
265 END;
266 IF LOSE<CLOSE AND LOSE<GETHOLD AND LOSE1<9999 AND LOSE2<9999 THEN BEGIN
267 STATION(NR2),U:=STATION(NR2),U-1;
268 STATION(NR1),U:=STATION(NR1),U-1;
269 REACTIVATE STATION(NR2) AFTER STATION(R);
270 REACTIVATE STATION(NR1) AFTER STATION(R);
271 GETHOLDOF:=TRUE;
272 GO TO SLUT;
273 END;
274 GETHOLDOF:=FALSE;
275 FALSK:=TRUE;
276 SLUT:
277 OVERKSAM:=TIME;
278 IF UTSKRIFT=1 THEN
279 BEGIN
280 OUTFIX(TIME,1,6);
281 OUTTEXT(" STATION");
282 OUTINT(R,2);
283 IF NOT FALSK THEN OUTTEXT(" FÄR");
284 ELSE
285 OUTTEXT(" SÖKER MEN FÄR EJ");
286 OUTINT(OP,2);
287 OUTTEXT(" OPERATÖRER");
288 OUTTEXT(" OVERKSAMMA OPERATÖRER=");
289 OUTINT(RU,2);
290 OUTIMAGE;
291 OUTTEXT(" ANT.OP ANT.BUNTAR/");
292 FOR I:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
293 INSPECT STATION(I) DO BEGIN
294 IF P=R AND NOT FALSK THEN
295 BEGIN
296 SLASK:=U+OP;
297 OUTINT(SLASK,3);
298 END
299 ELSE
300 OUTINT(U,3);
301 OUTINT(QUEU,CARDINAL,3);
302 OUTTEXT("/");
303 END;
304 END;
305 OUTIMAGE;
306 END;
307 END***GETHOLDOF***;
308
309 COMMENT*****;
310
311 INTEGER PROCEDURE GETRIDOF(R,OP);
312 INTEGER R,OP;
313 BEGIN
314 COMMENT***GETRIDOF ANROPAS NÄR EN MASKIN VILL MINSKA ANTALET
315 *** OPERATÖRER***;
316 COMMENT***R ANGER ANROPANDE MASKIN***;
317 COMMENT***OP ANGER ANTALET OPERATÖRER***;

```

```

318 INTEGER NR,J, NR1, NR2;
319 REAL PRIO, GET1, GET2, GET, GETTWO;
320 OVERKTID:=OVERKTID+(TIME-OVERKSAM)*RU;
321 IF OP=0 THEN GO TO SLUT;
322 START9:IF OP>=3 THEN BEGIN
323 OP:=OP-2;
324 GETRIDOF(R,2);
325 IF OP>2 THEN GO TO START9;
326 END;
327 IF OP=1 THEN GO TO ONE;
328 GETTWO:=0;
329 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
330 INSPECT STATION(J) DO BEGIN
331 IF NOT R=P AND B(U+2) AND NOT DONTWANT THEN BEGIN
332 PRIO:=W(P)*QUEUE,CARDINAL;
333 IF PRIO>GETTWO THEN
334 BEGIN
335 GETTWO:=PRIO;
336 NR:=J;
337 END;
338 END;
339 END; END;
340 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
341 INSPECT STATION(J) DO BEGIN
342 IF NOT R=P AND B(U+1) AND NOT DONTWANT THEN BEGIN
343 PRIO:=W(P)*QUEUE,CARDINAL;
344 IF PRIO>GET2 THEN BEGIN
345 GET1:=GET2;
346 GET2:=PRIO;
347 NR1:=NR2;
348 NR2:=J;
349 END
350 ELSE IF PRIO>GET1 THEN
351 BEGIN
352 NR1:=J;
353 GET1:=PRIO;
354 END;
355 END;
356 END;
357 END;
358 GET:=(GET1+GET2)*KONST1;
359 IF GET>GETTWO AND GET1>0.1 AND GET2>0.1 THEN
360 BEGIN
361 STATION(NR1),U:=STATION(NR1),U+1;
362 STATION(NR2),U:=STATION(NR2),U+1;
363 REACTIVATE STATION(NR1) AFTER STATION(R);
364 REACTIVATE STATION(NR2) AFTER STATION(R);
365 GO TO SLUT;
366 END;
367 IF GET>GETTWO AND GET2>0.1 THEN
368 BEGIN
369 STATION(NR2),U:=STATION(NR2),U+1;
370 REACTIVATE STATION(NR2) AFTER STATION(R);
371 RU:=RU+1;
372 GO TO SLUT;
373 END;
374 IF GETTWO>GET AND GETTWO>0.1 THEN
375 BEGIN
376 STATION(NR),U:=STATION(NR),U+2;
377 REACTIVATE STATION(NR) AFTER STATION(R);
378 GO TO SLUT;
379 END;
380 RU:=RU+2;
381 GO TO SLUT;

```

```

382 ONE: GET1:=0,0;
383 FOR J:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
384 INSPECT STATION(J) DO BEGIN
385 IF NOT R=P AND B(U+1) AND NOT DONTWANT THEN
386 BEGIN
387 PRIO:=W(P)*QUEU.CARDINAL;
388 IF PRIO>GET1 THEN
389 BEGIN
390 GET1:=PRIO;
391 NR:=J;
392 END;
393 END;
394 END;
395 END;
396 IF GET1>0,1 THEN
397 BEGIN
398 STATION(NR),U:=STATION(NR),U+1;
399 REACTIVATE STATION(NR) AFTER STATION(R);
400 GO TO SLUT;
401 END;
402 RU:=RU+1;
403 IF RU>=2 THEN
404 BEGIN
405 RU:=RU-2;
406 GETRIDOF(R,2);
407 END;
408 SLUT: BEGIN
409 OVERKSAM:=TIME;
410 IF UTSKRIFT=1 THEN
411 BEGIN
412 OUTFIX(TIME,1,6);
413 OUTTEXT(" STATION");
414 OUTINT(R,2);
415 OUTTEXT(" LÄMNAR");
416 OUTINT(OP,2);
417 OUTTEXT(" OPERATÖRER");
418 OUTTEXT(" OVERKSAMMA OPERATÖRER=");
419 OUTINT(RU,2);
420 OUTIMAGE;
421 OUTTEXT(" ANT.OP ANT.BUNTAR/");
422 FOR I:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
423 INSPECT STATION(I) DO BEGIN
424 OUTINT(U,3);
425 OUTINT(QUEU.CARDINAL,3);
426 OUTTEXT("/");
427 END;
428 END;
429 OUTIMAGE;
430 END;
431 END;
432 END***GETRIDOF***;
433
434 COMMENT*****;
435
436 B(1):=TRUE;
437 B(2):=TRUE;
438 START:IF QUEU.EMPTY THEN BEGIN
439 IF U>0 THEN
440 BEGIN
441 SLASK:=U;
442 U:=0;
443 GETRIDOF(P,SLASK);
444 END;
445 DONTWANT:=FALSE;

```

```

446 PASSIVATE;
447 GO TO START;
448 END;
449 FAKTOR:=0;
450 SERVED:=QUEU,FIRST;
451 START12:
452 SLAS1:=0;
453 FOR I:=1 STEP 1 UNTIL: 6 DO
454 BEGIN
455 IF WO(SERVED,TYPE,P,I)>0.01 THEN
456 BEGIN B(I):=TRUE; SLAS1:=SLAS1+1; END
457 ELSE B(I):=FALSE;
458 END;
459 START1:
460 IBEFORE:=P-1;
461 IF NOT P=1 AND NOT STATION(P-1).WORK THEN
462 BEGIN
463 STATION(IBEFORE).DONTWANT:=FALSE;
464 REACTIVATE STATION(IBEFORE) AFTER STATION(P);
465 END;
466 IAFTER:=P+1;
467 IF P=NROFSTATION THEN AFT:=TRUE;
468 IF NOT AFT AND STATION(IAFTER).QUEU.CARDINAL>STATION(IAFTER).STOP THEN
469 BEGIN
470 DONTWANT:=TRUE;
471 IF U>0 THEN
472 BEGIN
473 SLASK:=U;
474 U:=0;
475 GETRIDOF(P,SLASK);
476 END;
477 PASSIVATE;
478 END;
479 START14:
480 SLASK:=P+1;
481 IF SLASK=NROFSTATION AND NOT STATION(IAFTER).WORK THEN
482 REACTIVATE STATION(IAFTER) AFTER STATION(P);
483 IF P<=NROFSTATION-2 AND U>=1 AND NOT STATION(P+1).WORK AND
484 NOT STATION(P+2).WORK AND STATION(P+1).QUEU.CARDINAL>=1 THEN
485 REACTIVATE STATION(P+1) AFTER STATION(P);
486 IF NOT AFT AND STATION(IAFTER).QUEU.CARDINAL<STATION(IAFTER).WANT THEN
487 DONTWANT:=FALSE;
488 IF NOT AFT AND STATION(IAFTER).QUEU.CARDINAL>STATION(IAFTER).DANGER THEN
489 DONTWANT:=TRUE;
490 IF SLAS1=0 THEN
491 BEGIN
492 IF UTSKRIFT=1 THEN
493 BEGIN
494 OUTFIX(TIME,1,6);
495 OUTTEXT(" STATION");
496 OUTINT(P,2);
497 OUTTEXT(" HAR INGEN BEHANDLINGSTID UTAN LÄMNAR BUNTEN TILL NÄSTA ");
498 OUTTEXT("STATION");
499 OUTIMAGE;
500 END;
501 IF U>0 THEN
502 BEGIN
503 SLASK:=U;
504 U:=0;
505 GETRIDOF(P,SLASK);
506 END;
507 DONTWANT:=FALSE;
508 GO TO SLUT7;
509 END;

```

