



Pipe Chain®

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Dynamisk uppdatering av säkerhetstid

Utveckling och simulering av beräkningsmodeller
för PipeChain AB

Examensarbete 2010 – 2011
Institutionen för teknisk ekonomi och logistik
Avdelningen för produktionsekonomi

Författare: Ellinor Berkenäs och Måns Bynke

Handledare: Stig-Arne Mattsson, Lunds tekniska högskola
Klas Turesson, MA-system AB

Examinator: Ingela Elofsson, Lunds tekniska högskola

Förord

Detta examensarbete har utförts hösten 2010 och våren 2011 på PipeChain AB, ett dotterbolag till MA-system AB, som avslutning på undertecknades civilingenjörsutbildning i industriell ekonomi vid Lunds tekniska högskola (LTH). Det har varit ett givande och utmanande arbete som utmynnat i ett resultat som vi hoppas och tror att PipeChain kan ha nytta av i framtiden.

Vi vill rikta ett stort tack till Klas Turesson, vår handledare på MA-system AB, som var den som tillsåg att detta arbete överhuvudtaget blev av och som genom hela vårt arbete varit behjälplig med värdefulla och intressanta synpunkter. Ett stort tack riktas även till vår handledare vid LTH, adjungerad professor Stig-Arne Mattsson vid institutionen för teknisk ekonomi och logistik. Han har med sin långa erfarenhet och sitt engagemang för vårt examensarbete varit ett stort stöd för oss.

Dessutom vill vi tacka alla övriga trevliga och kunniga medarbetare på PipeChain och MA-system för kommentarer, hjälp och svar på frågor under arbetets gång. Utan er hjälp hade detta examensarbete inte kunnat genomföras i sin nuvarande form.

Lund i april 2011

Ellinor Berkenäs

Måns Bynke

Sammanfattning

Titel	Dynamisk uppdatering av säkerhetstid – Utveckling och simulering av beräkningsmodeller för PipeChain AB
Författare	Ellinor Berkenäs och Måns Bynke
Handledare	Stig-Arne Mattsson, Institutionen för teknisk ekonomi och logistik, Lunds tekniska högskola Klas Turesson, MA-system AB
Bakgrund	Företaget PipeChain utvecklar och distribuerar en mjukvarulösning, med samma namn som företaget, för styrning av ett företags flödeskedja. Den lagerstyrningslogik som används i PipeChain bygger på ett tidskoncept, där lagernivåer mäts i täcktider istället för kvantiteter. Detta innebär bl.a. att man använder begreppet säkerhetstid, vilket är den tidsperiods efterfrågan som lagret minst måste täcka.
Syfte	Projektets syfte är att ta fram en modell för dynamisk uppdatering av säkerhetstid samt att testa modellen med hjälp av simuleringar för att undersöka giltigheten och beskriva vilken eventuell förbättring den nya modellen frambringar.
Metod	En modell, den s.k. originalmodellen, som motsvarar hur lagerstyrningen fungerar i PipeChain idag har tagits fram. Med grund i en litteraturstudie och med hjälp av successiv testning med datasimuleringar har därefter fyra olika modeller för dynamisk uppdatering av säkerhetstid utvecklats. De dynamiska modellernas prestation har genom simuleringar undersökts och jämförts med originalmodellen. Resultaten har genomgått statistiska analyser. Indata till simuleringarna har dels varit teoretisk efterfrågan, genererad med hjälp av statistiska fördelningar, och dels verklig efterfrågan från ett av PipeChains kundföretag. Resultaten från den teoretiska efterfrågan ligger till grund för generella slutsatser, medan resultaten från verklig efterfrågan illustrerar och verifierar de teoretiska resultaten. De resultat som tagits fram är dels mått på uppnådd servicenivå (förmågan att styra mot målservicenivån, medelservicenivå samt standardavvikelse hos servicenivån) och dels medellagernivå.
Slutsatser	De dynamiska modellerna som tagits fram uppvisar alla en förbättrad instyrning mot önskad servicenivå jämfört med originalmodellen. Målet uppnås utan orimlig ackumulering av lager. Då systematiska variationer förekommer i efterfrågan ger framför allt modellen <i>traditionell algoritm med justeringar</i> mycket bra resultat. Då inga systematiska variationer finns kan teoretiskt sett en optimerad statisk säkerhetstid ge bättre resultat än en dynamisk. Det bedöms dock sällan vara möjligt att i praktiken välja en så optimerad statisk säkerhetstid, varför de dynamiska modellerna i de flesta fall måste rekommenderas.
Nyckelord	Lagerstyrning, dynamisk säkerhetstid, dynamiskt säkerhetslager, tidsfasad beställningspunkt, servicenivå, efterfrågan med säsongsvariationer

Summary

Title	Dynamic updating of safety time – Development and simulation of calculation models for PipeChain AB
Authors	Ellinor Berkenäs and Måns Bynke
Supervisors	Stig-Arne Mattsson, The Department of Industrial Management and Logistics, Faculty of Engineering at Lund University. Klas Turesson, MA-system AB
Background	PipeChain is a company that is developing and distributing a software solution with the purpose of controlling the supply chain of a company. The inventory control logics being used in PipeChain is founded on a time concept, where inventory levels are measured in days of supply instead of quantities. E.g., this means the use of a safety time, which is the minimum time period for which the inventory should last.
Objective	The objective of this project is to develop a model of dynamic updating of safety time and test this model using simulations to examine the validity and to describe what improvement, if any, the new model brings.
Method	<p>A model, the so called original model, which corresponds to the current logics of the inventory control in PipeChain has been developed. From the basis of a literature review and with the help of consecutive testing using computer simulations, four different models of dynamic safety time updating have been developed.</p> <p>The performance of the dynamic models has been compared to the original model using simulation. The results have been subject to statistical analysis. Input to the simulations has been both theoretical demand generated from statistical distributions and demand from one of PipeChain's customers. The results from using the theoretical demand have been the basis for general conclusions while the results from the real demand illustrates the theoretical results. The results consist of the achieved service level (ability to reach goal service level, mean and standard deviation of the service level) and average inventory level.</p>
Conclusions	The developed dynamic models all indicate a better performance than the original model when it comes to achievement of the goal service level. The goal is reached without unacceptable increase in inventory levels. With systematic changes in demand, the model <i>traditional algorithm with adjustments</i> performs best. With no systematic changes, an optimized static safety time gives better results, at least in theory. The assessment however, is that it is seldom possible to choose such an optimized static safety time, thus implying the recommendation of dynamic models in most cases.
Keywords	Inventory control, dynamic safety time, dynamic safety stock, time phased order point, service level, demand with seasonal variations

Innehållsförteckning

1	Inledning.....	1
1.1	Bakgrund.....	1
1.2	Problemformulering.....	3
1.3	Syfte.....	3
1.4	Avgränsningar.....	3
1.5	Rapportens disposition.....	4
2	Teoretisk referensram.....	5
2.1	Prognosmetoder.....	5
2.1.1	Enkel exponentiell utjämning.....	5
2.1.2	Prognosfel.....	6
2.2	Beordringssystem.....	7
2.2.1	Beställningssystem.....	7
2.2.2	Täckidsplanering.....	7
2.2.3	Tidsfasad beställningspunkt.....	8
2.2.4	Överdrag.....	9
2.3	Säkerhetslager.....	10
2.3.1	Servicenivåmått.....	12
2.3.2	Säkerhetslager baserat på $SERV_2$	13
2.4	Dynamisk optimering av säkerhetslager.....	15
2.4.1	Bestämmande av säkerhetslager genom simulering.....	16
2.4.2	Tidsbaserad avvikelse vid säkerhetslagerberäkning.....	17
2.4.3	Dynamisk planering av säkerhetslager.....	18
2.4.4	En ansats för dynamisk styrning av säkerhetslager under VMI.....	18
2.5	Standardavvikelse i beräkningar av säkerhetslager.....	19
2.5.1	Standardavvikelse baserat på efterfrågan respektive prognosfel.....	19
2.5.2	Stickprovsstorlek vid uppskattning av standardavvikelse.....	19
2.5.3	Ledtids efterfrågans standardavvikelse.....	20
2.5.4	Ledtidsjustering.....	21
2.5.5	Extremvärden.....	21
2.6	Vendor Managed Inventory – VMI.....	22
2.6.1	VMI och informationssystem.....	22

2.6.2	Fördelar med VMI för köparen.....	23
2.6.3	Fördelar med VMI för leverantören	23
3	Metod	25
3.1	Forskningsansats.....	25
3.2	Undersökningsmetod.....	25
3.3	Datainsamling.....	26
3.3.1	Genererad data.....	26
3.3.2	Verklig data	28
3.3.3	Extremvärden	28
3.4	Arbetsgång.....	29
3.4.1	Övergripande undersökningsmodell	30
3.4.2	Undersökningsupplägg.....	32
3.5	Giltighet.....	34
3.5.1	Validitet.....	34
3.5.2	Reliabilitet	34
3.5.3	Generaliserbarhet.....	35
3.5.4	Källkritik.....	35
4	Programvaran PipeChain	37
4.1	Typer av flöden för vilka PipeChain kan användas.....	37
4.1.1	Purchase Order Collaboration (POC).....	37
4.1.2	Sales Order Collaboration (SOC)	38
4.1.3	Vendor Managed Inventory (VMI) och Customer Vendor Managed Inventory (C VMI)	38
4.2	PipeChains beräkningsmodell.....	38
4.2.1	Parametrar	38
4.2.2	Grundläggande beordringsstrategi	39
4.3	Lagerstatus och servicenivå i PipeChain	40
4.4	Prognoser i PipeChain	41
4.5	Sambandet mellan de olika komponenterna i PipeChain-installationen	41
5	Dynamiska modeller för beräkning av säkerhetstid.....	43
5.1	Förutsättningar för samtliga modeller	43
5.2	Traditionell algoritm för beräkning av säkerhetslager.....	43

5.3	Traditionell algoritm med justeringar	45
5.3.1	Styrning mot varierande servicenivå istället för önskad servicenivå.....	46
5.3.2	Justering av säkerhetstiden grundat på senaste veckans prestation	46
5.4	Traditionell algoritm med justeringar och tidsbaserad avvikelse	48
5.5	Simulering av lämplig säkerhetstid	49
6	Resultat.....	51
6.1	Genererad efterfrågan med endast slumpmässiga variationer	51
6.1.1	Högrörliga artiklar	52
6.1.2	Lågrörliga artiklar	55
6.2	Genererad efterfrågan med systematiska variationer.....	58
6.2.1	Test med bättre prognos	60
6.3	Verklig efterfrågan.....	62
6.4	Tester med negativa säkerhetstider	63
7	Analys.....	65
7.1	Genererad efterfrågan med endast slumpmässiga variationer	65
7.1.1	Högrörliga artiklar	65
7.1.2	Lågrörliga artiklar	66
7.2	Genererad efterfrågan med systematiska variationer.....	67
7.2.1	Test med bättre prognos	68
7.3	Sammanfattande analys för genererad efterfrågan	68
7.3.1	Instabilitet orsakad av dynamisk säkerhetstid.....	69
7.4	Verklig efterfrågan.....	70
7.5	Tester med negativa säkerhetstider	70
7.6	Dynamisk säkerhetstid ur ett VMI-perspektiv	71
8	Slutsats	73
8.1	Framtida studier	74
	Referenser	77
	Bilaga 1 – 4	

1 Inledning

I detta kapitel introduceras bakgrunden och den problematik som initierat studien. Efter att syfte och avgränsningar definierats följer en kort beskrivning av rapportens disposition.

1.1 Bakgrund

Lagerstyrning har som syfte att åstadkomma lager som är lagom stora. Detta innebär att man vill kunna tillgodose efterfrågan i så hög utsträckning som möjligt utan att få brister, samtidigt som man vill hålla nere lagerhållningen eftersom lager binder kapital. Med hjälp av ett lagerstyrningssystem kan man hålla reda på hur, när och med vilken kvantitet ett lager ska fyllas på.¹ Med hjälp av moderna IT-system har konceptet *Vendor Managed Inventory* (VMI) blivit allt vanligare.² VMI innebär att leverantören ansvarar för påfyllningen av sin kunds lager och systemet har som mål att reducera lagernivåer och återfyllnadskostnader i hela försörjningskedjan.³ En viktig faktor inom VMI är utbytet av information, vilket främst innebär efterfrågedata.⁴

PipeChain – företaget och programvaran

Företaget PipeChain utvecklar och distribuerar en mjukvarulösning, med samma namn som företaget, för styrning av ett företags flödeskedja. PipeChain ägs av det svenska logistikonsultföretaget MA-system AB och grundades 1999. Idag har PipeChain mer än 300 aktiva kunder, med sammanlagt över 10 000 användare, i 30 länder. Kunderna består av såväl tillverkande som distribuerande företag i ett brett spektrum av branscher: elektronik, telekom, fordon, hälsa och sjukvård, detaljhandel samt förpackning.⁵

De effektiviseringsmöjligheter PipeChain erbjuder har som mål att användarna ska uppnå bättre lönsamhet och konkurrenskraft. Beroende på vilken mognadsgrad användarföretagets försörjningskedja har uppnått kan tillämpningen ske på tre olika nivåer.⁶ Den grundläggande nivån (*Information*) innebär att de traditionella orderprocesserna kan automatiseras, standardiseras och övervakas.⁷ I nästa nivå (*Insight*) kan man koordinera leveransflöden över hela världen och i flera steg av flödeskedjan, samtidigt som alla kan dela samma information i realtid. På detta sätt kan lagernivåer sänkas, brister undvikas och transportkostnader minskas.⁸ Den tredje nivån (*Innovation*) innebär att man frångår en orderdriven process till förmån för en efterfrågedriven, vilket uppnås genom leverantörs-påfyllning (VMI).⁹ Den lagerstyrningslogik som utförs i PipeChain bygger på ett tidskoncept, där lagernivåer mäts i täcktider istället för kvantiteter. Detta innebär bl.a. att man

¹ Axsäter, S. *Lagerstyrning*. Lund: Studentlitteratur, 1991, s. 13.