```

510 IF RU>=1 AND B(U+1) AND NOT DONTWANT AND GETHOLDOF(P,1) THEN U:=U+1;
511 IF RU>=2 AND B(U+2) AND NOT DONTWANT AND GETHOLDOF(P,2) THEN U:=U+2;
512 IF QUEUE.CARDINAL>DANGER AND NOT DONTWANT AND B(U+1) AND GETHOLDOF(P,1)
513 THEN U:=U+1;
514 IF QUEUE.CARDINAL>DANGER AND NOT DONTWANT AND NOT B(U+1) AND B(U+2)
515 AND GETHOLDOF(P,2) THEN U:=U+2;
516 IF NOT B(U) AND B(U+1) AND NOT DONTWANT AND GETHOLDOF(P,1) THEN U:=U+1;
517 IF NOT B(U) AND B(U+2) AND NOT DONTWANT AND GETHOLDOF(P,2) THEN U:=U+2;
518 IF NOT B(U) AND B(U-1) THEN
519 BEGIN
520 U:=U-1;
521 GETRIDOF(P,1);
522 END;
523 IF NOT B(U) AND B(U-2) THEN
524 BEGIN
525 U:=U-2;
526 GETRIDOF(P,2);
527 END;
528 IF NOT B(U) THEN BEGIN
529 IF U>0 THEN
530 BEGIN
531 SLASK:=U;
532 U:=0;
533 GETRIDOF(P,SLASK);
534 END;
535 DONTWANT:=FALSE;
536 PASSIVATE;
537 GO TO START1;
538 END;
539 START2:
540 IF NOT SERVED.RIGG=LAST THEN
541 WORKINGTIME:=R(SERVED.TYPE,P,U) ELSE WORKINGTIME:=0.0;
542 WORKINGTIME:=WORKINGTIME+W0(SERVED.TYPE,P,U)*SERVED.BUNT*0.01;
543 WORKINGT:=WORKINGTIME*(1-FAKTOR);
544 WORK:=TRUE;
545 STARTTIME:=TIME;
546 IF UTSKRIFT=1 THEN
547 BEGIN
548 OUTFIX(TIME,1,6);
549 OUTTEXT("  STARTAR STATION");
550 OUTINT(P,2);
551 OUTTEXT("  ");
552 OUTTEXT(SERVED.MAMN);
553 OUTIMAGE;
554 END;
555 FOR I:=1 STEP 1 UNTIL 6 DO BEGIN
556 IF B(I) THEN MAXU:=I; END;
557 HOLD(WORKINGT);
558 WORK:=FALSE;
559 UTNYTPRO:=UTNYTPRO+TIME-STARTTIME;
560 KAPUTNYTPRO:=KAPUTNYTPRO+(U/MAXU*(TIME-STARTTIME));
561 FINISHTIME:=STARTTIME+WORKINGT-0.001;
562 IF TIME<FINISHTIME THEN
563 BEGIN
564 FAKTOR:=(TIME-STARTTIME)/WORKINGT+FAKTOR;
565 GO TO START2;
566 END;
567 LAST:=SERVED.RIGG;
568 SLUT7:
569 IF P=NROFSTATION THEN GO TO SLUT1;
570 SLASK:=P+1;
571 IGEN:
572 IF W0(SERVED.TYPE,SLASK,1)<0.001 AND W0(SERVED.TYPE,SLASK,2)<0.001 THEN
573 BEGIN

```

```

574 IF SLASK=NROFSTATION THEN GO TO SLUT1
575 ELSE BEGIN SLASK:=SLASK+1; GO TO IGEN;
576 END;
577 END;
578 SERVED.INTO(STATION(P+1),QUEU);
579 GO TO START;
580 SLUT1:
581 OUTTEXT("
582 OUTTEXT(" ");
583 OUTFIX(TIME,1,6);
584 OUTTEXT(" ");
585 OUTTEXT(SERVED.NAMN);
586 OUTINT(SERVED.BUNT,6);
587 IF SERVED.STARTTIME<0,001 THEN
588 OUTTEXT(" INGEN STARTTID")
589 ELSE BEGIN
590 PASSINGTIME:=TIME-SERVED.STARTTIME;
591 OUTFIX(PASSINGTIME,2,6);
592 PASSING:=PASSING+PASSINGTIME;
593 ANTB:=ANTB+1;
594 END;
595 OUTIMAGE;
596 ANTPASSERA:=ANTPASSERA+1;
597 SERVED.OUT;
598 GO TO START;
599 END***MASKIN***;
600
601 COMMENT*****;
602
603 SIMPERIOD:=INREAL;
604 COMMENT***SIMPERIOD ANGER SIMULERINGENS LÄNGD I TIMMAR***;
605 OUTTEXT("SIMULERINGSLÄNGD=");
606 OUTFIX(SIMPERIOD,2,5);
607 OUTIMAGE;
608 UTSKRIFT:=ININT;
609 COMMENT***UTSKRIFT=1 GER EN KONTINUERLIG UTSKRIFT AV VAD SOM HÄNDER I
610 *** PROGRAMMET ÖNSKAS DETTA EJ SATT UTSKRIFT=0***;
611 KONST1:=INREAL;
612 KONST2:=INREAL;
613 KONST3:=INREAL;
614 KONST4:=INREAL;
615 COMMENT***KONST1 TILLHÖR GETRIDOF KONST2-4 TILLHÖR GETHOLDOF***;
616 OUTTEXT("KONST1-4");
617 OUTFIX(KONST1,2,5);
618 OUTFIX(KONST2,2,5);
619 OUTFIX(KONST3,2,5);
620 OUTFIX(KONST4,2,5);
621 OUTIMAGE;
622 OUTTEXT("STATION NR Q W VIKT");
623 OUTIMAGE;
624 FOR I:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
625 Q(I):=INREAL;
626 W(I):=INREAL;
627 VIKT(I):=INREAL;
628 OUTINT(I,10);
629 OUTFIX(Q(I),2,7);
630 OUTFIX(W(I),2,7);
631 OUTFIX(VIKT(I),2,7);
632 OUTIMAGE;
633 END;
634 OUTTEXT("BEHANDLINGSTID");
635 OUTIMAGE;
636 OUTTEXT(" 1 2 3 4 5 6");
637 OUTIMAGE;

```

```

638 FOR I:=1 STEP 1 UNTIL SORT DO BEGIN
639 FOR L:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
640 OUTTEXT("STATION");
641 OUTINT(L,2);
642 OUTTEXT("  SORT");
643 OUTINT(I,2);
644 FOR K:=1 STEP 1 UNTIL 6 DO BEGIN
645 WO(I,L,K):=INREAL;
646 R(I,L,K):=INREAL;
647 OUTFIX(WO(I,L,K),2,6);
648 END;
649 OUTIMAGE;
650 END;
651 END;
652 COMMENT***WO(I,J,K) BEARBETNINGSTID FÖR EN PLATTA AV TYP I VID STATION
653     ***J OCH ANTALET OPERATÖRER K***;
654 COMMENT***R(I,J,K) RIGGNINGSTID***;
655 FOR I:=1 STEP 1 UNTIL NROFSTATION DO BEGIN
656 INIMAGE;
657 STATION(I):=-NEW MASKIN(COPY(INTEXT(20)),ININT,NEW HEAD,ININT,ININT,
658 ININT,ININT,ININT);
659 K:=ININT;
660 FOR L:=1 STEP 1 UNTIL K DO
661 BEGIN
662 INIMAGE;
663 NEW PLATTA(COPY(INTEXT(20)),ININT,ININT,ININT).INTO(STATION(I),QUEU);
664 END;
665 END;
666 OUTIMAGE;
667 ANTALOP:=RU;
668 FOR I:=1 STEP 1 UNTIL NROFSTATION DO
669 BEGIN
670 INSPECT STATION(I) DO BEGIN
671 ANTALOP:=ANTALOP+U;
672 OUTTEXT("STATION(");
673 OUTINT(P,2);
674 OUTTEXT(") INDATA");
675 OUTIMAGE;
676 OUTTEXT("STOPPEDSTART=");
677 OUTINT(STOPPEDSTART,2);
678 OUTTEXT("  STOP=");
679 OUTINT(STOP,2);
680 OUTTEXT("    U=");
681 OUTINT(U,2);
682 OUTTEXT("    WANT=");
683 OUTINT(WANT,2);
684 OUTTEXT("    DANGER=");
685 OUTINT(DANGER,2);
686 OUTIMAGE;
687 OUTTEXT("INKÖNS LÅNGD=");
688 SLASK:=QUEU.CARDINAL;
689 OUTINT(SLASK,5);
690 OUTIMAGE;
691 IF QUEU.EMPTY THEN GO TO PASSERA;
692 PEKA:=-QUEU.FIRST;
693 IGEN10;
694 OUTTEXT(PEKA.NAMN);
695 OUTINT(PEKA.BUNT,7);
696 OUTIMAGE;
697 PEKA:=-PEKA.SUC;
698 IF NOT PEKA==NONE THEN GO TO IGEN10;
699 PASSERA;
700 END; END;
701 ACTIVATE NEW LAGER(ININT,ININT);

```

A12.

```
702 FOR I:=1 STEP 1 UNTIL NROFSTATION DO ACTIVATE STATION(I);
703 HOLD(SIMPERIOD);
704 REPORT;
705
706 END***ALFA***;
707
708 NEW ALFA(ININT,ININT,ININT);
709 END;
```


BEHANDLINGSTID

			1	2	3	4	5	6
STATION 1	SORT 1	1	1,17	0,00	0,00	0,00	0,00	0,00
STATION 2	SORT 1	1	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 1	1	5,17	2,67	1,83	1,33	1,03	0,92
STATION 4	SORT 1	1	1,67	0,83	0,67	0,50	0,00	0,00
STATION 5	SORT 1	1	0,00	0,00	0,00	0,00	0,00	0,00
STATION 6	SORT 1	1	1,33	0,00	0,00	0,00	0,00	0,00
STATION 7	SORT 1	1	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 2	2	1,17	0,00	0,00	0,00	0,00	0,00
STATION 2	SORT 2	2	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 2	2	5,17	2,67	1,83	1,33	1,03	0,92
STATION 4	SORT 2	2	1,67	0,83	0,67	0,50	0,00	0,00
STATION 5	SORT 2	2	0,00	1,00	0,00	0,00	0,00	0,00
STATION 6	SORT 2	2	1,33	0,00	0,00	0,00	0,00	0,00
STATION 7	SORT 2	2	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 3	3	1,17	0,00	0,00	0,00	0,00	0,00
STATION 2	SORT 3	3	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 3	3	8,00	4,00	2,67	2,00	1,60	1,33
STATION 4	SORT 3	3	3,33	1,67	1,33	0,83	0,00	0,00
STATION 5	SORT 3	3	0,00	0,00	0,00	0,00	0,00	0,00
STATION 6	SORT 3	3	1,33	0,00	0,00	0,00	0,00	0,00
STATION 7	SORT 3	3	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 4	4	1,17	0,00	0,00	0,00	0,00	0,00
STATION 2	SORT 4	4	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 4	4	8,00	4,00	2,67	2,00	1,60	1,33
STATION 4	SORT 4	4	3,33	1,67	1,33	0,83	0,00	0,00
STATION 5	SORT 4	4	0,00	1,33	0,00	0,00	0,00	0,00
STATION 6	SORT 4	4	1,33	0,00	0,00	0,00	0,00	0,00
STATION 7	SORT 4	4	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 5	5	0,00	1,83	0,00	0,00	0,00	0,00
STATION 2	SORT 5	5	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 5	5	0,00	4,50	0,00	2,33	0,00	1,50
STATION 4	SORT 5	5	0,00	1,83	0,00	0,00	0,00	0,00
STATION 5	SORT 5	5	0,00	0,00	0,00	0,00	0,00	0,00
STATION 6	SORT 5	5	0,00	1,17	0,00	0,00	0,00	0,00
STATION 7	SORT 5	5	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 6	6	0,00	1,83	0,00	0,00	0,00	0,00
STATION 2	SORT 6	6	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 6	6	0,00	4,50	0,00	2,33	0,00	1,50
STATION 4	SORT 6	6	0,00	1,83	0,00	0,00	0,00	0,00
STATION 5	SORT 6	6	0,00	2,33	0,00	0,00	0,00	0,00
STATION 6	SORT 6	6	0,00	1,17	0,00	0,00	0,00	0,00
STATION 7	SORT 6	6	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 7	7	0,00	2,17	0,00	0,00	0,00	0,00
STATION 2	SORT 7	7	0,00	0,00	0,00	0,00	0,00	0,00
STATION 3	SORT 7	7	0,00	4,50	0,00	2,33	0,00	1,50
STATION 4	SORT 7	7	0,00	1,67	0,00	0,83	0,00	0,00
STATION 5	SORT 7	7	0,00	0,00	0,00	0,00	0,00	0,00
STATION 6	SORT 7	7	0,00	1,17	0,00	0,00	0,00	0,00
STATION 7	SORT 7	7	0,00	0,00	0,00	0,00	0,00	0,00
STATION 1	SORT 8	8	0,00	2,17	0,00	0,00	0,00	0,00

APPENDIX B.

Indata för simulering av Alfa-Lavals plattvärmegrupp.

På sidorna B2 och B3 finns behandlingstider. Dessa skall tolkas enligt följande. På varje rad står behandlingstider för en till sex operatörer för en viss station och produkt. Vilken station och produkt kan läsas i vänstermarginalen. Behandlingstiden noll anger ett inte tillåtet antal operatörer.

Sidorna B4 och B5 innehåller en rad maskinbundna konstanter plus produkternas fördelning vid simuleringens start. Konstanten STOP anger inköns maximala längd. STOPPEDSTART anger när en maskin som stannats p.g.a. av nästa maskins kö kan startas igen. WANT och DANGER varnar för att inkön håller på att bli för lång.

STATION 2	SORT 8	0.00	0.00	0.00	0.00	0.00	0.00
STATION 3	SORT 8	0.00	4.50	0.00	2.33	0.00	1.50
STATION 4	SORT 8	0.00	1.67	0.00	0.83	0.00	0.00
STATION 5	SORT 8	0.00	3.17	0.00	0.00	0.00	0.00
STATION 6	SORT 8	0.00	1.17	0.00	0.00	0.00	0.00
STATION 7	SORT 8	0.00	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT 9	0.00	1.83	0.00	0.00	0.00	0.00
STATION 2	SORT 9	0.00	0.00	0.00	0.00	0.00	0.00
STATION 3	SORT 9	0.00	0.00	0.00	0.00	0.00	0.00
STATION 4	SORT 9	0.00	0.00	0.00	0.00	0.00	0.00
STATION 5	SORT 9	0.00	0.00	0.00	0.00	0.00	0.00
STATION 6	SORT 9	0.00	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT 9	19.17	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT10	0.00	2.08	0.00	0.00	0.00	0.00
STATION 2	SORT10	0.00	1.00	0.00	0.50	0.00	0.00
STATION 3	SORT10	0.00	3.50	0.00	1.83	0.00	1.17
STATION 4	SORT10	0.00	1.50	0.00	0.83	0.00	0.00
STATION 5	SORT10	0.00	0.00	0.00	0.00	0.00	0.00
STATION 6	SORT10	0.00	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT10	0.00	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT11	1.25	0.00	0.00	0.00	0.00	0.00
STATION 2	SORT11	1.83	1.00	0.67	0.50	0.00	0.00
STATION 3	SORT11	3.83	2.00	1.33	1.00	0.77	0.67
STATION 4	SORT11	1.50	0.67	0.50	0.33	0.00	0.00
STATION 5	SORT11	0.00	0.00	0.00	0.00	0.00	0.00
STATION 6	SORT11	0.00	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT11	0.00	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT12	1.33	0.00	0.00	0.00	0.00	0.00
STATION 2	SORT12	0.00	0.00	0.00	0.00	0.00	0.00
STATION 3	SORT12	6.33	3.17	2.11	1.67	1.27	1.05
STATION 4	SORT12	1.00	0.50	0.33	0.00	0.00	0.00
STATION 5	SORT12	0.00	0.00	0.00	0.00	0.00	0.00
STATION 6	SORT12	1.33	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT12	0.00	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT13	1.33	0.00	0.00	0.00	0.00	0.00
STATION 2	SORT13	2.67	1.33	0.89	0.67	0.53	0.45
STATION 3	SORT13	4.33	2.17	1.50	1.17	0.87	0.72
STATION 4	SORT13	2.00	1.00	0.67	0.50	0.00	0.00
STATION 5	SORT13	0.00	0.00	0.00	0.00	0.00	0.00
STATION 6	SORT13	1.33	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT13	0.00	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT14	1.50	0.00	0.00	0.00	0.00	0.00
STATION 2	SORT14	2.67	1.33	0.89	0.67	0.53	0.45
STATION 3	SORT14	4.33	2.17	1.50	1.17	0.87	0.72
STATION 4	SORT14	2.00	1.00	0.67	0.50	0.00	0.00
STATION 5	SORT14	0.00	1.17	0.00	0.00	0.00	0.00
STATION 6	SORT14	0.00	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT14	4.33	0.00	0.00	0.00	0.00	0.00
STATION 1	SORT15	1.00	0.00	0.00	0.00	0.00	0.00
STATION 2	SORT15	1.83	1.00	0.67	0.50	0.00	0.00
STATION 3	SORT15	4.33	2.17	1.50	1.17	0.87	0.72
STATION 4	SORT15	0.33	0.17	0.00	0.00	0.00	0.00
STATION 5	SORT15	0.00	2.17	0.00	0.00	0.00	0.00
STATION 6	SORT15	0.00	0.00	0.00	0.00	0.00	0.00
STATION 7	SORT15	0.00	0.00	0.00	0.00	0.00	0.00

P45	TI	VAGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P41	RST		100
A20	TI	VAGBAND	100
A20	TI	VRGBAND	100
A20	TI	VAGBAND	100
A10	RST	VAGBAND	100
A10	RST	VAGBAND	100
A10	RST		100
A10	RST		100
A10	RST		100
A10	RST		100
P45	RST		100
P45	RST		100
P45	RST		100
P45	RST		100
P45	RST		100
P45	RST		100
P45	RST		100
A15	TI	VAGBAND	100
A15	TI	VRGBAND	100
A15	TI	VRGBAND	100
A30	RST	VRGBAND	100
A30	RST	VRGBAND	100
A30	RST	VRGBAND	100
A30	PST	VAGBAND	100
A30	RST	VRGBAND	100
A30	RST	VRGBAND	100
A30	TI	VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
P252		VRGBAND	100
A20	TI		100
A20	TI		100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P45	TI	VRGBAND	100
P41	TI		100
P41	TI		100
P41	TI		100
A15	RST	VAGBAND	100
A15	RST	VRGBAND	100
A15	RST	VAGBAND	100
A10	TI	VAGBAND	100

APPENDIX C

Simuleringsresultat Alfa-Laval

För att kunna prioritera maskinerna olika har varje maskin tilldelats tre konstanter Q, W och VIKT.