² Claassen, M.J.T., van Weele, A.J. och van Raaij, E.M. Performance outcomes and success factors of vendor managed inventory (VMI). *Supply Chain Management: An International Journal*, vol. 13, nr. 6, 2008, s. 406-414.

³ Wang, C., Ji, S., Shen, J. och Wei, W. Supply chain model in vendor managed inventory. *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, vol. 2, 2008, s. 2110-2113.

⁴ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

⁵ PipeChain. *PipeChain*, 2010. <http://www.pipechain.com/o.o.i.s/542> (Hämtad 2011-03-04)

⁶ PipeChain. *Solutions*, 2010. <http://www.pipechain.com/o.o.i.s/686> (Hämtad 2011-03-04)

⁷ PipeChain. *Information*, 2010. <http://www.pipechain.com/o.o.i.s/689> (Hämtad 2011-03-04)

⁸ PipeChain. *Insight*, 2010. <http://www.pipechain.com/o.o.i.s/688> (Hämtad 2011-03-04)

⁹ PipeChain. *Innovation*, 2010. <http://www.pipechain.com/o.o.i.s/687> (Hämtad 2011-03-04)

använder begreppen maxtid och säkerhetstid. Maxtid är den tidsperiods efterfrågan som lagret ska täcka direkt efter att en inleverans har inkommit. Säkerhetstid är den tidsperiods efterfrågan som lagret minst måste täcka, d.v.s. om lagernivån sjunker under den nivå som motsvarar prognostiserad efterfrågan under säkerhetstiden måste en beställning läggas.

Säkerhetstid

För att täcka upp för all osäkerhet som är förknippad med materialhantering och lagerstyrning behövs i princip alltid ett säkerhetslager. Säkerhetslagret uttrycks i PipeChains fall som en säkerhetstid. Den buffert som säkerhetslagret utgör bör ställas i relation till osäkerhetens omfattning på ett tämligen precist sätt, men trots att utvecklingen av ny teknologi och IT-lösningar har förenklats och effektiviserat styrningen av försörjningskedjor använder endast en tredjedel av Sveriges företag analytiska beräkningar baserade på servicegrad vid beräkning av säkerhetslager.¹⁰ Dessutom uppdateras variablerna för lagerstyrningen i mer än hälften av de svenska företagen endast en gång per år eller ännu mer sällan.¹¹ Att uppdatera säkerhetslagret så sällan måste betraktas som ett tämligen statistiskt synsätt på säkerhetslager. Även en analytiskt beräknad säkerhetslagernivå, som tagits fram vid ett tillfälle och sedan inte räknats om, kan med tiden visa sig vara direkt olämplig. Att utveckla säkerhetslagerberäkningen mot att bli mer dynamisk handlar således både om att använda analytiska beräkningar och att uppdatera dessa regelbundet.

Istället för analytiska beräkningar sker i många fall fortfarande uppskattningen av orderstorlekar och säkerhetslager baserat på erfarenhet, ett förfarande som på många sätt kan ifrågasättas då möjligheterna till uppföljning uteblir, vilket i sin tur innebär att företaget förlorar kontrollen över kostnader och kapitalbindning.¹² Det är inte ovanligt att ett så grovt mått som en viss procentsats av den totala lagervolymen¹³, eller ett fast antal dagars förbrukning för alla artiklar i företaget, används vid bestämning av säkerhetslagret. Avsaknaden av analytiska beräkningar vid bestämmandet av säkerhetslager är sannolikt något som också gäller för programvaran PipeChains användare.

Eftersom verkligheten inte är statisk och efterfrågan ändrar sig över tid (d.v.s. förhållandena är icke-stationära) är det troligt att en dynamisk beräkningsmodell, där säkerhetslagrets storlek kontinuerligt beräknas om och stäms av i förhållande till verkligheten, ger bättre resultat än ett statistiskt säkerhetslager. Det är dock först under senare år forskning inom dynamiska säkerhetslager har börjat bli utbredd och det har hittills endast publicerats ett fåtal vetenskapliga artiklar i ämnet.

¹⁰ Jonsson, P. och Mattsson, S-A. *Materialplaneringsmetoder i svensk industri – Användning och användningssätt*. PLAN, 2005.

<http://www.plan.se/Jonsson%20och%20Mattsson%20-%20Materialplaneringsmetoder%20i%20svensk%20industri.pdf>

¹¹ Jonsson och Mattsson, *Materialplaneringsmetoder i svensk industri – Användning och användningssätt*.

¹² Mattsson, S-A. Bullwhipeffekter och andra konsekvenser av att använda förenklade lagerstyrningsmetoder. *PLAN's forsknings- och tillämpningskonferens*, 2008, s. 117-132.

¹³ Li, Q-Y. och Li, S-J. A dynamic model of the safety stock under VMI. *2009 International Conference on Machine Learning and Cybernetics*, vol. 3, 2009, s. 1304-1308.

1.2 Problemformulering

PipeChains lagerstyrning är redan idag dynamisk i viss mån, i och med användandet av täcktid istället för antal enheter i lager. En idé man vill undersöka är om ytterligare dynamik i programmet kan ge än mer optimerade resultat i form av servicenivå och lagernivå. Idén är att denna dynamik ska utgöras av kontinuerliga uppdateringar av kontrollparametrarna säkerhetstid och maxtid. Dynamiken motiveras av att kontrollparametrarnas korrekta värden kan vara svåra att uppskatta. Problematiken gäller dels hur kontrollparametrarna från början överhuvudtaget ska sättas men också det faktum att förändrade efterfrågemönster över tid kan ge anledning att förändra parametrarna, något som kan vara svårt att åstadkomma på manuell väg.

Dynamiken bör underbyggas med analytiska beräkningar för att ge bästa resultat. Visionen är att i ett långsiktigt perspektiv kunna skapa ett så automatiserat program att man utan expertkunskaper kan ställa in och använda programmet och ändå erhålla optimal utdata. Detta genom att analytiska och dynamiska beräkningar automatiskt utförs. Att utveckla en modell för hur en sådan beräkning kan ske är därför ett första steg mot att i framtiden kunna uppnå denna vision.

Viktiga frågor att ta ställning till kring en sådan modell är hur dess giltighet påverkas av ett VMI-perspektiv samt i vilken grad ökad komplexitet leder till ökat värde. Vidare frågor som kan tas i beaktande är om det behövs olika kontrollprinciper för att styra olika produktklassifikationer, vilken hänsyn som behöver tas till osäkerhet i prognos och ledtid samt hur orderuppdelning och batchstorlek påverkas.

1.3 Syfte

Projektets syfte är att ta fram en modell för dynamisk uppdatering av säkerhetstid samt att testa modellen med hjälp av simuleringar för att undersöka giltigheten och beskriva vilken eventuell förbättring den nya modellen frambringar. Dessutom ska den framtagna modellens prestation bedömas ur dels ett klassificerings- och dels ett VMI-perspektiv.

1.4 Avgränsningar

Fast maxtid

Säkerhetstiden är nära kopplad till maxtiden och det skulle därför kunna vara intressant att undersöka hur dessa beror av varandra, särskilt som maxtiden även kan komma att bli föremål för dynamisk uppdatering längre fram. Genom att istället behandla maxtiden som fast kan man dock isolera påverkan av att variera säkerhetstiden, samtidigt som komplexiteten minskas. Därför görs avgränsningen att maxtiden behandlas som fast.

Fast ledtid

Variationer i ledtiden kan antas ha stor inverkan på hur stort säkerhetslagret ska vara. Det tillför dock komplexitet till undersökningen att införa en slumpmässigt varierande ledtid och dessutom tros en varierande ledtid påverka statiska och dynamiska modeller på samma sätt. Med anledning av detta anses det inte medföra något mervärde att använda något annat än en fast ledtid i detta projekt.

Lagerstyrning i PipeChain

I programmet PipeChain kan hänsyn tas till leveransscheman, som exempelvis kan ange vilken dag i veckan transporter ska ske m.m. Inga sådana företeelser tas i beaktande i detta projekt, utan avgränsningen görs att all lagerstyrning i PipeChain antas utföras med hjälp av den grundläggande logiken som beskrivs i kapitel 4.

Batchstorlekar och orderuppdelning

I praktiken är det viktigt att se hur batchstorleken påverkar styrningen av order. Detta ökar dock komplexiteten i styrningen så mycket att man i ett första skede bör bortse från detta för att kunna fastställa de grundläggande ramarna för en modell. Därför görs avgränsningen att hänsyn ej tas till batchstorlekar, utan artiklar antas beställas och levereras styckvis.

1.5 Rapportens disposition

I kapitel 2 görs en genomgång av det teoretiska ramverk som finns kring lagerstyrning i allmänhet och säkerhetslager i synnerhet. Kärnan utgörs av avsnitt 2.4, där befintlig forskning inom dynamisk uppdatering av säkerhetslager presenteras. Kapitlet avslutas med en redogörelse för konceptet VMI. Därefter beskrivs i kapitel 3 metoden för hur detta projekt utförts.

Kapitel 4 redogör för hur programmet PipeChain fungerar. I kapitel 5 beskrivs de modeller för dynamisk uppdatering av säkerhetstiden, som har tagits fram i denna studie och som föreslås kunna komplettera programmet i framtiden.

I kapitel 6 presenteras resultaten från undersökningarna av de framlagda modellförslagen. En analys av resultaten utförs i kapitel 7 tillsammans med en diskussion kring användningen och påverkan av en dynamisk modell. I kapitel 8 dras slutsatser och ges rekommendationer utifrån resultat och analys.

2 Teoretisk referensram

I detta kapitel presenteras en litteraturstudie som inledningsvis beskriver två inom lagerstyrning grundläggande koncept: prognosmetoder samt beordringssystem. För dessa koncept sätts fokus på metoder som används i PipeChain. Därefter följer en studie av säkerhetslager som inkluderar en kartläggning av senare tids forskning inom dynamisk optimering av säkerhetslager, vilken ligger till grund för utarbetandet av modeller för dynamisk uppdatering av säkerhetslager i denna studie. Ett delkapitel ägnas speciellt åt efterfrågans standardavvikelse då denna är en viktig del i säkerhetslagerbestämning. Kapitlet avslutas med en redogörelse för VMI (leverantörsstyrda lager).

2.1 Prognosmetoder

Inom lagerstyrning är i princip alltid ledtiden för en artikel så lång att man är tvungen att basera de beställningar man gör på en framtida tänkt efterfrågan. Eftersom den framtida, tänkta efterfrågan av naturliga skäl ytterst sällan är känd måste man istället förlita sig på en uppskattning av medelefterfrågan framåt i tiden, en s.k. prognos. I princip kan man ta fram prognoser på två sätt; genom att extrapolera historiska data eller genom att bygga på olika underliggande orsaker.¹⁴

Vid extrapolering av historiska data byggs prognosen upp av tillgänglig efterfrågedata och med hjälp av tidsserieanalys prognostiseras den framtida efterfrågan med statistiska metoder. När underliggande orsaker används för att bedöma efterfrågan är det istället ofta manuella bedömningar som ligger till grund för prognosen. Manuella bedömningar kan vara nödvändiga när det är frågan om speciella tillfälliga beroenden även om användningen av historiska data är det mest allmänt förekommande och mest betydelsefulla sättet att åstadkomma kortsiktiga prognoser.¹⁵

2.1.1 Enkel exponentiell utjämning

Exponentiell utjämning är en vanligt förekommande metod för att extrapolera historiska data i vilken hänsyn tas till att mer recenta värden bör ha ett högre informationsvärde och därför tillfästas större betydelse än äldre värden. Vid användning av exponentiell utjämning brukar man beteckna den vikt som ska kopplas till det färskaste efterfrågevärdet α , denna vikt kallas för utjämningskonstanten. Förenklat innebär metoden att om α t.ex. valts till 0,2 ska det mest recenta efterfrågevärdet få en vikt om 20 % medan den gamla prognosens vikt blir 80 %.¹⁶ Noterbart är att även om man tar hänsyn till tidigare efterfrågan så behöver man inte lagra ett närmast oändligt antal efterfrågevärden. Istället är det tillräckligt att känna till det senaste verkliga efterfrågevärdet samt föregående prognos.¹⁷ Matematiskt kan man alltså visa att exponentiell utjämning kan definieras enligt följande:

¹⁴ Axsäter, *Lagerstyrning*, s. 13.

¹⁵ Axsäter, *Lagerstyrning*, s. 14

¹⁶ Mattsson, S-A. och Jonsson, P. *Produktionslogistik*. Lund: Studentlitteratur, 2003, s. 156.

¹⁷ Mattsson och Jonsson, *Produktionslogistik*, s. 372-373.

$$P_{t+1} = \alpha \cdot D_t + (1 - \alpha) \cdot P_t \quad (2.1)$$

där

- P_{t+1} = prognostiserad efterfrågan för period $t + 1$
- D_t = verklig efterfrågan under period t
- P_t = prognostiserad efterfrågan för period t
- α = utjämningskonstanten

Valet av utjämningskonstanten α avgör hur känslig metoden blir för förändringar i efterfrågan. Ett litet värde på α innebär att man tar hänsyn till många tidigare efterfrågevärden medan ett större värde betyder det motsatta. I praktiken har det visat sig vara vanligt förekommande att utjämningskonstanten sätts till ett värde mellan 0,1 och 0,3 när exponentiell utjämning används som en metod för månadsvis prognosuppdatering.¹⁸

2.1.2 Prognosfel

När prognosfel beräknas används ofta begreppet MAD (*Mean Absolute Deviation*) som innebär det förväntade absolutvärdet av prognosfelet. Från början användes MAD istället för prognosfelets standardavvikelse eftersom MAD var enklare att beräkna och begreppet har hängt kvar sedan dess, trots att det beräkningstekniska behovet inte längre finns.¹⁹ I dagens läge kan man således lika gärna, eller hellre, använda standardavvikelsen. Sambandet mellan MAD och standardavvikelsen för normalfördelade slumpvariabler framgår av ekvation (2.2).²⁰

$$\sigma = \sqrt{\pi/2}MAD \approx 1,25MAD \quad (2.2)$$

MAD beräknas enligt ekvation (2.3).

$$MAD = E|X - \bar{X}| = \frac{1}{n} \sum_{i=1}^n |X_i - \bar{X}| \quad (2.3)$$

där \bar{X} betecknar medelvärdet av X . Ifall en prognos är medelvärdesriktig kan den sägas motsvara efterfrågans medelvärde och \bar{X} kan då ersättas med prognosen. Om efterfrågan betecknas E och prognosen P blir prognosens genomsnittliga absolutfel²¹:

$$MAD = \frac{1}{n} \sum_{i=1}^n |E_i - P_i| \quad (2.4)$$

Ett vanligt sätt att uppdatera MAD är att använda exponentiell utjämning. Vid månadsvis uppdatering kan $\alpha = 0,1$ vara rimligt eftersom prognosfelen kan ha ganska stora

¹⁸ Axsäter, *Lagerstyrning*, s. 20.