När en maskin söker operatörer multipliceras dess kölängd med dess Q-värde för att få fram ett mått hur gärna maskinen vill ha operatörerna. En maskin med högt Q värde har stor sannolikhet att få tag på operatörer när den söker.

En maskin med lågt VIKT värde har lätt att förlora operatörer till andra maskiner.

När en maskin vill bli av med operatörer försöker den fördela dessa bland de andra maskinerna. Operatörerna vill programmet ge till maskiner som har lång kö och som är känsliga i det att de måste arbeta hela tiden. Högt W-värde innebär att en maskin lätt får operatörer som andra gör sig av med.

I den första simuleringen sid C2 prioriteras alla maskiner lika. Se tabellen härunder.

KONST1-4	0.40	0.70	0.30	1.30
STATION NR	Q	W	VIKT	
1	1.00	1.00	1.00	
2	1.00	1.00	1.00	
3	1.00	1.00	1.00	
4	1.00	1.00	1.00	
5	1.00	1.00	1.00	
6	1.00	1.00	1.00	
7	1.00	1.00	1.00	

ANTALET OPERATÖRER 10
 ANTAL BUNTAR GENOM SYSTEMET 65
 GENOMSNITTLIG GENOMLOPPSTID 50,00
 OVERKSAMMA MANTIMMAR 91,67
 EXCENTER PRESS
 STATION NR.1
 TIDSUTNYTTJANDEPROCENT= 89,26
 KAPACITETSUTNYTTJANDEPROCENT= 89,26
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P41 TI 12 100

GRÄDNING
 STATION NR.2
 TIDSUTNYTTJANDEPROCENT= 26,73
 KAPACITETSUTNYTTJANDEPROCENT= 11,76
 P45 TI VÄGBAND 14 100

LIMNING
 STATION NR.3
 TIDSUTNYTTJANDEPROCENT= 93,64
 KAPACITETSUTNYTTJANDEPROCENT= 64,26
 RENSNING
 STATION NR.4
 TIDSUTNYTTJANDEPROCENT= 63,61
 KAPACITETSUTNYTTJANDEPROCENT= 37,20
 A20 TI 5 100
 A20 TI 5 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100

VÄGBANDSVETSNING
 STATION NR.5
 TIDSUTNYTTJANDEPROCENT= 69,13
 KAPACITETSUTNYTTJANDEPROCENT= 69,13
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100

PRESSVETSNING
 STATION NR.6
 TIDSUTNYTTJANDEPROCENT= 57,84
 KAPACITETSUTNYTTJANDEPROCENT= 57,84
 A30 TI VÄGBAND 8 100

PUNKTSVETSNING
 STATION NR.7
 TIDSUTNYTTJANDEPROCENT= 43,18
 KAPACITETSUTNYTTJANDEPROCENT= 43,18

I de följande simuleringarna används de värden på konstanterna som visas i tabellen nedan. Dessa värden ger ett bra simuleringresultat men har tagit många körningar att få fram.

KONST1-4	0,40	0,70	0,30	1,30
STATION NR	Q	W	VIKT	
1	5,00	5,00	5,00	
2	1,50	1,00	1,20	
3	1,50	1,40	1,20	
4	2,10	1,40	1,40	
5	4,50	1,60	3,10	
6	2,80	0,80	2,50	
7	2,30	0,70	2,00	

"Optimal" körning, tio operatörer.

ANTALET OPERATÖREP 10
 ANTAL BUNTAR GENOM SYSTEMET 70
 GENOMSNITTLIG GENOMLOPPSTID 41.00
 OVERKSAMMA MANTIMMAR 17.10
 EXCENTER PRESS
 STATION NR.1
 TIDSUTNYTTJANDEPROCENT=100.00
 KAPACITETSUTNYTTJANDEPROCENT=100.00
 A15 RST VÄGBAND 4 100
 A15 RST VÄGBAND 4 100
 A10 TI VÄGBAND 2 100

GRADNING
 STATION NR.2
 TIDSUTNYTTJANDEPROCENT= 30.75
 KAPACITETSUTNYTTJANDEPROCENT= 14.67
 A15 RST VÄGBAND 4 100

LIMNING
 STATION NR.3
 TIDSUTNYTTJANDEPROCENT= 99.43
 KAPACITETSUTNYTTJANDEPROCENT= 66.98
 P41 TI 12 100
 P41 TI 12 100
 P41 TI 12 100

RENSNING
 STATION NR.4
 TIDSUTNYTTJANDEPROCENT= 75.07
 KAPACITETSUTNYTTJANDEPROCENT= 40.10
 A20 TI 5 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100

VÄGBANDSVETSNING
 STATION NR.5
 TIDSUTNYTTJANDEPROCENT= 76.06
 KAPACITETSUTNYTTJANDEPROCENT= 76.06
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 A20 TI 5 100

PRESSVETSNING
 STATION NR.6
 TIDSUTNYTTJANDEPROCENT= 58.81
 KAPACITETSUTNYTTJANDEPROCENT= 58.81
 PUNKTSVETSNING
 STATION NR.7
 TIDSUTNYTTJANDEPROCENT= 43.18
 KAPACITETSUTNYTTJANDEPROCENT= 43.18

"Optimal körning, nio operatörer.

ANTALET OPERATÖRER 9
 ANTAL BUNTAR GENOM SYSTEMET 63
 GENOMSNIITTLIG GENOMLOPPSTID 47,00
 ÖVERKSAMMA MANTIMMAR 11,75
 EXCENTER PRESS
 STATION NR.1
 TIDSUTNYTTJANDEPROCENT= 99,42
 KAPACITETSUTNYTTJANDEPROCENT= 99,42
 A15 RST VÄGBAND 4 100
 A15 RST VÄGBAND 4 100
 A15 RST VÄGBAND 4 100
 A10 TI VÄGBAND 2 100

GRÄDNING
 STATION NR.2
 TIDSUTNYTTJANDEPROCENT= 47,01
 KAPACITETSUTNYTTJANDEPROCENT= 10,99
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P41 TI 12 100
 P41 TI 12 100
 P41 TI 12 100

LIMNING
 STATION NR.3
 TIDSUTNYTTJANDEPROCENT=100,00
 KAPACITETSUTNYTTJANDEPROCENT= 59,59
 P252 VÄGBAND 15 100
 A20 TI 5 100
 A20 TI 5 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100

RENSNING
 STATION NR.4
 TIDSUTNYTTJANDEPROCENT= 70,21
 KAPACITETSUTNYTTJANDEPROCENT= 37,35
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100

VÄGBANDSVETSNING
 STATION NR.5
 TIDSUTNYTTJANDEPROCENT= 62,87
 KAPACITETSUTNYTTJANDEPROCENT= 62,87
 A30 RST VÄGBAND 8 100
 A30 RST VÄGBAND 8 100
 A30 TI VÄGBAND 8 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100

PRESSVETSNING
 STATION NR.6
 TIDSUTNYTTJANDEPROCENT= 55,92
 KAPACITETSUTNYTTJANDEPROCENT= 55,92
 PUNKTSVETSNING
 STATION NR.7
 TIDSUTNYTTJANDEPROCENT= 43,18
 KAPACITETSUTNYTTJANDEPROCENT= 43,18

"Optimal" körning, elva operatörer.

ANTALET OPERATÖRER 11
 ANTAL BUNTAR GENOM SYSTEMET 71
 GENOMSNITTLIG GENOMLOPPSTID 41.00
 OVERKSAMMA MANTIMMAR 149.75
 EXCENTER PRESS
 STATION NR. 1
 TIDSUTNYTTJANDEPROCENT=100.00
 KAPACITETSUTNYTTJANDEPROCENT=100.00
 A15 RST VÄGBAND 4 100
 A15 RST VÄGBAND 4 100
 A10 TI VÄGBAND 2 100

GRADNING
 STATION NR. 2
 TIDSUTNYTTJANDEPROCENT= 27.19
 KAPACITETSUTNYTTJANDEPROCENT= 15.80
 LIMNING
 STATION NR. 3
 TIDSUTNYTTJANDEPROCENT= 86.91
 KAPACITETSUTNYTTJANDEPROCENT= 69.09
 P41 TI 12 100
 P41 TI 12 100
 P41 TI 12 100
 A15 RST VÄGBAND 4 100

RENSNING
 STATION NR. 4
 TIDSUTNYTTJANDEPROCENT= 61.14
 KAPACITETSUTNYTTJANDEPROCENT= 40.98
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100

VÄGBANDSVETSNING
 STATION NR. 5

TIDSUTNYTTJANDEPROCENT= 77.51
 KAPACITETSUTNYTTJANDEPROCENT= 77.51
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 A20 TI 5 100
 A20 TI 5 100
 P45 TI VÄGBAND 14 100

PRESSVETSNING
 STATION NR. 6
 TIDSUTNYTTJANDEPROCENT= 58.81
 KAPACITETSUTNYTTJANDEPROCENT= 58.81
 PUNKTSVETSNING
 STATION NR. 7
 TIDSUTNYTTJANDEPROCENT= 43.18
 KAPACITETSUTNYTTJANDEPROCENT= 43.18

"Optimal" körning, tio operatörer. Här tillåts också produkterna att passera varandra i kön.

ANTALET OPERATÖRER 10
 ANTAL BUNTAR GENOM SYSTEMET 69
 GENOMSNITTLIG GENOMLOPPSTID 40,00
 OVERKSAMMA MANTIMMAR 11,84
 EXCENTER PRESS
 STATION NR.1
 TIDSUTNYTTJANDEPROCENT=100,00
 KAPACITETSUTNYTTJANDEPROCENT=100,00
 A15 RST VÄGBAND 4 100
 A15 RST VÄGBAND 4 100
 A10 TI VÄGBAND 2 100

GRADNING
 STATION NR.2
 TIDSUTNYTTJANDEPROCENT= 38,28
 KAPACITETSUTNYTTJANDEPROCENT= 13,85
 LIMNING
 STATION NR.3
 TIDSUTNYTTJANDEPROCENT= 99,08

KAPACITETSUTNYTTJANDEPROCENT= 67,92
 P41 TI 12 100
 P41 TI 12 100
 P41 TI 12 100
 A15 RST VÄGBAND 4 100

RENSNING
 STATION NR.4
 TIDSUTNYTTJANDEPROCENT= 76,81
 KAPACITETSUTNYTTJANDEPROCENT= 41,30
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100
 P45 TI VÄGBAND 14 100

VÄGBANDSVETSNING
 STATION NR.5
 TIDSUTNYTTJANDEPROCENT= 75,19
 KAPACITETSUTNYTTJANDEPROCENT= 75,19
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 P252 VÄGBAND 15 100
 A20 TI 5 100
 A20 TI 5 100
 P45 TI VÄGBAND 14 100

PRESSVETSNING
 STATION NR.6
 TIDSUTNYTTJANDEPROCENT= 58,81
 KAPACITETSUTNYTTJANDEPROCENT= 58,81
 PUNKTSVETSNING
 STATION NR.7
 TIDSUTNYTTJANDEPROCENT= 43,18
 KAPACITETSUTNYTTJANDEPROCENT= 43,18

APPENDIX D

Interaktivt simuleringsprogram med manuella beslutsregler

LAVAL*EAFIL(1).HPRUG