¹⁹ Axsäter, *Lagerstyrning*, s. 30-31.

²⁰ Axsäter, *Lagerstyrning*, s. 30-31.

²¹ Axsäter, *Lagerstyrning*, s. 30-31.

slumpvariationer mellan olika perioder. Beräkningarna ovan används då endast för att bestämma ett startvärde.²²

Oavsett om man använder MAD eller standardavvikelse för att få en storlek på prognosfelet blir måttet felet per period. Det som oftast är intressant i lagerstyrningssammanhang är prognosfelet under ledtiden. Hur standardavvikelsen kan ledtidsjusteras beskrivs i stycke 2.5.4 Ledtidsjustering. Ifall MAD används istället för standardavvikelsen måste standardavvikelsen per period först uppskattas med hjälp av ekvation (2.2) och därefter ledtidsjusteras.²³

2.2 Beordringssystem

För att avgöra när en order behöver läggas och hur stor kvantitet som ska beordras behövs ett lagerstyrningssystem. Detta system bestämmer genom beslutsregler med utgångspunkt i lagernivå, efterfrågan och kostnadsfaktorer när ett beslut om beordring ska tas.²⁴

2.2.1 Beställningspunktsystem

Den vanligaste strategin för beordring inom lagerstyrning är beställningspunktsystemet.²⁵ I detta system är meningen att beställningspunkten ska sättas på ett sådant sätt att den kvantitet som finns i lager vid tidpunkten också räcker för det behov som föreligger under ledtiden.²⁶ En vanlig typ av beställningspunktsystem är det så kallade (R, Q)-systemet. I detta system bestämmer man att en beställning med orderkvantiteten Q ska ske då lagerpositionen är lika med eller understiger R.²⁷

2.2.2 Täcktidsplanering

Ett system som är nära besläktat med (R, Q)-systemet är täcktidsplanering. Istället för att uttrycka behovet som en kvantitet uttrycker man det dock här som en tid. Täcktiden beräknas som aktuellt lagersaldo delat med medelefterfrågan per en viss tidsperiod, exempelvis vecka²⁸:

$$TT = \frac{\text{lagersaldo}}{\text{medelefterfrågan per tidsperiod}} \quad (2.5)$$

Säkerhetslagret i kvantitetsform blir översatt till tid²⁹:

$$\text{säkerhetslager} = \text{säkerhetstid} \cdot \text{efterfrågan per tidsperiod} \quad (2.6)$$

Den beslutsregeln som används vid täcktidsplanering kan enligt Mattsson formuleras³⁰:

²² Axsäter, *Lagerstyrning*, s. 31-32.

²³ Axsäter, *Lagerstyrning*, s. 33.

²⁴ Axsäter, *Lagerstyrning*, s. 40.

²⁵ Axsäter, *Lagerstyrning*, s. 40-44.

²⁶ Mattsson, S-A. *Effektiv materialstyrning – en handbok för att lyckas*. Helsingborg: Permatron, 2010, s. 103.

²⁷ Axsäter, *Lagerstyrning*, s. 41-42.

²⁸ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 103

²⁹ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 104

³⁰ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 104

*Planera in en ny order om täcktiden (TT) är mindre än ledtiden (LT) för återanskaffningen plus en säkerhetstid (ST). Sätt leveranstidpunkten till dagens datum plus artikelns ledtid.*³¹

Täcktdisplanering lämpar sig framförallt för styrning av artiklar med oberoende behov, speciellt där efterfrågan är jämn över tid. Metoden fungerar sämre för lågfrekventa, stora behov, långa genomloppstider och i miljöer med trender, säsongsvariationer eller stora systematiska efterfrågevariationer.³²

2.2.3 Tidsfasad beställningspunkt

En annan vanlig beordringsmetod är materialbehovsplanering. Metoden materialbehovsplanering vid oberoende efterfrågan används, som namnet antyder, främst vid oberoende efterfrågan, men kan även användas vid härledda behov. För att styra artiklar med härledda behov används annars materialbehovsplanering med behovsnedbrytning. Materialbehovsplanering vid oberoende efterfrågan kallas även tidsfasad beställningspunkt därför att en ny order läggs när lagersaldot gått ner till en nivå som beräknas inte räcka längre än ledtiden. Detta skiljer sig från ett traditionellt system där en ny order läggs när lagersaldot går under beställningspunkten. I materialbehovsplanering är det enkelt att ta hänsyn till säsong och trend, eftersom efterfrågan är uppdelad i perioder.³³

Beslutsregeln vid materialbehovsplanering vid oberoende efterfrågan är enligt Mattsson:

*Planera in en ny order för inleverans vid första nettobehov. Beräkna beställningstidpunkten som leveranstidpunkten minskad med artikelns ledtid.*³⁴

Nettobehov innebär att beräknad lagertillgång är negativ.

För att få skydd mot brister som beror på osäkerhet kan antingen säkerhetslager eller säkerhetstid användas i materialbehovsplaneringen. Ett sätt att använda säkerhetstid är att, när man bestämmer leveranstidpunkten, först hitta den tidpunkt då första nettobehov uppstår och sedan backa säkerhetstiden. Beslutsregeln blir då i modifierad form:

*Planera in en ny order för inleverans vid tidpunkten för första nettobehov minskad med säkerhetstiden. Beräkna beställningstidpunkten som leveranstidpunkten minskad med artikelns ledtid.*³⁵

Materialbehovsplaneringen kan vara transaktionsorienterad eller periodisk och orderkvantiteten kan vara fast eller bygga på ett visst antal perioders nettobehov. Dessa alternativ ger upphov till fyra olika varianter av metoden:

³¹ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 104

³² Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 106.

³³ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 109-113.

³⁴ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 110.

³⁵ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 113.

Tabell 1 Olika beställningssystem inom materialbehovsplanering

	Transaktionsorienterad	Periodisk
Fast orderkvantitet	(-, 0, Q)-system	(I, 0, Q)-system
Antal perioders nettobehov	(-, 0, S)-system	(I, 0, S)-system

Variabeln I betecknar att behovsberäkning sker periodiskt med detta givna intervall. Ett streck innebär således att behovsberäkningen sker transaktionsvis. Siffran 0 i mittenpositionen innebär att en ny order skapas då lagret blir mindre än 0. Q innebär en fast orderkvantitet, medan S är en återfyllnadsnivå. I systemen med variabeln S beräknas orderkvantiteten som summan av nettobehoven under en viss täcktid.³⁶

Periodisk inspektion

Vid periodisk inspektion bör det ingående saldot minskas med medelefterfrågan under I/2 dagar, det vill säga den genomsnittliga tidsperiod som har gått sedan senaste förbrukning. Även ledtidjustering av standardavvikelsen påverkas av periodisk inspektion. Istället för att bara justera för ledtiden, justerar man för ledtiden plus inspektionsperioden, se stycke 2.5.4. Ledtidjustering.³⁷

2.2.4 Överdrag

Om uttag från lager alltid sker med en enhet i taget kommer det lagersaldo som innebär att en ny beställning ska läggas alltid kunna nås med exakthet. I verkligheten sker dock uttag oftast om flera enheter åt gången och därför kan beställningspunkten komma att underskridas. Det antalet enheter med vilket beställningspunkten underskridits kallas överdrag och kan alltså variera mellan noll och den största uttagstransaktion som skett minus 1. Överdragens existens gör att den kvantitet som beställningspunkten är dimensionerad för kan bli för liten vilket leder till brister och lägre servicenivå än vad man önskat.

Problemet med överdrag har uppmärksammats av bl.a. Hill, som härleder approximativa beräkningar för att ta hänsyn till problemet,³⁸ och Janssen et al. som påpekar att överdraget har en betydande inverkan på systemets prestanda vid sammansatta efterfrågeprocesser.³⁹ Mattsson visar genom simulering att överdraget kan ha stor betydelse i förhållande till beställningspunkten samt att simulerade värden på överdrag stämmer bra överens med teoretiskt beräknade dito.⁴⁰

³⁶ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 110-111.

³⁷ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 110-111.

³⁸ Hill, R.M. Stock control and the undershoot of the reorder level. *Journal of the Operational Research Society*, vol. 39, nr. 2, 1988, s. 173-181.

³⁹ Janssen, F., Heuts, R. och de Kok, T. Theory and methodology on the (R, s, Q) inventory model when demand is modeled as a compound Bernoulli process. *European journal of operational research*, vol. 104, nr. 3, februari 1998, s. 423-436.

⁴⁰ Mattsson, S-A. *Materialstyrningsmodeller med hänsyn tagen till överdrag och olika efterfrågefördelningar*. Lund: Lunds universitet, 2007, s. 27-30.

Ett enkelt sätt att lösa problemet som överdrag ger upphov till är att sikta mot en högre servicenivå än man verkligen behöver. Det finns dock flera nackdelar med denna enkla lösning, t.ex. att servicenivån i hög grad måste baseras på intuition, att en enskild artikel inte nödvändigtvis får rätt servicenivå och att man får en spridning av erhållna servicenivåer. En mer sofistikerad lösning är att försöka ta hänsyn till överdraget genom beräkningar. Dessa beräkningar kan i regel göras med hjälp av den information som normalt finns i ett företags affärssystem, dock förutsätter beräkningarna ofta att speciella antaganden är uppfyllda, såsom en inte alltför låg eller hög variationskoefficient eller avsaknad av autokorrelation.⁴¹ Vid beräkning av överdrag måste man alltså först verifiera att aktuella antaganden är uppfyllda vilket inte alltid behöver vara fallet.

Överdrag kan beräknas enligt ekvation (2.7)

$$m_{\text{överdrag}} = \frac{\sigma^2 + \bar{E}^2}{2\bar{E}} - \frac{1}{2} \quad (2.7)$$

där \bar{E} är dagsefterfrågans medelvärde och σ dess standardavvikelse per dag. I ett täcktidsplaneringssystem med hänsyn till överdrag sker beställning ifall täcktiden sjunker under en viss gräns, se ekvation (2.8) där TT = täcktid, LT = leddid, ST = säkerhetstid och E är dagsefterfrågan.

$$TT \leq LT + ST + \frac{m_{\text{överdrag}}}{E} \quad (2.8)$$

Säkerhetstiden beräknas i ett sådant system enligt ekvation (2.9)

$$ST = k\sigma\sqrt{LT + 1}/E \quad (2.9)$$

där standardavvikelsen multipliceras med $\sqrt{LT + 1}$ för att motsvara en kombination av efterfrågans och överdragets standardavvikelser (de kan inte bara adderas eftersom de är beroende av varandra). Parametern k definieras enligt ekvation (2.18).⁴²

2.3 Säkerhetslager

All materialplanering är i större eller mindre utsträckning sammankopplad med osäkerheter av olika typer. Det gäller kvantitetsosäkerheter, t.ex. vad som kommer att säljas eller förbrukas i framtiden, men också tidsosäkerheter som t.ex. om leverantörer kommer att leverera beställda kvantiteter i rätt tid.⁴³ För att kunna hantera dessa osäkerheter håller man säkerhetslager som kan fungera som en buffert och göra att efterfrågan kan fortsätta att tillfredsställas vid avvikelser från förväntade värden.⁴⁴ Det är viktigt att säkerhetslagret

⁴¹ Mattsson, *Materialstyrningsmodeller med hänsyn tagen till överdrag och olika efterfrågefördelningar*, s. 18-19.

⁴² Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, avsnitt C92.

⁴³ Mattsson och Jonsson, *Produktionslogistik*, s. 473.

⁴⁴ Silver, E.A. och Peterson, R. *Decision Systems for Inventory Management and Production Planning*. New York: Wiley, 1979, s. 253.

endast används för att täcka upp för denna osäkerhet och inte för att väga upp systematiska prognosfel.⁴⁵

Eftersom hållandet av säkerhetslager innebär kapitalbindning är det viktigt att kunna dimensionera ett säkerhetslager till en rimlig nivå. Vad som är en rimlig nivå beror på vad man vill åstadkomma, t.ex. i form av minimerade lagerhållningskostnader eller en minsta servicenivå.⁴⁶

Silver och Peterson nämner fem övergripande beslutsregler för hur korrekta beslut om säkerhetslager ska tas⁴⁷:

1. Etablering av säkerhetslager genom användandet av en gemensam faktor

Det här tillvägagångssättet innebär oftast att man använder en gemensam tidsåtgång som grund för säkerhetslagret för varje artikel. Detta simplifierade synsätt har i regel stora brister. Den gemensamma faktorn kan vara att sätta en säkerhetstid som motsvarar ledtiden utan att ta hänsyn till osäkerheten i ledtid för varje artikel.

2. Säkerhetslager baserat på kostnaden för brister

Genom att explicit eller implicit bestämma kostnaden för en brist bestäms säkerhetslagret så att brist- och lagerhållningskostnaden blir så låg som möjligt.

3. Säkerhetslager baserat på en servicenivå

En kontrollparameter benämnd servicenivå introduceras och används som ett villkor för hur mycket säkerhetslager som ska hållas. T.ex. kan servicenivån vara att ett företag bestämmer sig för att 95 procent av all efterfrågan ska kunna tillfredsställas direkt från lager.

4. Säkerhetslager baserat på effekterna av att inte uppfylla sina åtaganden för kommande efterfrågan

Denna metod för att modellera effekterna av brister innebär i princip att framtida efterfrågan görs till en funktion av den service som just nu erbjuds. Om än tilltalande är denna metod i praktiken väldigt svår att implementera. Det finns dock exempel på hur företag implicit använt idén och lyckats få väldigt höga servicenivåer.