```

1      BEGIN
2      SIMULATION CLASS ALFA(PEKNR);
3      INTEGER PEKNR;
4      BEGIN
5          REF(STATION)ARRAY PLATS(1:PEKNR);
6          REF(PRINTFILE)FIL;
7          CHARACTER SVAR;
8          INTEGER ANTSTATION,STYRLAG,EJSYSSELSATTAOP;
9          INTEGER ANTPLATTSORT,I,J,K;
10         REAL SIMPERIOD,PERIOD,OVERKTID,OVERKSTART;
11         BOOLEAN HOPPOVER;
12
13         PROCEDURE RAPPORT;
14         BEGIN
15             REF(PLATTA)FORST;
16             OUTTEXT(∇TID∇);
17             OUTFIX(TIME,1,6);
18             OUTIMAGE;
19             IF EJSYSSELSATTAOP>0 THEN
20                 BEGIN
21                     OUTTEXT(∇ANTAL EJ SYSSSELSATTA OPERATÖRER=∇);
22                     OUTINT(EJSYSSELSATTAOP,2);
23                     OUTIMAGE;
24                 END;
25             OUTTEXT(∇PLATS ∇);
26             FOR I:=1 STEP 1 UNTIL ANTSTATION DO
27                 BEGIN
28                     OUTINT(I,2);
29                     OUTTEXT(∇ ∇);
30                 END;
31             OUTIMAGE;
32             OUTTEXT(∇KÖLANGD ∇);
33             FOR I:=1 STEP 1 UNTIL ANTSTATION DO
34                 BEGIN
35                     OUTINT(PLATS(I).QUEU.CARDINAL,2);
36                     OUTTEXT(∇ ∇);
37                 END;
38             OUTIMAGE;
39             OUTTEXT(∇ANT OP ∇);
40             FOR I:=1 STEP 1 UNTIL ANTSTATION DO
41                 BEGIN
42                     INSPECT PLATS(I) WHEN MASKIN DO
43                         BEGIN
44                             OUTINT(PLATS(I).ANTOP,2);
45                             OUTTEXT(∇ ∇);
46                         END
47                     OTHERWISE
48                         OUTTEXT(∇ ∇);
49                 END;
50             OUTIMAGE;
51             OUTTEXT(∇TILLÄTET∇);
52             FOR I:=1 STEP 1 UNTIL ANTSTATION DO
53                 BEGIN
54                     INSPECT PLATS(I) WHEN MASKIN DO
55                         BEGIN
56                             IF NOT QUEU.EMPTY THEN
57                                 BEGIN
58                                     FORST:=QUEU.FIRST;
59                                     FOR J:=1 STEP 1 UNTIL 6 DO
60                                         BEGIN
61                                             IF BEARBETTID(FORST,PLATTSORT,J)>0.001 THEN
62                                                 OUTINT(J,1)
63                                             ELSE

```



```

192
193     BOOLEAN PROCEDURE JASVAR;
194     BEGIN
195         CHARACTER SVAR;
196         JA1:    INIMAGE;
197         SVAR:=INCHAR;
198         IF SVAR = 'J' THEN
199             JASVAR:=TRUE
200         ELSE
201             BEGIN
202                 IF SVAR = 'N' THEN
203                     JASVAR:=FALSE
204                 ELSE
205                     BEGIN
206                         IF SVAR = 'H' THEN
207                             HELP
208                         ELSE
209                             BEGIN
210                                 OUTTEXT(∇FEL SVARAT FÖRSÖK IGEN!∇);
211                                 OUTIMAGE;
212                                 GO TO JA1;
213                             END;
214                         END;
215                     END;
216             END*****JASVAR*****;
217             COMMENT*****;
218
219     INTEGER PROCEDURE INLASHEL;
220     BEGIN
221         INLASHEL:=ININT;
222     END*****INLASHEL*****;
223     COMMENT*****;
224
225     REAL PROCEDURE INLASREEL;
226     BEGIN
227         INLASREEL:=INREAL;
228     END*****INLASREEL*****;
229     COMMENT*****;
230
231     PROCEDURE SKAPASYST;
232     BEGIN
233         INTEGER SLASK;
234         REF(OUTFILE)SYST;
235         SYST:=NEW OUTFILE(∇HJALPFIL1∇);
236         SYST.OPEN(BLANKS(30));
237         OUTTEXT(∇HUR MÅNGA STATIONER ÄR DET?∇);
238         OUTIMAGE;
239         ANTSTATION:=INLASHEL;
240         SYST.OUTINT(ANTSTATION,5);
241         OUTTEXT(∇HUR MÅNGA PRODUKTER ÄR DET?∇);
242         OUTIMAGE;
243         SYST.OUTINT(INLASHEL,3);
244         SYST.OUTIMAGE;
245         OUTTEXT(∇ANGE I ORDNINGSFÖLJD STATIONERNAS VARIABLER OCH N
246         OUTIMAGE;
247         FOR I:=1 STEP 1 UNTIL ANTSTATION DO
248             BEGIN
249                 OUTTEXT(∇STATIONENS TYP?∇);
250                 OUTIMAGE;
251                 SLASK:=INLASHEL;
252                 SYST.OUTINT(SLASK,1);
253                 IF SLASK=1 THEN
254                     BEGIN
255                         OUTTEXT(∇STOP∇);

```

```

256         OUTIMAGE;
257         SYST.OUTINT(INLASHL,3);
258     END
259     ELSE
260     IF SLASK=Z THEN
261     BEGIN
262         OUTTEXT(VHUR OFTA SKALL INLAGRET AKTIVERAS?V);
263         OUTIMAGE;
264         SYST.OUTFIX(INLASREEL,1,6);
265     END;
266     SYST.SETPOS(10);
267     OUTTEXT(VNAMN PA PLATSEN?V);
268     OUTIMAGE;
269     INIMAGE;
270     SYST.OUTTEXT(INTEXT(20));
271     SYST.OUTIMAGE;
272     END;
273     SYST.CLOSE;
274     END*****SKAPASYST*****;
275     COMMENT*****;
276
277     PROCEDURE LANKASYST;
278     BEGIN
279         REAL RSLASK;
280         INTEGER I,STATIONTYP,SLASK;
281         REF(INFILE)SYST;
282         SYST:=NEW INFILE(VHJALPFIL1V);
283         SYST.OPEN(BLANKS(30));
284         SYST.INIMAGE;
285         ANTSTATION:=SYST.ININT;
286         ANTPLATTSORT:=SYST.ININT;
287         FOR I:=1 STEP 1 UNTIL ANTSTATION DO
288         BEGIN
289             SYST.INIMAGE;
290             STATIONTYP:=SYST.ININT;
291             IF STATIONTYP = 1 THEN
292             BEGIN
293                 SLASK:=SYST.ININT;
294                 SYST.SETPOS(10);
295                 PLATS(I):=-NEW MASKIN(I,STATIONTYP,SYST.INTEXT(20),
296                     NEW HEAD,SLASK);
297             END
298             ELSE
299             BEGIN
300                 IF STATIONTYP = 2 THEN
301                 BEGIN
302                     RSLASK:=SYST.INREAL;
303                     SYST.SETPOS(10);
304                     PLATS(I):=-NEW INLAGER(I,STATIONTYP,SYST.INTEXT(
305                         NEW HEAD,RSLASK);
306                 END
307             ELSE
308             BEGIN
309                 IF STATIONTYP = 3 THEN
310                 BEGIN
311                     SYST.SETPOS(10);
312                     PLATS(I):=-NEW FARDIGLAGER(I,STATIONTYP,
313                         SYST.INTEXT(20),NEW HEAD);
314                 END;
315             END;
316         END;
317     END;
318     SYST.CLOSE;
319     END*****LANKASYST*****;

```