5. Säkerhetslager baserat på aggregerade överväganden

Detta generella tillvägagångssätt innebär i princip att ett visst säkerhetslager bestäms för varje artikel fördelat på lämpligt sätt tills dess att en given budget för hela artikelgruppen är använd. Därefter siktar man på att erbjuda bästa möjliga

⁴⁵ Mattsson, S-A. *Standardavvikelser för säkerhetslagerberäkningar*. Lund: Next generation innovative logistics (NGIL), 2007, s. 3.

⁴⁶ Mattsson, *Standardavvikelser för säkerhetslagerberäkningar*, s. 3.

⁴⁷ Silver och Peterson, *Decision Systems for Inventory Management and Production Planning*, s. 260-266.

sammanvägda service för hela artikelgruppen genom att försöka minimera den förväntade bristkostnaden.

De ovannämnda angreppssätten har alla styrkor och svagheter som gör dem lämpliga i olika situationer och en bedömning måste göras från fall till fall.⁴⁸ Gemensamt för metoderna är att för att kunna göra en egentlig beräkning av säkerhetslagrets storlek behöver man utgå från antingen en bristkostnad eller ett visst krav på service. Generellt sett är det enklare och mer lämpligt att utgå från en viss servicenivå,⁴⁹ alltså angreppssätt nummer 3 ovan.

Själva beräkningarna av säkerhetslager kan ske på en mängd olika sätt. Zizka identifierar 27 olika analytiska metoder för beräkning av säkerhetslager som delas in i två grupper. Den ena gruppen av metoder (24 stycken) beräknar säkerhetslagret till en fix nivå medan den andra gruppen (3 stycken) använder någon form av variabelt säkerhetslager. Enligt Zizka lämpar sig den förra gruppen av metoder för efterfrågan som är någorlunda stationär över tiden medan den senare gruppen kan användas i fall med icke-stationär efterfrågan. Utöver de analytiska metoderna presenterar Zizka även en metod för att genom simulering bestämma en lämplig nivå på säkerhetslagret givet uppfyllandet av en viss önskad servicegrad.⁵⁰

Efter en kort introduktion till olika servicenivåmått kommer de mest traditionella och i teorin väletablerade beräkningarna av säkerhetslager presenteras. I nästa avsnitt (2.4 Dynamisk optimering av säkerhetslager) redogörs för ett antal tillvägagångssätt för att dynamiskt uppdatera säkerhetslager.

2.3.1 Servicenivåmått

En servicenivå kan definieras på många olika sätt. Definitionen skulle t.ex. kunna utgå från att en kunds medelväntetid ska vara mindre än ett bestämt antal dagar, men vanligtvis uttrycks servicenivån istället som en sannolikhet, dvs. ett tal mellan noll och ett. Vad som alltid är viktigt är att man inom en organisation preciserar servicenivån på ett väldefinierat och otvetydigt sätt. Dessutom är det viktigt att man kan följa upp den service man faktiskt åstadkommer. Eftersom man normalt inte har samma servicekrav för alla produkter, men det samtidigt är alltför omständligt att ha en servicenivå för varje enskild artikel, fastställer man oftast en viss servicenivå för en bestämd grupp av artiklar.⁵¹

SERV₁

SERV₁ brukar definieras som sannolikheten att inte få brist under en ordercykel och kan därmed även sägas uttrycka sannolikheten att en viss leverans kommer fram i tid. Även om SERV₁ är lätt att räkna med har den betydande nackdelar som är förknippade med att ingen hänsyn tas till leveranskvantiteten, dvs. orderkvantiteten. Med anledning av detta riskerar

⁴⁸ Silver och Peterson, *Decision Systems for Inventory Management and Production Planning*, s. 260-266.

⁴⁹ Axsäter, *Lagerstyrning*, s. 68.

⁵⁰ Zizka, M. The Analytic Approach vs. the Simulation Approach to Determining Safety Stock. *Problems & Perspectives in Management*, nr. 3, 2005, s. 119-128.

⁵¹ Axsäter, *Lagerstyrning*, s. 68.

$SERV_1$ antingen att bli missvisande hög eller missvisande låg beroende på om orderkvantiteten är liten eller stor.⁵²

$SERV_1$ = sannolikheten att inte få brist under en ordercykel

$SERV_2$

Jämfört med $SERV_1$ är $SERV_2$ (även kallad fyllnadsgrad) lite besvärligare att använda, samtidigt som begreppet i regel är ett bra mått på den verkliga kundservicen. Om efterfrågan alltid antas omfatta en enhet i taget kan $SERV_2$ tolkas som sannolikheten att en kund får vänta. Allmänt kan man ge $SERV_2$ följande definition:

$SERV_2$ = andel av efterfrågan som kan hämtas direkt från lager⁵³

$SERV_3$

$SERV_3$ (*ready rate*) definieras som den andel av tiden som ineliggande lager är positivt. I fallet med Poissonfördelad efterfrågan och under antagande om att en kund beställer en enhet i taget är $SERV_3$ ekvivalent med $SERV_2$. Om en kund däremot kan beställa många enheter åt gången kan det hända att lagret inte är tillräckligt för att tillfredsställa en stor order, vilket alltså inte avspeglas i $SERV_3$. Om lagret oftast innehåller ett litet antal enheter kan $SERV_3$ vara hög samtidigt som $SERV_2$ är låg om några av kunderna beställer stora kvantiteter. Begreppet kan definieras:

$SERV_3$ = andel av tiden med positivt ineliggande lager⁵⁴

2.3.2 Säkerhetslager baserat på $SERV_2$

Med utgångspunkt i $SERV_2$, d.v.s. andel av efterfrågan som kan levereras direkt från lager, kan en lämplig nivå på säkerhetslagret bestämmas. För en viss leveranskvantitet Q finns en viss genomsnittlig brist. Här förutsätts att bristen restnoteras och levereras när inleveransen kommer. Bristen uppstår då medelvärdet för efterfrågan under ledtiden u överstiger SL , närmare bestämt blir bristen $u - SL$ då $SL \leq u \leq SL + Q$, eller Q om $u > SL + Q$. Om avvikelsen överstiger $SL + Q$ ska nämligen denna täckas med nästföljande orderkvantitet och därför hör avvikelsen inte till den studerade ordercykeln.⁵⁵

För att kunna utföra beräkningarna måste man känna till vilken statistisk fördelning efterfrågan kan anses ha, vilket kan avgöras med statistisk analys.⁵⁶ Det vanligaste är att man utgår ifrån att den är normalfördelad, vilket ofta är fallet vid hög efterfrågan⁵⁷ eftersom de verkliga variationerna i efterfrågan består av en mängd sammanlagda oberoende slumpmässiga händelser. Normalfördelningen ger dock inte en korrekt beskrivning då efterfrågevariationerna är stora, vid lågrörliga artiklar samt vid korta ledtider, då det finns

⁵² Axsäter, *Lagerstyrning*, s. 68.

⁵³ Axsäter, *Lagerstyrning*, s. 68.

⁵⁴ Axsäter, S. *Inventory Control*. New York: Springer, 2006, s. 95.

⁵⁵ Axsäter, *Lagerstyrning*, s. 71.

⁵⁶ Mattsson och Jonsson, *Produktionslogistik*, s. 475.

⁵⁷ Axsäter, *Lagerstyrning*, s. 65.

risk för negativa värden på efterfrågan.⁵⁸ Det är vanligt förekommande att ett antagande om normalfördelning görs om medelefterfrågan under ledtiden är större än 10 enheter eller större än standardavvikelsen och dessutom större än 5.⁵⁹ Ett annat antagande som förekommer är att om medelefterfrågan är större än standardavvikelsen multiplicerad med en viss faktor kan efterfrågan anses normalfördelad. Faktorns storlek varierar mellan olika författare, exempelvis kan den vara 1,7 eller 2,5 eller så stor som 5.⁶⁰ Då uttaget från lagret är litet, en eller ett par enheter, anses istället Poissonfördelning återspegla verkligheten betydligt bättre än normalfördelning.⁶¹ Fortsättningsvis beskrivs dock hur säkerhetslager kan beräknas vid normalfördelad efterfrågan.

Låt standardavvikelsen under ledtiden betecknas σ_{LT} . Vid normalfördelad efterfrågan under ledtiden får då avvikelsen frekvensfunktionen $\frac{1}{\sigma_{LT}}\varphi\left(\frac{u}{\sigma_{LT}}\right)$. Axsäter härleder utifrån denna funktion ordercykelns genomsnittliga bristkvantitet enligt följande:⁶²

$$\begin{aligned} BRIST &= \int_{SL}^{SL+Q} (u - SL) \frac{1}{\sigma_{LT}} \varphi\left(\frac{u}{\sigma_{LT}}\right) du + Q \int_{SL+Q}^{\infty} \frac{1}{\sigma_{LT}} \varphi\left(\frac{u}{\sigma_{LT}}\right) du \\ &= \int_{SL}^{\infty} (u - SL) \frac{1}{\sigma_{LT}} \varphi\left(\frac{u}{\sigma_{LT}}\right) du - \int_{SL+Q}^{\infty} (u - SL - Q) \frac{1}{\sigma_{LT}} \varphi\left(\frac{u}{\sigma_{LT}}\right) du \end{aligned} \quad (2.10)$$

Genom införande av beteckningen

$$G(v) = \int_v^{\infty} (x - v) \varphi(x) dx = \varphi(v) - v(1 - \Phi(v)) \quad (2.11)$$

som brukar betecknas *loss function*⁶³ eller servicefunktion⁶⁴ kan *BRIST* uttryckas enligt:⁶⁵

$$BRIST = \sigma_{LT} G\left(\frac{SL}{\sigma_{LT}}\right) - \sigma_x G\left(\frac{SL + Q}{\sigma_{LT}}\right) \quad (2.12)$$

Om Q är stor blir den andra termen nära noll och därför är det vanligt att man använder följande approximation:⁶⁶

$$BRIST = \sigma_{LT} G\left(\frac{SL}{\sigma_{LT}}\right) \quad (2.13)$$

⁵⁸ Mattsson, S-A. *Efterfrågefördelningar för bestämmande av säkerhetslager*. Lund: Next generation innovative logistics (NGIL), 2007, s. 9 och 29-30.

⁵⁹ Mattsson och Jonsson, *Produktionslogistik*, s. 475

⁶⁰ Mattsson, *Efterfrågefördelningar för bestämmande av säkerhetslager*, s. 9.

⁶¹ Mattsson, *Efterfrågefördelningar för bestämmande av säkerhetslager*, s. 29-30.

⁶² Axsäter, *Lagerstyrning*, s.71.

⁶³ Axsäter, *Inventory Control*, s. 91.

⁶⁴ Mattsson och Jonsson, *Produktionslogistik*, s. 479.

⁶⁵ Axsäter, *Lagerstyrning*, s. 72.

⁶⁶ Axsäter, *Lagerstyrning*, s. 72.

Eftersom $BRIST/Q$ är andelen brist gäller att $BRIST/Q = 1 - SERV_2$. Genom detta fås en ekvation för att bestämma säkerhetslagret enligt:⁶⁷

$$G\left(\frac{SL}{\sigma_{LT}}\right) - G\left(\frac{SL+Q}{\sigma_{LT}}\right) = \frac{Q}{\sigma_{LT}}(1 - SERV_2) \quad (2.14)$$

Om Q är stor och $BRIST$ approximerats enligt ovan erhålles sambandet:⁶⁸

$$G\left(\frac{SL}{\sigma_{LT}}\right) = \frac{Q}{\sigma_{LT}}(1 - SERV_2) \quad (2.15)$$

Säkerhetslagret kan alltså uttryckas som:⁶⁹

$$SL = \sigma_{LT} \cdot G^{-1}\left(\frac{Q}{\sigma_{LT}}(1 - SERV_2)\right) = \sigma_{LT} \cdot k \quad (2.16)$$

Eftersom servicefunktionen p.g.a. sin matematiska definition inte tillåter ett enkelt uttryck av inversen kan man istället använda ett uttryck där inversen approximeras med k , som beräknas med hjälp av variabeln z enligt följande formler:⁷⁰

$$z = \sqrt{\ln\left(\frac{25}{G\left(\frac{SL}{\sigma_{LT}}\right) \cdot G\left(\frac{SL}{\sigma_{LT}}\right)}\right)} \quad (2.17)$$

$$k = \frac{a_0 + a_1 \cdot z + a_2 \cdot z^2 + a_3 \cdot z^3}{b_0 + b_1 \cdot z + b_2 \cdot z^2 + b_3 \cdot z^3 + b_4 \cdot z^4} \quad (2.18)$$

där	$a_0 = -5,3925569$	$b_0 = 1$
	$a_1 = 5,6211054$	$b_1 = -7,2496485 \cdot 10^{-1}$
	$a_2 = -3,8836830$	$b_2 = 5,07326622 \cdot 10^{-1}$
	$a_3 = 1,0897299$	$b_3 = 6,69136868 \cdot 10^{-2}$
		$b_4 = -3,29129114 \cdot 10^{-3}$

2.4 Dynamisk optimering av säkerhetslager

Traditionellt sett har man inom lagerstyrning bara bestämt en önskad nivå på säkerhetslagret genom att uppskatta osäkerhet och sedan applicerat den för kommande

⁶⁷ Axsäter, *Lagerstyrning*, s. 72.

⁶⁸ Axsäter, *Lagerstyrning*, s. 72.

⁶⁹ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 206.

⁷⁰ Mattsson, *Effektiv materialstyrning – en handbok för att lyckas*, s. 206.

tidsperioder.⁷¹ Beroende på hur bra man lyckats uppskatta riskerna och hur efterfrågeutvecklingen sett ut över tiden kan därmed resultaten av lagerstyrningen komma att variera kraftigt. I mer avancerade metoder kan man istället för att bara sätta en bestämd nivå på säkerhetslagret dynamiskt uppdatera säkerhetslagernivån när olika villkor ändras. De företag som faktiskt använder analytiska beräkningar för säkerhetslagret gör rimligen också i något tidsintervall uppdaterande beräkningar, i praktiken ofta en gång i månaden.⁷² Beroende på vilket tidsperspektiv man har kan detta betraktas som en enkel form av dynamisk uppdatering. En annan enkel form av dynamik är att uttrycka säkerhetslagret som en säkerhetstid.

Vid stationär efterfrågan är en fast säkerhetslagernivå i princip tillräcklig. Det är först vid icke-stationär efterfrågan som ett dynamiskt beräknat säkerhetslager verkligen kommer till sin rätt.^{73, 74} Av naturliga skäl gör ju fluktuationerna här att olika säkerhetslagernivåer blir lämpliga vid olika tidpunkter.

2.4.1 Bestämmande av säkerhetslager genom simulering

Zizka presenterar en metod för bestämmande av säkerhetslager som tar sin utgångspunkt i en empirisk fördelning som beskriver efterfrågan hos en viss produkt. Ett antagande om att alla leveranser består av en fix kvantitet som inkommer i ett förutbestämt intervall görs och därefter körs en simulering över ett större antal återfyllnadscyklar. Genom att anta att inget säkerhetslager hålls överhuvudtaget uppstår under simuleringstiden brister vars storlekar och tidpunkter sparas. Det är logiskt att utgå från att säkerhetslagret inte ska vara större än den största brist som uppstår under simuleringens gång. Med denna nivå på säkerhetslagret fås en servicegrad på 100 %. Beroende på vilken servicegrad som önskas kan nu en nivå på säkerhetslagret enkelt härledas genom att utnyttja simuleringens data över brister.⁷⁵

Säkerhetslagerberäkningar enligt ovanstående kan utföras för såväl $SERV_1, SERV_2$ som $SERV_3$. Om en simulering över 1000 dagar t.ex. ger att brist uppstår 3 olika dagar om 20, 50 respektive 10 enheter skulle detta innebära att $SERV_3 = \frac{997}{100} = 99,7\%$ vid inget säkerhetslager alls. Om man i detta fall väljer att introducera ett säkerhetslager på 10 enheter blir $SERV_3 = 99,8\%$, vid 20 enheter blir $SERV_3 = 99,9\%$ och med ett säkerhetslager på över 50 enheter blir $SERV_3 = 100\%$.

Ett annat exempel på bestämmande av säkerhetslager genom simulering är den s.k. *Safety Stock Adjustment Procedure (SSAP)* som introduceras av Køhler och de Kok. Med denna teknik utgår man från tidsfasad beställningspunkt som materialplaneringsmetod för att genom simulering uppnå en viss målservicenivå. Metoden kräver att all efterfrågan som inte omedelbart kan tillfredsställas restnoteras men ställer inga krav på hur efterfrågan uppkommer eller prognostiseras. SSAP bygger på en approximation av lagerprocessens

⁷¹ Kanet, J. J., Gorman, M.F. och Stöblein, M. Dynamic planned safety stocks in supply networks. *International Journal of Production Research*, vol. 48, nr.22, 2010, s. 6859–6880.

⁷² Mattsson, *Standardavvikelser för säkerhetslagerberäkningar*, s. 2.

⁷³ Zizka, The Analytic Approach vs. the Simulation Approach to Determining Safety Stock.

⁷⁴ Kanet et al., Dynamic planned safety stocks in supply networks.

⁷⁵ Zizka, The Analytic Approach vs. the Simulation Approach to Determining Safety Stock.

empiriska fördelning som med hjälp av ett godtyckligt värde för säkerhetslagret genereras genom simulering. Med utgångspunkt i fördelningen bestäms en formel för den s.k. justeringskvantiteten som kan beräknas för $SERV_1$, $SERV_2$ eller $SERV_3$. Justeringskvantiteten används för att justera säkerhetslagret så att önskad servicegrad uppnås.⁷⁶

2.4.2 Tidsbaserad avvikelse vid säkerhetslagerberäkning

Krupp menar att man, genom att mäta avvikelser i tid istället för i kvantitet, kan ta hänsyn till trender och säsongsvariationer vid beräkning av säkerhetslager och lägger därför fram ett alternativ till ett fixt säkerhetslager, som syftar till att få ett säkerhetslager som följer förändringar över tiden. När säkerhetslagret bestäms som en konstant kvantitet finns risker såväl då efterfrågan går upp som när den minskar. Vid växande efterfrågan finns risk för brister i kundservice eftersom säkerhetslagrets storlek inte längre är tillräcklig. Då efterfrågan sjunker finns istället risk för överskottslager eftersom säkerhetslagret ligger kvar på samma nivå som då efterfrågan var hög, vilket i synnerhet blir ett problem då en produkt närmar sig slutet på sin livscykel. Med Krupps alternativa metod kommer istället säkerhetslagret minska i takt med efterfrågan och då efterfrågan dör ut kommer säkerhetslagret också att ha nått noll. Krupp har funnit att detta är det fall där hans metod ger allra störst fördelar.⁷⁷

Då efterfrågan är extremt instabil och då det förekommer stora hopp i efterfrågan kan ett varierande säkerhetslager dock innebära stora problem. När säkerhetslagret ändras förändras planeringen för den aktuella slutprodukten. Detta innebär även att planeringen för nedströms lageråterfyllnad påverkas och därigenom leveransscheman till leverantörerna. På detta sätt blir hela planeringssystemet ”nervöst”. Även den kortsiktiga kapacitetsplaneringen påverkas av variationerna i säkerhetslagrets storlek. Om säkerhetslagret minskar finns plötsligen tillgänglig kapacitet, medan ytterligare kapacitet måste tas i anspråk när säkerhetslagret ökar. Detta kan påverka analyser och beslut. För att kunna använda ett varierande säkerhetslager krävs att det finns flexibilitet i såväl planeringen av återfyllnad som kapacitetsplaneringen. Det måste bedömas i varje enskilt sammanhang ifall fördelarna i form av lägre lagerkostnader och högre servicenivå väger upp problemen och ”nervositeten” som kan uppstå.⁷⁸

Istället för att använda MAD (se stycke 2.1.2 Prognosfel) som grund vid beräkning av säkerhetslagret inför Krupp termen tidsbaserad MAD, som beräknas enligt ekvation (2.19).

$$TBM_n = \frac{1}{n} \sum_{i=1}^n \frac{|E_i - P_i|}{P_i} \quad (2.19)$$

⁷⁶ Køhler Gudum, C. och de Kok, T. G. *A Safety Stock Adjustment Procedure to Enable Target Service Levels in Simulation of Generic Inventory Systems*. Köpenhamn: Copenhagen Business School, Department of Management Science and Statistics, Preprint No 1, 2002.

⁷⁷ Krupp, J. Safety stock management. *Production and Inventory management journal*, 3rd qtr, 1997, s. 11-18.

⁷⁸ Krupp, Safety stock management.

Krupp presenterar först en traditionell beräkningsformel:

$$SL = k \cdot MAD\sqrt{LT} \quad (2.20)$$

Till skillnad från denna väljer han att istället beräkna säkerhetslagret för den kommande planeringsperioden t enligt ekvation (2.21).

$$SL_t = k \cdot TBM_n P_{t+1} \sqrt{LT} \quad (2.21)$$

Multiplikationen med prognosen för den efterföljande perioden (P_{t+1}) i ekvation (2.21) görs för att uppnå att säkerhetslagret vid slutet av tidsperiod t är lagom stort för att uppfylla den kommande periodens behov.⁷⁹

2.4.3 Dynamisk planering av säkerhetslager

För att få en än mer sofistikerad dynamisk modell introducerar Kanet et al. en styrningsmetod där man inte bara uppdaterar säkerhetslagernivån vid givna villkor utan också planerar för kommande tidsberoende förändringar i nivån, alltså framtida mål för säkerhetslagernivå.⁸⁰

Kanet et al. har genom sin forskning konstaterat att dynamiskt planerade säkerhetslager, jämfört med ett konstant säkerhetslager, ger bättre resultat både vad gäller lagernivå och service. I ett specifikt fall visar Kanet et al. att säkerhetslagernivån kunde sänkas med 14 procent med bibehållen nivå på fyllnadsgrad. Förutsättningarna för att dynamisk planering ska ge ett positivt resultat är att minst en parameter av efterfrågan och ledtiden är icke-stationär, eller att företagets önskade servicenivå varierar över tiden. Kanet et al. visar på att icke-stationäritet i efterfrågan och ledtid är mycket utbrett. Även i tillförlitliga och väldefinierade prognosmodeller är detta vanligt förekommande, vilket är ett bevis för att dynamisk planering av säkerhetslager kan vara av betydelse även för företag med välbeprövade prognosmodeller. Vidare konstaterar Kanet et al. att säsongsvariationer är det som driver företagets säkerhetslager och de har därför i sin forskning använt sig av ojusterad data som innehåller olika säsongsvariationer, för att inte det verkliga behovet av säkerhetslager ska dämpas.⁸¹

2.4.4 En ansats för dynamisk styrning av säkerhetslager under VMI

Li och Li har utvecklat en modell för dynamisk styrning av säkerhetslager, som främst är tänkt att användas i VMI-samarbeten. De beskriver ansatsen med dynamiska säkerhetslager som ett helt nytt forskningsområde. Enligt Li och Li gör en dynamisk styrning att säkerhetslagret överensstämmer bättre med det verkliga behovet av osäkerhetstäckning, eftersom en ostadig försäljning kommer alstra ett större säkerhetslager och vice versa.

⁷⁹ Krupp, Safety stock management.

⁸⁰ Kanet et al., Dynamic planned safety stocks in supply networks.

⁸¹ Kanet et al., Dynamic planned safety stocks in supply networks.

Grunden för den dynamiska kontrollen i den framtagna modellen är den dynamiska skillnaden mellan prognos och verklig efterfrågan.⁸²

2.5 Standardavvikelse i beräkningar av säkerhetslager

Som framgått av beskrivningen av säkerhetslagerberäkning är standardavvikelsen en mycket viktig komponent i beräkningarna. Den kan beräknas på många olika sätt och det är av stor betydelse för resultatet att standardavvikelsen blir så korrekt och noggrann som möjligt. Det är särskilt viktigt med en korrekt och noggrann beräkning av standardavvikelsen då ledtiden är kort, inleveranskvantiteten liten eller efterfrågevariationen stor, eftersom felkänsligheten är som störst i dessa fall.⁸³

2.5.1 Standardavvikelse baserat på efterfrågan respektive prognosfel

Standardavvikelsen kan baseras på hur efterfrågan historiskt sett varierat, eller hur prognosfelet varierat. Det sistnämnda är det vanligaste beräkningssättet i affärssystem. Dessa två mått kommer bara att vara lika ifall prognosen är medelvärdesriktig, vilket sällan är fallet. Exempelvis är en prognos beräknad med exponentiell utjämning eller glidande medelvärde inte medelvärdesriktig. Standardavvikelse som baserats på prognosfel är ungefär nio procent större än efterfrågestandardavvikelse.⁸⁴ I fortsättningen av detta avsnitt kommer efterfrågans standardavvikelse avses om inget annat anges.

2.5.2 Stickprovsstorlek vid uppskattning av standardavvikelse

Den standardavvikelse som används vid beräkningar av säkerhetslager är en uppskattning av ett antal historiska efterfrågevärden. Ju större stickprovet är desto närmare det riktiga värdet kommer uppskattningen att vara. För att få en noggrannhet där standardavvikelsen ligger inom en felmarginal på tio procent från den korrekta standardavvikelsen (som räknats fram med den maximalt tillgängliga stickprovsstorleken) krävs enligt Mattssons generella rekommendation minst fyrtio historiska efterfrågevärden i stickprovet. Denna siffra gäller för högrörliga artiklar (tre respektive tio order per dag). För lågrörliga artiklar (en order per två veckor respektive två månader) krävs minst åttio historiska efterfrågevärden. För en medelrörlig artikel (en order per två dagar) kan drygt femtio värden behövas. Mattssons resultat från undersökningar av fallföretag visar att det i dessa fall behövs mellan cirka 50 och 100 värden i stickproven, beroende på rörligheten hos artiklarna. Ju mer efterfrågan varierar, vilket motsvaras av ju mer lågrörlig artikeln är, desto större stickprov behövs.⁸⁵

Standardavvikelse uppskattad med stickprov varierar över tiden när nya beräkningar görs i varje ny period, även om inga egentliga förändringar föreligger i efterfrågans struktur och medelvärde. Detta innebär i sin tur att säkerhetslager och erhållna servicenivåer varierar över tiden.⁸⁶ Stora variationer i servicenivån är inte önskvärt då det innebär en större total

⁸² Li och Li, A dynamic model of the safety stock under VMI.

⁸³ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 15.

⁸⁴ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 43-46

⁸⁵ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 26 och 28-30

⁸⁶ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 26-27 och 30-33

kapitalbindning.⁸⁷ Störst variationer i servicenivån uppstår vid stora efterfrågevariationer och lång ledtid. Även i detta fall är stora stickprov lösningen på problemet. Ju större stickprov desto mindre varierar standardavvikelsen, och följaktligen servicenivån, över tiden.⁸⁸

2.5.3 Ledtidsefterfrågans standardavvikelse

Vid beräkning av säkerhetslager är det standardavvikelsen för efterfrågan under ledtiden som ska användas. Denna kan man få fram genom att antingen direkt beräkna standardavvikelsen för efterfrågan under ledtiden, eller genom att beräkna standardavvikelsen för efterfrågan av en viss annan period och därefter ledtidsjustera den.⁸⁹ Om exempelvis efterfrågan per dag är känd och ledtiden är tre dagar innebär det första beräkningssättet, direkt beräkning av standardavvikelsen för efterfrågan under ledtiden, att efterfrågan för tre dagar summeras i ej överlappande intervall. Därefter beräknas standardavvikelseerna för dessa värden (som alltså utgör efterfrågan under ledtidsperioder). Det andra sättet att räkna innebär att standardavvikelsen för efterfrågan per dag beräknas och därefter görs en justeringen för att få fram standardavvikelsen för efterfrågan under ledtiden. Denna ledtidsjustering innebär en approximation, men Mattsson drar i sina försök slutsatsen att skillnaden mellan dessa sätt att räkna är försumbar, så länge stickproven är tillräckligt stora.⁹⁰ Vidare beskrivning av ledtidsjustering finns i stycke 2.5.4 Ledtidsjustering.