```

320      COMMENT*****;
321
322      PROCEDURE SKAPATID;
323      BEGIN
324          REF(OUTFILE)SYST;
325          SYST:=NEW OUTFILE(∇HJALPFIL2∇);
326          SYST.OPEN(BLANKS(40));
327          OUTTEXT(∇NU SKALL VI LÄGGA IN TILLVERKNINGS OCH ∇);
328          OUTTEXT(∇RIGGNINGSTIDER∇);
329          OUTIMAGE;
330          OUTTEXT(∇FÖR VARJE MASKIN (OBS EJ PLATS) SKALL ∇);
331          OUTTEXT(∇VI LÄGGA IN ETT FÄLT∇);
332          OUTIMAGE;
333          OUTTEXT(∇PÅ VARJE RAD SKALL STA TIDER FÖR ∇);
334          OUTTEXT(∇1-6 OPERATÖRER FÖR VISS PLATTSORT∇);
335          OUTIMAGE;
336          FOR I:=1 STEP 1 UNTIL ANTSTATION DO
337              BEGIN
338                  INSPECT PLATS(I) WHEN MASKIN DO
339                      BEGIN
340                          OUTTEXT(∇BEARBETNINGSTIDER FÖR PLATS∇);
341                          OUTINT(STATIONNR,3);
342                          OUTIMAGE;
343                          FOR J:=1 STEP 1 UNTIL ANTPLATTSORT DO
344                              BEGIN
345                                  INIMAGE;
346                                  SYST.OUTTEXT(INTEXT(40));
347                                  SYST.OUTIMAGE;
348                              END;
349                          OUTTEXT(∇RIGGTIDER FÖR PLATS∇);
350                          OUTINT(STATIONNR,3);
351                          OUTIMAGE;
352                          FOR J:=1 STEP 1 UNTIL ANTPLATTSORT DO
353                              BEGIN
354                                  INIMAGE;
355                                  SYST.OUTTEXT(INTEXT(40));
356                                  SYST.OUTIMAGE;
357                              END;
358                          END;
359                      END;
360                  SYST.CLOSE;
361          END*****SKAPATID*****;
362      COMMENT*****;
363
364      PROCEDURE LANKATID;
365      BEGIN
366          REF(INFILE)SYST;
367          SYST:=NEW INFILE(∇HJALPFIL2∇);
368          SYST.OPEN(BLANKS(40));
369          FOR I:=1 STEP 1 UNTIL ANTSTATION DO
370              BEGIN
371                  INSPECT PLATS(I) WHEN MASKIN DO
372                      BEGIN
373                          FOR J:=1 STEP 1 UNTIL ANTPLATTSORT DO
374                              BEGIN
375                                  SYST.INIMAGE;
376                                  FOR K:=1 STEP 1 UNTIL 6 DO
377                                      BEGIN
378                                          BEARBETTID(J,K):=SYST.INREAL;
379                                      END;
380                                  END;
381                              END;
382                          FOR J:=1 STEP 1 UNTIL ANTPLATTSORT DO
383                              BEGIN
384                                  SYST.INIMAGE;

```

```

384         FOR K:=1 STEP 1 UNTIL 6 DO
385             BEGIN
386                 RIGGTID(J,K):=SYST.INREAL;
387             END;
388         END;
389     END;
390 END;
391 SYST.CLOSE;
392 END*****LANKATID*****;
393 COMMENT*****;
394
395 PROCEDURE SKAPAPROGRAM;
396 BEGIN
397     INTEGER SLASK;
398     REF(OUTFILE)SYST;
399     SYST:=NEW OUTFILE(∇HJALPFIL3∇);
400     SYST.OPEN(BLANKS(33));
401     OUTTEXT(∇NU SKALL PRODUKTIONSPROGRAMMET GÖRAS UPP∇);
402     OUTIMAGE;
403     OUTTEXT(∇FÖRST ANGER DU KÖN STORLEK VID ∇);
404     OUTTEXT(∇PLATS IV);
405     OUTIMAGE;
406     OUTTEXT(∇SEDAN ANGER DU KÖNS INNEHÅLL I ORDNINGSFÖLJD∇);
407     OUTIMAGE;
408     FOR I:=1 STEP 1 UNTIL ANTSTATION DO
409         BEGIN
410             OUTTEXT(∇PLATS∇);
411             OUTINT(PLATS(I).STATIONNR,2);
412             OUTIMAGE;
413             SLASK:=INLASHEL;
414             SYST.OUTINT(SLASK,3);
415             SYST.OUTIMAGE;
416             FOR J:=1 STEP 1 UNTIL SLASK DO
417                 BEGIN
418                     INIMAGE;
419                     SYST.OUTTEXT(INTEXT(20));
420                     SYST.OUTINT(INLASHEL,4);
421                     SYST.OUTINT(INLASHEL,4);
422                     SYST.OUTINT(INLASHEL,4);
423                     SYST.OUTIMAGE;
424                 END;
425             END;
426             SYST.CLOSE;
427         END*****SKAPAPROGRAM*****;
428         COMMENT*****;
429
430 PROCEDURE LANKAPROGRAM;
431 BEGIN
432     INTEGER SLASK;
433     REF(INFILE)SYST;
434     SYST:=NEW INFILE(∇HJALPFIL3∇);
435     SYST.OPEN(BLANKS(33));
436     FOR I:=1 STEP 1 UNTIL ANTSTATION DO
437         BEGIN
438             SYST.INIMAGE;
439             SLASK:=SYST.ININT;
440             FOR J:=1 STEP 1 UNTIL SLASK DO
441                 BEGIN
442                     SYST.INIMAGE;
443                     NEW PLATTA(SYST.INTEXT(20),SYST.ININT,SYST.ININT,
444                     SYST.ININT).INTO(PLATS(I).QUEU);
445                 END;
446             END;
447             SYST.CLOSE;

```

```

448     END*****LANKAPROGRAM*****;
449     COMMENT*****;
450
451     PROCEDURE FORDELAOP;
452     BEGIN
453         OUTTEXT(∇HUR MANGA OPERATÖRE?∇);
454         OUTIMAGE;
455         EJSYSSELSATTAOP:=IHLASHEL;
456     END*****FORDELAOP*****;
457     COMMENT*****;
458
459     LINK CLASS PLATTA(PLATTNAMN,PLATTYP,PLATTBUNT,PLATTSORT);
460     INTEGER PLATTSORT,PLATTYP,PLATTBUNT;
461     TEXT PLATTNAMN;
462     BEGIN
463         REAL FINISHTIME;
464     END*****PLATTA*****;
465     COMMENT*****;
466
467     PROCESS CLASS STATION(STATIONNR,STATIONTYP,NAMN,QUEU);
468     REF(HEAD)QUEU;
469     TEXT NAMN;
470     INTEGER STATIONTYP,STATIONNR;
471     BEGIN
472         REF(PLATTA)SERVED;
473         INTEGER ANTOP;
474
475     END*****MASKIN*****;
476     COMMENT*****;
477
478     STATION CLASS MASKIN(STOP);
479     INTEGER STOP;
480     BEGIN
481         REAL TOTARBETSTID,FAKTOR,STARTTID;
482         REAL ANVANOTIM,UTNYTJTIM,KANSLIG;
483         INTEGER ANT;
484         REAL ARRAY BEARBETTID,RIGGTID(1:ANTPLATTSORT,0:6);
485         BOOLEAN ARBETA;
486         REF(PLATTA) LASTSERVED;
487
488         BOOLEAN PROCEDURE KANJAGARBETA;
489         BEGIN
490             BOOLEAN PROCEDURE KJASTYRLAG1;
491             BEGIN
492                 BOOLEAN KLAR;
493                 INSPECT PLATS(STATIONNR+1) WHEN MASKIN DO
494                 BEGIN
495                     IF QUEU.CARDINAL<STOP THEN
496                         KLAR:=TRUE
497                     ELSE
498                         BEGIN
499                             OUTTEXT(∇PLATS∇);
500                             OUTINT(STATIONNR-1,2);
501                             OUTTEXT(∇STOPPAD PGA NÄSTA KÖ∇);
502                             OUTIMAGE;
503                         END;
504                     END
505                     OTHERWISE
506                         KLAR:=TRUE;
507                     IF KLAR AND BEARBETTID(SERVED,PLATTSORT,ANTOP)>
508                         0.001 THEN
509                         KJASTYRLAG1:=TRUE;
510                 END*****KJASTYRLAG1*****;
511                 KANJAGARBETA:=KJASTYRLAG1;

```