Ett tredje sätt att utföra beräkningen av standardavvikelsen är att, istället för att utgå från dagsefterfrågan, göra en omräkning till månadsefterfrågan varpå standardavvikelsen för dessa värden beräknas. Till sist görs motsvarande ledtidsjustering som i fallet med dagsefterfrågan. Mattssons jämförelse av standardavvikelse baserad på dagsefterfrågan respektive månadsefterfrågan visar, liksom jämförelsen mellan standardavvikelse för ledtidsefterfrågan respektive dagsefterfrågan, att då stickproven är tillräckligt stora är skillnaden mellan de två beräkningssätten försumbar.⁹¹ Det finns dock teoretiska nackdelar med att använda en månadsefterfrågan. Eftersom stickproven måste vara stora för att ge hög noggrannhet innebär månadsefterfrågan att flera års data måste användas för att komma upp i tillräckligt stora stickprov. Sannolikheten är stor att så gammal efterfrågedata inte är representativ för hur efterfrågan ser ut idag och i framtiden, eftersom såväl kund- och orderstruktur som lagerstruktur troligtvis har förändrats under den långa tidsperiod som förflutit.⁹² Mattssons rekommendation är därför att då det är möjligt basera standardavvikelsen på dagsefterfrågan istället för månadsefterfrågan.⁹³

⁸⁷ Mattsson, *Materialstyrningsmodeller med hänsyn tagen till överdrag och olika efterfrågefördelningar*, s. 3.

⁸⁸ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 26-27 och 30-33

⁸⁹ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 37-38.

⁹⁰ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 40.

⁹¹ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 40.

⁹² Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 37.

⁹³ Mattsson, *Standardavvikelse för säkerhetslagerberäkning*, s. 48.

2.5.4 Ledtidsjustering

Standardavvikelsen för ledtiden fås från standardavvikelsen för en period genom⁹⁴:

$$\sigma_{LT} = \sigma LT^c \quad (2.22)$$

där c är en konstant mellan 0,5 och 1. Ifall alla slumpvärden är statistiskt oberoende används $c = 0,5$. Formeln för standardavvikelsen under ledtiden blir då⁹⁵:

$$\sigma_{LT} = \sigma\sqrt{LT} \quad (2.23)$$

Denna formel används ofta i praktiken och är godtagbar så länge standardavvikelsen baseras på efterfrågan och efterfrågan i olika perioder ej är beroende av varandra. Ifall standardavvikelsen beräknas utifrån prognosfel istället för historisk efterfrågan och exponentiell utjämning eller motsvarande prognosmetod används uppstår korrelation. Även i de fall prognosmetoden inte tar hänsyn till systematiska variationer kan korrelation uppstå.⁹⁶

Då periodisk inspektion tillämpas (se stycke 2.2.3 Tidsfasad beställningspunkt) måste hänsyn även tas till inspektionsperioden vid tidsjustering av standardavvikelsen. Om inspektionsperioden betecknas I utvidgas ekvation (2.23) till ekvation (2.24).⁹⁷

$$\sigma_{LT} = \sigma\sqrt{LT + I} \quad (2.24)$$

2.5.5 Extremvärden

Det är vanligt att det bland efterfrågevärden finns extremvärden som är exceptionellt mycket större än övriga värden. Exempelvis kan dessa extremvärden ha uppstått genom en unikt stor order, eller genom att slumpen gjort att några relativt stora, men var för sig inte onormala, order samlats till en och samma tidpunkt. Extremvärden kan också uppstå efter leveranssvårigheter eller vara rena felaktigheter vid registrering. Felaktiga extremvärden bör naturligtvis inte påverka beräkningarna av säkerhetslager. Inte heller verkliga extremvärden bör påverka, eftersom sådan extrem efterfrågan inte bör skötas via lagerstyrningen utan genom specialhantering. Då extremvärden exkluderas från beräkningen av standardavvikelsen, som ligger till grund för säkerhetslagret, minskar såväl standardavvikelsen som säkerhetslagret samtidigt som den erhållna servicenivån ökar.⁹⁸

Mattsson redogör för olika sätt att eliminera extremvärden. En metod är att manuellt plocka bort en order som anses vara exceptionell. En annan utgörs av en kontroll av huruvida en periods utleveransvolym kan anses vara representativ för normalt varierande efterfrågan, för att på så sätt även komma åt en slumpmässig anhopning av relativt stora order. Perioder med utleveransvolym över en viss kontrollgräns exkluderas. En annan variant är att värdena

⁹⁴ Axsäter, *Lagerstyrning*, s. 33.

⁹⁵ Axsäter, *Lagerstyrning*, s. 33.

⁹⁶ Mattsson, *Standardavvikelser för säkerhetslagerberäkning*, s. 38.

⁹⁷ Axsäter, *Lagerstyrning*, s. 41.

⁹⁸ Mattsson, *Standardavvikelser för säkerhetslagerberäkning*, s. 17-23.

över kontrollgränsen istället för att tas bort ersätts med kontrollgränsens värde. Den metod Mattsson i sin undersökning använder för eliminering av extremvärden är att ersätta efterfrågevärden över en viss kontrollgräns med medelefterfrågan per dag. Kontrollgränsen bestäms till medelefterfrågan per dag plus tre gånger standardavvikelsen per dag.⁹⁹

2.6 Vendor Managed Inventory – VMI

Vendor Managed Inventory (VMI) är ett partnerskap mellan en köpare och en leverantör (ofta en tillverkare) som innebär att köparen inte längre ansvarar för sitt eget lager. Istället är leverantören ansvarig för att styra och återfylla lagret, bestämma orderkvantiteter m.m.^{100,101} Leverantören kan ses som en central planerare som agerar för att maximera vinsten i det integrerade systemet. VMI är en mycket vanlig typ av strategisk allians och möjliggörs av att köparen kommunicerar signaler om efterfrågan till leverantören.¹⁰² Målet med VMI är att reducera det sammanlagda lagret och återfyllnadskostnaderna i hela försörjningskedjan. Detta gynnar båda parter, leverantör såväl som köpare.¹⁰³ De lägre lagernivåerna i distributionscenter och hos återförsäljare beror på att gods inte levereras ifall det inte behövs, medan leverans sker synbarligen automatiskt när det verkligen behövs.¹⁰⁴

Det traditionella sättet att göra affärer på är att alla parter i försörjningskedjan gör sina egna efterfrågeprognoser och bestämmer orderkvantiteter individuellt, vilket leder till ineffektivitet i försörjningskedjan.¹⁰⁵ Genom VMI uppnås bättre transparens i flödeskedjan och bättre kontroll över densamma. Den tidiga tillgången till information gör det möjligt för leverantörer att agera proaktivt och detta minskar ledtiderna. Lokal suboptimering förebyggs när aktörerna inte längre planerar individuellt. Tidig och kontinuerlig kommunikation mellan parterna minskar även den s.k. bullwhip-effekten,¹⁰⁶ vilken innebär att efterfrågevariationer förstärks genom försörjningskedjan. Eftersom prognosen i ett led bygger på prognosen i det tidigare ledet byggs fler och fler fel in för varje led,¹⁰⁷ vilket undviks i VMI-sammanhang eftersom information sprids genom leden.

2.6.1 VMI och informationssystem

VMI behöver informationsteknologi för att kunna fungera på ett effektivt sätt. EDI (electronic data interchange) är ett vanligt och viktigt verktyg. Till exempel kan leverantören få daglig information om lagernivåer via EDI. Efter att en order sammanställts, men innan den levereras, kan leverantören få en elektronisk avisering och senare kan betalningen ske elektroniskt.^{108,109} Det finns även särskild mjukvara för VMI-

⁹⁹ Mattsson, *Standardavvikelser för säkerhetslagerberäkning*, s. 18-19.

¹⁰⁰ Emigh, J. Vendor-Managed Inventory. *Computerworld*, vol. 33, nr. 34, 23 augusti 1999, s. 52-53.

¹⁰¹ Wang et al., Supply chain model in vendor managed inventory.

¹⁰² Wu, J. och Zhai, X. The Value of Centralization in a Vendor-Managed Inventory System. *2007 International Conference on Management Science and Engineering*, 2007, s. 704-710.

¹⁰³ Wang et al., Supply chain model in vendor managed inventory.

¹⁰⁴ Emigh, Vendor-Managed Inventory.

¹⁰⁵ Wu och Zhai, The Value of Centralization in a Vendor-Managed Inventory System.

¹⁰⁶ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

¹⁰⁷ Shao, Ju-ping et al. Research review on bullwhip effect controlling methods in a supply chain under uncertainty environments. *2008 3rd IEEE Conference on Industrial Electronics and Applications*, 2008, s. 1803-1808.

¹⁰⁸ Emigh, Vendor-Managed Inventory.

implementationer, som kan integreras med större system för försörjningskedjan där lagerstyrning kombineras med verktyg för samarbete inom försäljningsplanering och prognostisering.^{110,111} Målsättningen med IT i ett VMI-sammanhang (och generellt inom supply chain management) är att göra information tillgänglig och synlig, möjliggöra en enskild kontaktpunkt för data, samla information om försörjningskedjan för att möjliggöra beslut baserade på hela försörjningskedjan, samt att möjliggöra samarbete mellan parterna i försörjningskedjan.¹¹²

2.6.2 Fördelar med VMI för köparen

Den största fördelen med VMI är att kostnader och lagernivåer kan reduceras genom hela flödeskedjan. För köparen är det en fördel att ägandeskapet av lagret flyttas över till leverantören, vilket innebär att råmaterial och produkter i arbete försvinner från balansräkningen, vilket i sin tur minskar behovet av rörelsekapital.¹¹³ Även administrations- och transportkostnader kan minskas med hjälp av VMI. Även om kostnadsreduktion vanligtvis anses vara en av de viktigaste fördelarna med VMI menar Claassen et al. att denna förbättring inte är lika betydelsefull som förbättrade servicenivåer och kontroll över försörjningskedjan. De förbättrade servicenivåerna kommer till uttryck genom färre akutbeställningar och minskat antal felaktiga order.¹¹⁴

Wang et al. drar slutsatsen att den optimala återfyllnadsfrekvensen är högre för en köpare som använder VMI än för en som inte gör det. Å andra sidan är de optimala bristkvantiteterna och de optimala orderkvantiteterna mindre med VMI än utan.¹¹⁵

2.6.3 Fördelar med VMI för leverantören

Inte bara köparen utan också leverantören bör få nytta av ett VMI-samarbete, eftersom ett äkta VMI-system är en situation som båda parter tjänar på. Man bör komma överens om servicenivåer tillsammans, men i övrigt bör leverantören fritt kunna planera produktion och transporter på ett optimalt sätt för att få en stabil produktion. Fördelar i kapacitetsplanering uppnås ifall leverantören implementerar VMI-samarbete med ett stort antal kunder. Leverantörens lagerkostnad kan sänkas genom att säkerhetslagren kan minskas när osäkerheten minskas. Den lägre osäkerheten emanerar från leverantörens viktigaste VMI-fördel, nämligen att denne, genom tidig information om efterfrågeprognoser, kan planera sina produktionsprocesser efter efterfrågan på ett bättre sätt. Även transportkostnader kan sänkas ifall VMI implementeras storskaligt, eftersom leverantören då får möjlighet till ett flexibelt återfyllnadsschema där godset kan transporteras i fyllda lastbärare. En annan mycket viktig fördel är att leverantören får lojala kunder och en tryggad försäljning, p.g.a. att VMI-samarbete skapar långvariga och pålitliga kundrelationer.¹¹⁶

¹⁰⁹ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

¹¹⁰ Emigh, Vendor-Managed Inventory.

¹¹¹ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

¹¹² Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

¹¹³ Emigh, Vendor-Managed Inventory.

¹¹⁴ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

¹¹⁵ Wang et al., Supply chain model in vendor managed inventory.

¹¹⁶ Claassen et al., Performance outcomes and success factors of vendor managed inventory (VMI).

3 Metod

Detta kapitel redogör för den metod som använts i studien. Efter en inledande beskrivning av ansats och undersökningsmetod följer en genomgång av datainsamling och den praktiska arbetsgången. Avslutningsvis förs en argumentation kring studiens giltighet.

En studies övergripande metodik kan delas in i beskrivande, utforskande, förklarande och problemlösande. Den problemlösande metodiken har till syfte att finna en lösning till ett separat identifierat problem¹¹⁷ och har bedömts vara det mest relevanta tillvägagångssättet för detta arbete. En problemlösande metodik stämmer väl överens med att en dynamisk parametersättning successivt utvecklas från en befintlig modell i syfte att lösa problemet huruvida en dynamisk parametersättning förbättrar slutresultatet.

3.1 Forskningsansats

För att få fram en modell som vilar på underbyggd, teoretisk grund har i detta arbete en s.k. hypotetisk-deduktiv ansats använts. Den hypotetisk-deduktiva metoden fäster större vikt vid teoribildning än den induktiva metoden, som grundar sig på en förutsättningslös datainsamling från vilken man försöker dra allmängiltiga slutsatser. Nackdelen med den induktiva metoden kan anses vara att annan teori än den som framkommer genom det empiriska materialet saknas. En hypotetisk-deduktiv ansats innebär istället att man ur befintlig teori formulerar en hypotes som ska prövas empiriskt. Hypotesen kan till exempel prövas genom ett experiment där olika faktorer kan varieras systematiskt och effekterna av förändringarna mäts.¹¹⁸

I denna studie har, utifrån teori kring dynamisk parametersättning, hypotesen att det specifika system som ska undersökas kan styras utifrån någon eller några av de framtagna metoderna för uppdatering av säkerhetstid formats. Efter justeringar av metoderna har hypotesen om styrning prövats genom simuleringar där olika faktorer varierats systematiskt. Detta innebär att en hypotetisk-deduktiv ansats använts. Den induktiva metoden kan inte anses vara ett alternativ, då den saknar såväl den teoretiska grunden som hypotesprövningsmomentet.