```

512      END*****KANJAGARBETA*****;
513      COMMENT*****;
514
515      PROCEDURE HARINGETATTGORA;
516      BEGIN
517          PROCEDURE HIAGSTYRLAG1;
518          BEGIN
519              IF ANTOP=0 THEN
520                  GO TO H1;
521                  OUTTEXT(∇PLATS∇);
522                  OUTINT(STATIONNR,2);
523                  OUTTEXT(∇ ∇);
524                  OUTTEXT(NAMN);
525                  OUTTEXT(∇ HAR INGET ATT GÖRA∇);
526                  OUTIMAGE;
527                  RAPPORT;
528                  STYRLAG1FLYTTA;
529      H1:      END*****HIAGSTYRLAG1*****;
530          HIAGSTYRLAG1;
531      END*****HARINGETATTGORA*****;
532      COMMENT*****;
533
534      PROCEDURE GORASIGAVMEDOP;
535      BEGIN
536          PROCEDURE GSAMOSTYRLAG1;
537          BEGIN
538              OUTTEXT(∇PLATS∇);
539              OUTINT(STATIONNR,2);
540              OUTTEXT(∇ ∇);
541              OUTTEXT(NAMN);
542              OUTTEXT(∇ KAN EJ ARBETA VIDARE∇);
543              OUTIMAGE;
544              RAPPORT;
545              STYRLAG1FLYTTA;
546          END*****GSAMOSTYRLAG1*****;
547          GSAMOSTYRLAG1;
548      END*****GORASIGAVMEDOP*****;
549      COMMENT*****;
550
551      BOOLEAN PROCEDURE SKALLJAGARBETAHAR;
552      BEGIN
553          BOOLEAN SANN;
554          FOR I:=1 STEP 1 UNTIL 6 DO
555              BEGIN
556                  IF BEARBETTID(SERVED.PLATTSORT,I)>0.000001 THEN
557                      SANN:=TRUE;
558              END;
559          SKALLJAGARBETAHAR:=SANN;
560      END*****SKALLJAGARBETAHAR*****;
561
562      COMMENT*****;
563      PROCEDURE STYRLAG1FLYTTA;
564      BEGIN
565          PROCEDURE FORDELAOP;
566          BEGIN
567              I:=INLASHEL;
568              PLATS(I).ANTOP:=PLATS(I).ANTOP+1;
569              REACTIVATE PLATS(I) DELAY 0.000001;
570              EJSYSSELSATTAOP:=EJSYSSELSATTAOP-1;
571          END*****FORDELAOP*****;
572          COMMENT*****;
573
574          PROCEDURE SKAFFAOP;
575          BEGIN

```

```

640      END;
641      SERVED:=-QUEU.FIRST;
642      FAKTOR:=0.0;
643      M4:  IF NOT SKALLJAGARBETAHAR THEN
644          BEGIN
645              HOPPOVER:=FALSE;
646              INSPECT PLATS(STATIONNR+1) WHEN MASKIN DO
647                  BEGIN
648                      IF QUEU.CARDINAL<STOP THEN
649                          HOPPOVER:=TRUE;
650                  END
651              OTHERWISE
652                  HOPPOVER:=TRUE;
653              IF HOPPOVER THEN GO TO M3;
654              IF ANTOP>0 THEN
655                  BEGIN
656                      OUTTEXT(▽PRODUKTEN SKALL EJ BEHANDLAS VID PLATS▽);
657                      OUTINT(STATIONNR,2);
658                      OUTTEXT(▽ MEN KAN EJ SKICKAS VIDARE▽);
659                      OUTIMAGE;
660                      GORASIGAVMEDOP;
661                  END;
662                  PASSIVATE;
663                  GO TO M4;
664          END;
665      M2:  IF NOT KANJAGARBETA THEN
666          BEGIN
667              IF ANTOP>0 THEN
668                  GORASIGAVMEDOP;
669              IF ANTOP=0 THEN
670                  PASSIVATE;
671                  GO TO M2;
672          END;
673      IF LASTSERVED==NONE OR
674      NOT SERVED.PLATTYP=LASTSERVED.PLATTYP THEN
675          TOTARBETSTID:=RIGGTID(SERVED.PLATTSORT,ANTOP)
676      ELSE
677          TOTARBETSTID:=0.0;
678          TOTARBETSTID:=TOTARBETSTID+REARBETTID(SERVED.PLATTSORT,ANTOP);
679          TOTARBETSTID:=TOTARBETSTID*(1-FAKTOR)*SERVED.PLATTBUNT/100;
680          TOTARBETSTID:=TOTARBETSTID*KANSLTG;
681          ANT:=ANTOP;
682          STARTTID:=TIME;
683          ARBETA:=TRUE;
684          HOLD(TOTARBETSTID);
685          ARBETA:=FALSE;
686          ANVANDTIM:=ANVANDTIM+TIME-STARTTID;
687          FOR I:=1 STEP 1 UNTIL 6 DO
688              IF REARBETTID(SERVED.PLATTSORT,I)>0.001 THEN
689                  J:=I;
690                  UTNYTJUTIM:=UTNYTJUTIM+ANT*(TIME-STARTTID)/J;
691                  IF STARTTID+TOTARBETSTID>TIME+0.0001 THEN
692                      BEGIN
693                          FAKTOR:=(TIME-STARTTID)/TOTARBETSTID;
694                          GO TO M2;
695                      END;
696      M3:  LASTSERVED:=-SERVED;
697          INSPECT FIL DO
698          BEGIN
699              OUTFIX(TIME,1,6);
700              OUTTEXT(▽ PLATS▽);
701              OUTINT(STATIONNR,2);
702              OUTTEXT(▽ KLAR MED ▽);
703              OUTTEXT(SERVED.PLATTNAMN);

```

```

768     OUTIMAGE;
769     IF NOT JASVAR THEN
770     SKAPASYST;
771     LANKASYST;
772     OUTTEXT(∇FINNS TILLV. OCH RIGGNINGSTIDER PÅ FIL?∇);
773     OUTIMAGE;
774     IF NOT JASVAR THEN
775     SKAPATID;
776     LANKATID;
777     OUTTEXT(∇FINNS PRODUKTIONSPROGRAMMET PÅ FIL?∇);
778     OUTIMAGE;
779     IF NOT JASVAR THEN
780     SKAPAPROGRAM;
781     LANKAPROGRAM;
782     OUTTEXT(∇VILL DU HA UTSKRIFT AV INDATA?∇);
783     OUTIMAGE;
784     IF JASVAR THEN INDATARAPPORT;
785     FÖRDELAOP;
786     OUTTEXT(∇HUR LANG SIMULERING?∇);
787     OUTIMAGE;
788     SIMPERIOD:=INREAL;
789     OUTTEXT(∇VIKTA BEHANDLINGSTIDERNA FÖR ATT TESTA KÄNSLIGHETEN?∇);
790     OUTIMAGE;
791     IF JASVAR THEN
792     BEGIN
793         FOR I:=1 STEP 1 UNTIL ANTSTATION DO
794         BEGIN
795             INSPECT PLATS(I) WHEN MASKIN DO
796             BEGIN
797                 OUTTEXT(∇PLATS∇);
798                 OUTINT(I,2);
799                 OUTIMAGE;
800                 KANSLIG:=INREAL;
801             END;
802         END;
803     END
804     ELSE
805     BEGIN
806         FOR I:=1 STEP 1 UNTIL ANTSTATION DO
807         BEGIN
808             INSPECT PLATS(I) WHEN MASKIN DO
809             KANSLIG:=1.0;
810         END;
811     END;
812     FIL:=-NEW PRINTFILE(∇HJALPFIL4∇);
813     FIL.OPEN(BLANKS(80));
814     FOR I:=1 STEP 1 UNTIL ANTSTATION DO
815     ACTIVATE PLATS(I);
816     IF STYRLAG = 1 THEN
817     BEGIN
818         OUTTEXT(∇HUR LANG TID ÖNSKAR DU MELLAN RAPPORTERNA?∇);
819         OUTIMAGE;
820         PERIOD:=INREAL;
821         WHILE TIME < SIMPERIOD DO
822         BEGIN
823             RAPPORT;
824             INSPECT PLATS(2) WHEN MASKIN DO
825             STYRLAGIFLYTTA;
826             HOLD(PERIOD);
827         END;
828     END
829     ELSE
830     HOLD(SIMPERIOD);
831     F2: RAPPORT;

```



```

704         OUTTEXT(▽BUNT ▽);
705         OUTINT(SERVED.PLATTBUNT,3);
706         OUTTEXT(▽ ANT OP▽);
707         OUTINT(ANTOP,2);
708         OUTIMAGE;
709         END;
710         SERVED.INTO(PLATS(STATIONNR+1).QUEUE);
711         INSPECT PLATS(STATIONNR+1) WHEN FARDIGLAGER DO
712         REACTIVATE PLATS(STATIONNR) AT TIME
713         WHEN MASKIN DO
714         IF QUEUE.EMPTY THEN
715         REACTIVATE PLATS(STATIONNR) DELAY 0.000001;
716         GO TO M1;
717         END*****MASKIN*****;
718         COMMENT*****;
719
720         STATION CLASS INLAGER(HUROFTA);
721         REAL HUROFTA;
722         BEGIN
723         WHILE NOT QUEUE.EMPTY DO
724         BEGIN
725         SERVED:=QUEUE.FIRST;
726         INSPECT PLATS(STATIONNR+1) WHEN MASKIN DO
727         IF QUEUE.CARDINAL < STOP THEN
728         PLATS(STATIONNR-1).SERVED.INTO(QUEUE)
729         ELSE
730         HOLD(HUROFTA);
731         END;
732         END*****INLAGER*****;
733         COMMENT*****;
734
735         STATION CLASS FARDIGLAGER;
736         BEGIN
737         FA1:   IF QUEUE.EMPTY THEN
738         BEGIN
739         PASSIVATE;
740         GO TO FA1;
741         END;
742         SERVED:=QUEUE.FIRST;
743         SERVED.FINISHTIME:=TIME;
744         WHILE TRUE DO
745         BEGIN
746         WHILE SERVED.SUC/=NONE DO
747         BEGIN
748         SERVED:=SERVED.SUC;
749         SERVED.FINISHTIME:=TIME;
750         END;
751         PASSIVATE;
752         END;
753         END*****FARDIGLAGER*****;
754         COMMENT*****;
755
756         OUTTEXT(▽VILKEN STYRLAG VILL DU ANVANDA?▽);
757         OUTIMAGE;
758         F1: STYRLAG:=INLASHET;
759         IF NOT STYRLAG = 1 THEN
760         BEGIN
761         OUTTEXT(▽DEN STYRLAGEN FINNS EJ▽);
762         OUTIMAGE;
763         OUTTEXT(▽FÖRSÖK IGEN!▽);
764         OUTIMAGE;
765         GO TO F1;
766         END;
767         OUTTEXT(▽FINNS SYSTEMET PA FIL?▽);

```