3.2 Undersökningsmetod

I forskning inom *supply chain management* (SCM) är det ofta ett problem att avgöra vilka eventuella förbättringar i prestanda som härrör sig från vilka förändringar. Kvantitativ empirisk forskning måste utformas så att validiteten hos de kvantitativa, teoretiska modeller som ligger till grund för forskningen bibehålls med hänsyn till den data man använder. För att åstadkomma detta lämpar sig ofta händelsedrivna simuleringsmodeller och simulering kan sägas vara en av de viktigare undersökande metoderna inom SCM.¹¹⁹ Den grundläggande styrkan i simulering som en metod inom SCM-forskning ligger i metodens förmåga att hantera variabilitet och ömsesidiga beroenden. Simulering erbjuder därför en

¹¹⁷ Höst, M., Regnell, B. och Runeson, P. *Att genomföra examensarbete*. Lund: Studentlitteratur, 2006, s. 29.

¹¹⁸ Wallén, G. *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur, 1993, s. 44-45.

¹¹⁹ Kotzab, H., Seuring, S., Müller, M. och Reiner, G. *Research Methodologies in Supply Chain Management*. Heidelberg: Physica-Verlag, 2005, s. 437.

möjlighet att visualisera vilka genomslag förändringar i systemkomponenter får för prestandan i hela försörjningskedjan.¹²⁰

I denna studie har simuleringsmodeller använts för att undersöka och validera hur en dynamisk parametersättning påverkar systemets prestanda. Användningen av en kvantitativ metod och speciellt simulering är nödvändig för att kunna behandla tillräckligt stora datamängder och konfigurationer i ljuset av att modellen bör ha generell giltighet. Ingen annan metod har bedömts vara möjlig för att uppfylla arbetets mål. Simulering som metod är också lämplig eftersom den erbjuder enkel replikering (upprepbarhet), en liten tidsåtgång i relation till vilka tidsrymder som kan behandlas och att ett direkt experimenterande är uteslutet på grund av de stora extra kostnader som är behäftade med en felaktig lagerstyrning.

3.3 Datainsamling

Den data som ligger till grund för undersökningarna är av kvantitativ natur eftersom den är konkret och kan bearbetas med statistisk analys.¹²¹ Dataunderlaget blir också stort och av kvantitativ art genom att t.ex. efterfrågedata kan genereras i obegränsad mängd utifrån statistiska fördelningar. I och med de avgränsningar som gjorts för studien har hänsyn inte tagits till verkliga företeelser och undantag, som skulle kunnat innebära behov av data av kvalitativ art. Således har enbart kvantitativ data förekommit i studien.

För att ta fram en modell för dynamisk optimering av kontrollparametrar som har en allmän giltighet för större kundgrupper krävs en större mängd representativ data. Historisk data finns tillgänglig för olika företag på PipeChains servrar. För att få statistiskt säkerställda resultat krävs mycket stora mängder data som i praktiken motsvarar många års historik, vilket inte finns tillgängligt i form av verklig data. Istället kan teoretiskt genererad data användas för att dra generella slutsatser med statistisk signifikans. Verklig data utgör ett komplement som kan verifiera resultaten från teoretisk data och ge exempel på hur styrningen fungerar i verkligheten.

3.3.1 Genererad data

Efterfrågedata med endast slumpmässiga variationer

Mattsson föreslår en typ av efterfrågedata att använda för att få så verklighetsnära värden som möjligt.¹²² Den föreslagna genereringsmetodiken har använts även i denna studie, genom att efterfrågedata skapats med hjälp av två oberoende fördelningar. Antalet kundorder har genererats enligt en Poissonfördelning med sex olika ankomstintensiteter (även betecknat λ). Vid varje enskild ankomst har sedan en order skapats vars storlek är ett rektangelfördelat tal mellan 1 och 10. Detta har således gett sex olika fall av efterfrågedata. Dessa efterfrågedata har sedan använts i kombination med olika parametersättningar i form av ledtid och maxtid (se tabell 2), vilket har gett 120 olika fall.

¹²⁰ Wyland, B., Buxton, K. och Fuqua, B. Simulating the supply chain. *IIE Solutions*, vol. 32, nr. 1, 2000, s. 37.

¹²¹ Höst et al., *Att genomföra examensarbete*, s. 30.

¹²² Mattsson, *Standardavvikelser för säkerhetslagerberäkningar*, s. 5-6.

Tabell 2 Parametrar och variabler för genererade artiklar

	Värde	Enhet
Ankomstintensitet	50, 10, 3, 1/2, 1/10 och 1/40	Ankomster/dag
Orderkvantitet	$R(1, 10)$	Styck
Ledtid	2, 5, 10, 20 och 40	Dagar
Maxtid	5, 10, 20 och 60	Dagar

Med utgångspunkt i arbetsdagar har förenklingen gjorts att en vecka har 5 dagar, en månad 20 dagar och ett år 240 dagar. Efterfrågedata för 6000 dagar, d.v.s. 25 år, har tagits fram i samtliga 120 fall, även detta i enlighet med Mattssons tidigare nämnda studie.

Systematiska variationer

För att verkligen undersöka nyttan av en dynamisk parametersättning har även efterfrågan med systematiska variationer tagits fram. Detta eftersom det är speciellt under förändrade förutsättningar som en förändrad parametersättning över tid har relevans. Efterfrågedata har genererats på samma sätt som ovan, d.v.s. med Poissonfördelade kundankomster och orderkvantiteter rektangelfördelade mellan 1 och 10 som bas. Ovanpå detta har olika systematiska variationer lagts, vilka beskrivs nedan. Variationerna bygger på generella mönster hos verkliga artiklar, som observerats av anställda på PipeChain. Efterfrågestrukturerna har kombinerats med samtliga ledtider och maxtider i tabell 2, vilket har gett totalt 60 olika fall.

1. **Lång säsong:** Efterfrågan ökar långsamt under en period, ligger sedan länge på en hög nivå, sjunker långsamt under en period och ligger sedan länge på en låg nivå.

Ankomstintensiteten följer under hela simuleringstiden följande mönster som upprepas årsvis, d är dagsnumret årsvis:

Tabell 3 Variabler för efterfrågestrukturen "lång säsong"

Dag	Lambda
0-39	Linjär ökning från 10 till 30 $\lambda_d = 10 + 0,5d$
40-119	30
120-159	Linjär minskning från 30 till 10 $\lambda_d = 90 - 0,5d$
160-239	10

2. **Kort säsong:** Efterfrågan ligger under en stor del av året på en låg nivå, men ligger på en hög nivå under en kort period. Ankomstintensiteten följer under hela simuleringstiden följande mönster som upprepas årsvis, d är dagsnumret årsvis:

Tabell 4 Variabler för efterfrågestrukturen "kort säsong"

Dag	Lambda
0-69	10
70-89	Linjär ökning från 10 till 40 $\lambda_d = 1,5d - 95$
90-149	40
150-169	Linjär minskning från 40 till 10 $\lambda_d = 265 - 1,5d$
170-239	10

3. **Veckovariation:** Efterfrågan går upp i slutet av veckan och ligger resten av veckan på en jämn, låg nivå.

Ankomstintensiteten följer under hela simuleringstiden följande mönster som upprepas veckovis:

Tabell 5 Variabler för efterfrågestrukturen "veckovariation"

Dag	Lambda
Måndag-onsdag	10
Torsdag	45
Fredag	60

3.3.2 Verklig data

Den verkliga data från PipeChains server som använts består av data för 35 olika artiklar från ett företag med ett välfungerande logistikflöde och en relativt hög kunskapsnivå vad gäller parametersättningen. För varje artikel finns dagsvisa värden för ledtid, säkerhetstid, maxtid och s.k. verkligt utflöde. Det verkliga utflödet har använts som efterfrågedata i modellen vilket innebär en viss förenkling av verkligheten. Det kan nämligen ha funnits efterfrågan under tidsperioder när lagret är tomt som inte avspeglas i det verkliga utflödet eftersom ingenting kan gå ut från lagret när lagernivån är noll, vilket också kan få till följd att det efter en sådan period uppstår toppar i utflödet då restorder tillfredsställs. När felräkningar och felregistreringar uppstår i en verklig situation kan korrigeringar krävas som ger negativa värden på verkligt utflöde. Då dessa negativa siffror inte bör tolkas som efterfrågedata har de ersatts med nollor. Övriga parametrar kan användas direkt av modellen i obehandlad form.

3.3.3 Extremvärden

Oavsett vilken typ av indata som används måste extremvärden rensas bort. Även i verkligheten kan order uppstå som är oproportionerligt stora jämfört med nästan alla andra order. Dessa förutsätts dock få någon form av specialhantering, då ett företag ej kan ha säkerhetslager för att täcka upp denna typ av order. Extremvärden har därför rensats bort från all indata, verklig såväl som teoretiskt genererad, eftersom simuleringarna bara ska hantera order från det vanliga lagret och inte ta hänsyn till specialförfaranden.

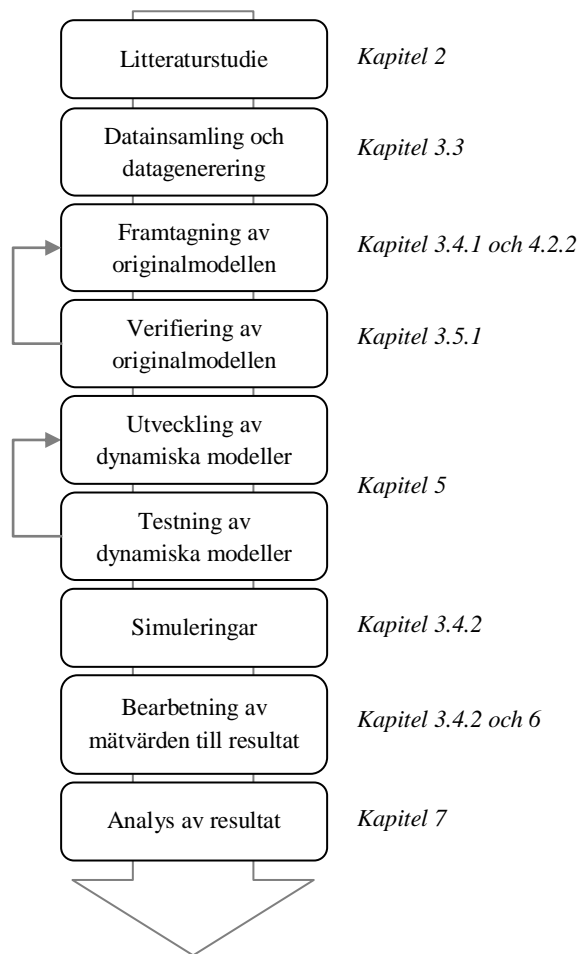
En vanlig metod är att värden som ligger över medelvärdet plus ett visst antal standardavvikelser rensas bort. Eftersom PipeChains flödesmätare rensar bort värden som ligger mer än tre standardavvikelser över medelvärdet, och eftersom tre är en vanlig faktor även i teorin, har extremvärdesrensningen i denna studie implementerats genom att värden som ligger mer än tre standardavvikelser över medelvärdet ersatts med medelvärdet.

3.4 Arbetsgång

Den övergripande arbetsgången i denna studie beskrivs i figur 1, där även hänvisning till var de olika momenten behandlas finns. Inledningsvis genomfördes en litteraturstudie för att få en god överblick över befintliga metoder för dynamisk beräkning av säkerhetslager. Därefter togs en simuleringsmodell (hädanefter kallad originalmodellen) fram som replikerar beteendet hos ett lagersystem med tidsfasad beställningspunkt och som är en förenklad bild av hur lagerstyrningen sker i PipeChain idag, så som det beskrivs i avsnitt 4.2.2 Grundläggande beordringsstrategi. Den framtagna modellen verifierades mot genererad data av olika slag. I nästa steg utvecklades, med utgångspunkt i litteraturstudien, fyra modeller för dynamisk uppdatering av säkerhetstid. Dessa modeller kan var och en för sig användas som ett komplement till originalmodellen. Simuleringar av samtliga modeller utfördes därefter med såväl genererad som verklig data. Slutligen genomfördes en analys av resultaten där servicenivå och medellagernivå jämfördes mellan originalmodellen och de dynamiska modellerna.

Verktyg

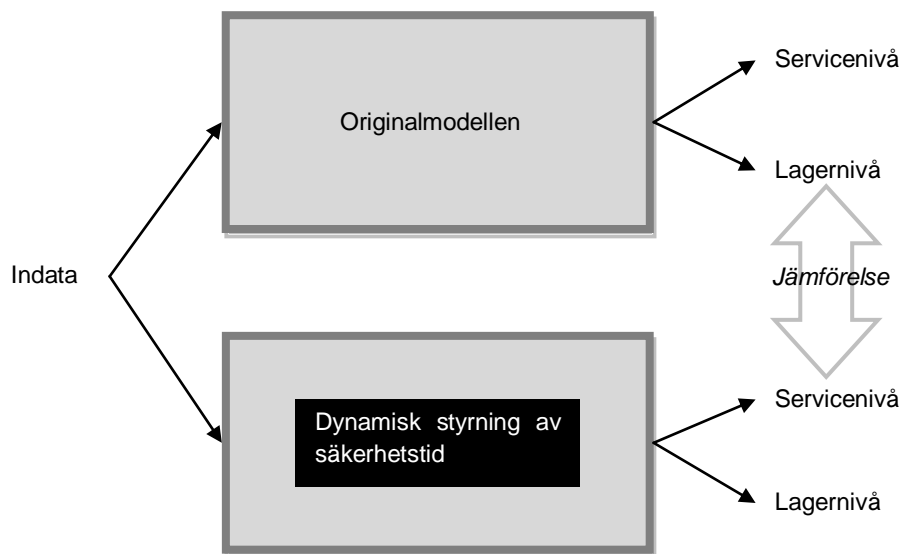
Samtliga modeller, originalmodellen såväl som de dynamiska, har skapats med en objektorienterad design i programmeringsspråket Java i Eclipse. Denna programmeringsmiljö har möjliggjort direkt integration med använda databaser genom SQL. Även generering av efterfrågan har skett genom körning av egenskrivna Java-program. Simuleringsresultat har skrivits ut med egendefinierade format och har sedan visualiserats i Excel där ytterligare beräkningar och statistiska jämförelser har utförts.