```
832     SIMPERIOD:=TIME;  
833     OUTIMAGE;  
834     OUTTEXT(VSIMULERINGEN SLUTV);  
835     OUTIMAGE;  
836     SLUTRAPPORT;  
837     FIL.CLOSE;  
838     END****ALFA****;  
839     NEW ALFA(10);  
840     END;  
841
```

Inledande exekvering

Under denna sker uppbyggnaden av systemet.

En kort sammanfattning av procedurerna som handhar uppbyggnaden av systemet.

SKAPASYST (rad 230-274) begär upplysningar om produktionlinjens innehåll. Den begär även värden på stationsberoende konstanter.

LANKASYST (276-319) Sköter att det i programmet skapas den produktionslinje som antingen byggts upp med Skapasyst eller som finns kvar sedan tidigare.

SKAPATID (rad 321-361) begär in bearbetnings- och riggtider för maskinerna.

LANKATID (rad 363-392) tilldelar maskinerna behandlingstider som antingen fåtts genom Skapatid eller funnits kvar från tidigare simuleringar.

SKAPAPROGRAM (rad 394-427) begär uppgifter om produktionsprogrammet.

LANKAPROGRAM (rad 429-448) tilldelar stationerna aktuellt produktionsprogram.

APPENDIX E.

Manual till interaktivt simuleringsprogram.

Uppstartning

Starta terminalen som vanligt, och ge run identitet.

▽RUN ALFA,kontonr,LAVAL

Programmet finns lagrat på en fil som heter EAFIL//AXLER.

Skall du köra ett redan inlagt system (se avsnitt 10.2) måste data flyttas ut till HJALPFIL1-3. Det kan vara lämpligt att använda ett addkommando (se programmerhandboken LDC)

▽ADD EAFIL.elementnamn

Vill du själv styra utmatningen av HJALPFIL4., som efter exekveringen innehåller en detaljerad beskrivning av vad som hänt, måste du asigna den.

▽ASG,T HJALPFIL4.

Om systemet som du tänker skapa innehåller fler än tio stationer måste rad 803 (NEW ALFA(10)) ändras. Istället för tio skall önskat antal stationer skrivas. Vidare får man räkna att utskriften i vissa rapporter blir kladdig och bör ändras.

Nu skall du kompilera programmet. Finns det i symbolisk form gör du så här.

▽S*I.M,N EAFIL.HPROG,HPROG

▽MAP,IN A,ABS

IN HPROG

LIB S I

▽XQT ABS

I appendix F finns en komplett uppstartning och kompilering.

```

576         I:=INLASHEL;
577         PLATS(I).ANTOP:=PLATS(I).ANTOP-1;
578         REACTIVATE PLATS(I) DELAY 0.000001;
579         EJSYSSELSATTAOP:=EJSYSSELSATTAOP+1;
580     END*****SKAFFAOP*****;
581     COMMENT*****;
582
583     OUTTEXT(∇VILL DU FLYTTA PA NÅGRA OPERATÖRER?∇);
584     OUTIMAGE;
585     IF JASVAR THEN
586     BEGIN
587         OVERKTID:=OVERKTID+(TIME-OVERKSTART)*EJSYSSELSATTA
588         OVERKSTART:=TIME;
589         IF EJSYSSELSATTAOP>0 THEN
590         BEGIN
591             OUTTEXT(∇TÄNK PA ATT DET FINNS LEDIGA OPERATÖRER
592             OUTIMAGE;
593         END;
594         OUTTEXT(∇TA EN ÖP? OM JA VARIFRAN?∇);
595         OUTIMAGE;
596         IF JASVAR THEN
597         BEGIN
598             ST1:         SKAFFAOP;
599             OUTTEXT(∇EN TILL? OM JA VARIFRAN?∇);
600             OUTIMAGE;
601             IF JASVAR THEN
602             GO TO ST1;
603         END;
604         OUTTEXT(∇NU KAN DU FÖRDELA∇);
605         OUTINT(EJSYSSELSATTAOP,2);
606         OUTTEXT(∇ OPERATÖRER∇);
607         OUTIMAGE;
608         OUTTEXT(∇GE EN ÖP? OM JA VART??∇);
609         OUTIMAGE;
610         IF JASVAR THEN
611         BEGIN
612             ST2:         FÖRDELAOP;
613             IF EJSYSSELSATTAOP>=1 THEN
614             BEGIN
615                 OUTTEXT(∇EN TILL? OM JA VART? ∇);
616                 OUTIMAGE;
617                 IF JASVAR THEN
618                 BEGIN
619                     FÖRDELAOP;
620                     GO TO ST2;
621                 END;
622             END;
623         END;
624         OUTTEXT(∇AKTIVERA EN OM JA VILKEN?∇);
625         OUTIMAGE;
626         IF JASVAR THEN
627         BEGIN
628             I:=INLASHEL;
629             REACTIVATE PLATS(I) DELAY 0.000001;
630         END;
631     END;
632     END*****STYRLAGIFLYTTA*****;
633     COMMENT*****;
634     M1:         IF QUEU.EMPTY THEN
635     BEGIN
636         SERVED:=-NONE;
637         HARINGETATTGORA;
638         PASSIVATE;
639         GO TO M1;

```

Frågor under den inledande exekveringen.

Frågor av logisk typ besvaras med: J=ja, N=nej, H=avbryter exekveringen och ger slutrapport.

<u>Fråga</u>	<u>Tydning</u>
VILKEN STYRLAG VILL DU ANVÄNDA?	1=manuell styrning (den enda som finns).
FINNS SYSTEMET PÅ FIL?	Finns systemet redan utlagt på HJALPFIL1. (med ett ADD-kommando). Vid ja svar sker en automatisk uppbyggnad av systemet. Vid nej sker en interaktiv uppbyggnad.
HUR MÅNGA STATIONER ÄR DET?	Hur många stationer inklusive inlager, färdidlager och maskiner består linjen av.
HUR MÅNGA PRODUKTER ÄR DET?	Hur många olika sorters produkter tillverkas det.
ANGE I ORDNINGSFÖLJD STATIONERNAS VARIABLER OCH NAMN	Ur efterföljande frågor skapas produktionslinjen.
STATIONSTYP?	Ange stationstyp enligt följande: 1=maskin 2=inlager 3=färdiglager
STOP?	Gäller vid typ maskin. Anger det maximala antal tillverkningsenheter som får plats framför maskinen.
HUR OFTA SKALL INLAGRET AKTIVERAS?	Anger hur ofta inlagret skickar produkter till första maskin.
NAMN PÅ PLATSEN?	Stationens namn (max 20 tecken)
FINNS TILLVERKNINGS- OCH RIGGNINGSTIDER PÅ FIL?	Finns nödvändiga tider för bearbetningen redan inlagda? (HJALPFIL2.)
FÖR VARJE MASKIN (OBS EJ PLATS) SKALL VI LÄGGA IN ETT FÄLT	Det är bara stationer av typ maskin som bearbetar vilket medför att det är bara de som skall ha tider.
PÅ VARJE RAD SKALL STÅ TIDER FÖR 1-6 OPERATÖRER FÖR VISSE PLATTSORT	
BEARBETNINGSTIDER FÖR PLATS(I)	Nu skall du lägga in bearbetningstider för station I. Du börjar med tiderna för den produkt du betecknat med sort 1. På raden lägger du in tiderna för 1 till 6 operatörer. ex. 3.2 1.6 1.2 0.0 0.0 0.0
RIGGTIDER FÖR PLATS(I)	Följer samma mönster som bearbetningstiderna.
FINNS PRODUKTIONSPROGRAMMET PÅ FIL?	Finns programmet för den period du vill simulera på HJALPFIL3.
FÖRST ANGER DU KÖNS STORLEK VID PLATS I	Hur många tillverkningsenheter ligger det framför stationen.

SEDAN ANGER DU I ORDNINGS-
FÖLJD KÖNS INNEHÅLL

Vad som skall anges är i ordningsföljd

1. Produktens namn (max 20 tecken)
2. Produktens typ som anger vilken riggning som är aktuell. Mellan produkter med samma typ sker ingen omriggning.
3. Hur många detaljer som finns i tillverkningsenheten.
4. Detaljens individuella sort som behandlingstiden bestäms av.

PLATS I

Nu sker uppräknigen

ex.

2)
A20 VÅGBAND)
3-95 5)
A30 VÅGBAND)
4 100 6)

VILL DU HA UTSKRIFT AV INDATA?

Ger en dokumentation av hur systemet är uppbyggt, behandlingstider och köinnehåll.

HUR MÅNGA OPERATÖRER?

Det totala antalet operatörer i systemet.

HUR LÅNG SIMULERING?

Simuleringens totala längd.

HUR LÅNG TID ÖNSKAR DU MELLAN
RAPPORTERNA?

Med detta intervall får du en utskrift av den aktuella situationen samtidigt som du kan flytta operatörerna.