Figur 1 Arbetsgångens moment samt i vilka kapitel de behandlas.

3.4.1 Övergripande undersökningsmodell

Den övergripande modellen för undersökning beskrivs i figur 2. Indata till de båda modelltyperna är densamma medan utdata skiljer sig och är föremål för jämförelse.



Figur 2 Övergripande undersökningsmodell.

Indata

Indata består av efterfrågan som genererats utifrån statistiska fördelningar eller hämtats från en databas. Ingående parametrar utgörs av efterfrågan, säkerhetstid, maxtid och ledtid.

Utdata

Utdata utgörs av medelvärde och standardavvikelse för servicenivå ($SERV_2$), servicenivåns avvikelse från ett riktvärde samt medelvärde för lagernivån.

Styrning

Målet är att styra mot en viss servicenivå ($SERV_2$), vilket i första hand innebär att man vill komma så nära sitt mål som möjligt. I jämförelsen mellan modellerna är en högre servicenivå dock alltid bättre än en lägre så länge ökningen inte innebär en högre lagernivå. En högre servicenivå ställs alltså i förhållande till en eventuell ökning av lagernivån.

Antaganden

Modellen av det befintliga systemet är just en modell, d.v.s. en förenklad bild av verkligheten som visar alla de viktigaste aspekterna men förenklar bort en lång rad praktiska förhållanden och undantag som skulle göra modellen orimlig att hantera. Följande antaganden och förutsättningar gäller:

- Tidsupplösningen på simuleringen är 1 dag.
- För respektive dag antas inleverans ske före tillfredsställande av efterfrågan. Restorder tillfredsställs före dagens nya efterfrågan.
- Säkerhetstiden begränsas till minimalt noll och maximalt maxtiden.
- Hänsyn tas ej till att transporter och godsmottagning kanske bara kan utföras på vissa fördefinierade dagar.

- Helgdagar antas ej existera i simuleringen (efter fredag kommer måndag). Detta innebär också att inga ledtider kan löpa över helgdagar och ingen efterfrågan kan uppstå. Därmed har en vecka 5 dagar, en månad 20 dagar och ett år 240 dagar.
- Efterfrågedata antas ej vara autokorrelerad.

3.4.2 Undersökningsupplägg

Originalmodellen och tre av de dynamiska modellerna (*traditionell algoritm*, *traditionell algoritm med justeringar* och *simulering*, se kapitel 5) har testats med hjälp av simuleringar, med både genererad och verklig efterfrågedata. För de genererade efterfrågestrukturerna har 30 artiklar med 6000 dagsvärden var simulerats. Simuleringar har utförts för en målservicenivå på 95 respektive 98 procent. För verklig efterfrågan är målservicenivån 98 procent, som är den nivå som använts i praktiken. Den verkliga efterfrågan består av 35 artiklar som simulerats över de 1434 dagar av data som fanns tillgängliga. Resultaten från dessa simuleringar finns i avsnitt 6.1 till 6.3.

Prognosen som använts i samtliga dessa simuleringar är exponentiell utjämning, eftersom detta är den prognosmetod som finns i PipeChain. Kunderna väljer i de flesta fall att utnyttja sina egna prognosverktyg eller produktionsplaner. Detta är inte möjligt att återskapa i simuleringsmodellen och således har exponentiell utjämning använts.

Ett kompletterande test har utförts för att se hur utfallet av originalmodellen förändras då en bättre prognos används när efterfrågan innehåller systematiska variationer, eftersom det är främst i detta fall en bättre prognos behövs. I detta test har således endast genererad efterfrågedata med systematiska variationer använts. Per varje efterfrågestruktur har också i detta test 30 artiklar med 6000 dagsvärden använts. Även den fjärde dynamiska modellen (som inte ingick i huvudtesterna beskrivna ovan) *tidsbaserad avvikelse* (se kapitel 5) har testats under dessa förutsättningar eftersom denna modell kommer till sin fulla rätt endast med en bättre prognos. Dessutom har en av de tre andra dynamiska modellerna (*traditionell algoritm med justeringar*) tagits med i detta test i referenssyfte. Den bättre prognosen har genererats på samma sätt som själva efterfrågan har genererats, vilket innebär att den bygger på samma systematiska variationer men med andra slumpmässiga avvikelser. Resultaten från detta kompletterande test finns i avsnitt 6.2.1.

Ytterligare ett kompletterande test har gjorts med de tre förstnämnda dynamiska modellerna för att undersöka vilken inverkan det skulle få på utfallet att bortse från antagandet att säkerhetstiden ska vara noll eller positiv, d.v.s. ett test där negativa säkerhetstider tillåts. I detta test har endast genererad efterfrågan utan systematiska variationer använts och endast enstaka artiklar simulerade över 6000 dagar har jämförts. Resultaten återfinns i avsnitt 6.4.

Referensmodeller

För att få en jämförelse mellan de dynamiska modellerna och originalmodellen vid genererad efterfrågan har två referensfall med statiska säkerhetstider, som är samma genom hela simuleringen, tagits fram. I *original(optimal)* används en optimal statisk säkerhetstid som tagits fram genom simulering av samtliga 6000 dagsvärden. Detta är en teoretiskt optimal tid som är helt och hållet en efterhandskonstruktion och som inte är möjlig att få

fram i praktiken, eftersom man inte har tillgång till 25 års representativ historisk data. I *original(statisk)* används en statistiskt satt säkerhetstid som är 1,5 gånger så stor som ledtiden, men aldrig större än halva maxtiden, d.v.s. beräknad enligt $ST = \min(1,5LT; 0,5MT)$. Denna statistiska beräkning av säkerhetstiden är en grov approximation av verkliga parametrar, som tagits fram genom att studera förhållanden hos verklig data. För verklig efterfrågan har som referens istället en originalmodell som använder verkliga, registrerade säkerhets-, max- och ledtider använts.

För att identifiera den undre respektive övre gränsen för vilken servicenivå som kan uppnås genom att enbart variera säkerhetstiden används ytterligare två referensfall. Dessa är *original(0)*, där säkerhetstiden är konstant noll genom hela simuleringen, samt *original(maxtid)* med säkerhetstiden lika med maxtiden under hela simuleringen. Dessa gränser beror på att säkerhetstiden är begränsad till just 0 och maxtiden i dagens PipeChain-system (se avsnitt 4.2.2 Grundläggande beordringsstrategi).

Framtagning och analys av mätvärden

De mätvärden som tagits fram är servicenivåns avvikelse från riktvärdet, servicenivåns medelvärde, servicenivåns standardavvikelse och totalt medellager. Dessa mätvärden har beräknats steg för steg som medelvärden, såsom beskrivs nedan. Dessutom sätts i slutskedet servicenivå och medellager i relation till varandra genom mätvärdet antal enheter i lager per procentenhet av den uppnådda servicenivån.

Det s.k. riktvärdet för servicenivån är normalt sett lika med målservicenivån. Om det förhåller sig så att *original(0)* presterar över målservicenivån blir denna servicenivå istället riktvärdet eftersom detta är den lägsta servicenivå som i så fall kan nås (så länge negativa säkerhetstider inte tillåts). Om istället *original(maxtid)* presterar under målservicenivån blir detta riktvärdet eftersom detta då är den högsta servicenivå som är möjlig att nå.

För varje simulering av en grupp efterfrågestrukturer (med grupp menas här genererad efterfrågan av högrörliga respektive lågrörliga artiklar med enbart slumpmässiga variationer, genererad efterfrågan med systematiska variationer respektive verklig efterfrågan) med olika ledtider och maxtider har mätvärdena för respektive enskilt fall (ett fall motsvaras av en efterfrågestruktur kombinerad med en viss ledtid och maxtid) registrerats, se bilaga 1-3.

I nästa skede har de enskilda värdena från respektive fall summerats för hela gruppen av efterfrågestrukturer, så att mätvärdet beskriver prestationen för samtliga efterfrågestrukturer och tidsparametrar inom gruppen. I fallet med verklig efterfrågan är det denna summa som är det slutliga mätvärdet.

I fallen med genererade efterfrågestrukturer har simuleringarna av varje grupp (genererad efterfrågan av högrörliga respektive lågrörliga artiklar med enbart slumpmässiga variationer samt genererad efterfrågan med systematiska variationer) körts 30 gånger, vilket kan ses som att det för varje efterfrågestruktur finns 30 artiklar. Dessa simuleringar har således genererat 30 värden för var och en av de undersökta modellerna. Det sista steget som

utförts för de genererade efterfrågestrukturerna är att ett medelvärde av måtten beräknats från de 30 simuleringarna. På dessa medelvärden har signifikanstest gjorts för att undersöka ifall de olika modellernas prestation skiljer sig ifrån varandra.

Måttet antal enheter i lager per procentenhet av den uppnådda servicenivån har i slutskedet räknats ut som ett medelvärde av samtliga fall och simuleringar inom gruppen. De olika modellernas förmåga att styra in mot en viss servicenivå sett dels till respektive efterfrågestrukturer och dels till de olika ledtidfallen har också undersökts.

3.5 Giltighet

3.5.1 Validitet

En god validitet innebär att man faktiskt mäter det man hade för avsikt att mäta¹²³, vilket också kan uttryckas som att en modell med god validitet saknar systematiska fel. I denna studie har validiteten säkerställts genom att de grundläggande parametrarna bygger på vedertagna modeller som har klara definitioner på relevanta variabler och deras samband, d.v.s. en god teoretisk validitet.¹²⁴ Den s.k. empiriska validiteten, dvs. modellens förmåga att återge ett experimentellt utfall¹²⁵, har säkerställts genom att utvecklandet av modellen har skett i en iterativ process där utdata verifierats visuellt steg för steg i Excel och sedan jämförts med den teoretiska bakgrunden. Först utfördes manuella beräkningar som därefter visualiserades med hjälp av Excel. När dessa två led gav samma resultat implementerades motsvarande logik i den slutliga simuleringsmodellen, som verifierades mot Excel-beräkningarna.

3.5.2 Reliabilitet

Reliabilitet, dvs. tillförlitligheten hos mått, mätningar¹²⁶, datainsamling och analys när det gäller slumpmässiga variationer¹²⁷, är en förutsättning för validitet.¹²⁸ I denna studie har teoretisk indata använts som genererats helt utifrån statistisk teori och håller nödvändig reliabilitet. Den verkliga indata som använts är historisk data sparad i en databas. Denna har använts utan några omfattande justeringar och måste därför också anses uppfylla en god reliabilitet. Den enda justering av verklig indata som gjorts är av det s.k. verkliga utflödet, se avsnitt 3.3.2 Verklig data.

För att avgöra robusthet och signifikans hos resultaten i en kvantitativ studie som denna är statistisk analys avgörande.¹²⁹ För vart och ett av de undersökta fallen har därför måtten servicenivåns avvikelse från riktvärdet, servicenivåns medelvärde och standardavvikelse samt totalt medellager hypotesprövats för att avgöra om de uppmätta skillnaderna mellan de olika modellerna är statistiskt signifikanta. Den valda signifikansnivån är i första hand 1 procent och i andra hand 5 procent. En signifikansnivå på 1 procent innebär i detta

¹²³ Höst et al., *Att genomföra examensarbete*, s. 41.

¹²⁴ Wallén, *Vetenskapsteori och forskningsmetodik*, s. 61.

¹²⁵ Wallén, *Vetenskapsteori och forskningsmetodik*, s. 61.

¹²⁶ Bryman, A. *Samhällsvetenskapliga metoder*. Malmö: Liber ekonomi, 2002, s.86.

¹²⁷ Höst et al., *Att genomföra examensarbete*, s. 41.

¹²⁸ Bryman, *Samhällsvetenskapliga metoder*, s. 90.

¹²⁹ Patel, R. och Davidson, B. *Forskningsmetodikens grunder*. Lund: Studentlitteratur, 2003, s. 14.

sammanhang att risken är 1 procent att två värden som påstås vara signifikant olika i själva verket är lika. Eftersom värdena är oberoende och 30 observationer erhållits för varje mått har hypotesprövning med så kallat Z-test med följande testfunktion¹³⁰ utförts:

$$Z = \frac{(\bar{X}_1 - \bar{X}_2) - d_0}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (3.1)$$

där n_1 och n_2 är stickprovsstorlekarna
 \bar{X}_1 och \bar{X}_2 stickprovens medelvärden
 s_1 och s_2 standardavvikelseerna i de båda stickproven och
 d_0 differensen mellan populationsmedelvärdena enligt nollhypotesen

3.5.3 Generaliserbarhet

Syftet med att använda teoretiskt genererad efterfrågedata är att få fram resultat som är generella och allmängiltiga för många verkliga situationer, även om det i praktiken alltid kommer att finnas artiklar som inte överensstämmer med generella mönster. De genererade efterfrågestrukturer som använts i denna studie är så pass generella att de kan anses täcka in många verkliga situationer, eftersom datamängderna är stora och baserade på verklighets-trogna statistiska fördelningar med stor bredd på ingående parametrar. Dessutom har de undersökts över en lång simuleringstid och med två olika målservicenivåer. En begränsning i generaliserbarheten för artiklar med låga ankomstintensiteter kan dock anses föreligga eftersom de enskilda orderkvantiteterna aldrig blir högre än tio. Modellerna har också verifierats mot verklig historisk data som ger en fingervisning om användbarheten i praktiska fall.

3.5.4 Källkritik

Den teori som ligger till grund för modelldesignen är hämtad från väletablerad litteratur inom lagerstyrning samt från källor som genomgått en vetenskaplig granskningsprocess. Teoribildningen och källornas trovärdighet får här också empirisk prövning och en möjlighet till konkret jämförelse av prestanda genom utförandet av praktiska simuleringar med grund i de beskrivna modellerna.

¹³⁰ Körner, S. och Wahlgren, L. *Statistisk dataanalys*. Lund: Studentlitteratur, 2000, s.208

4 Programvaran PipeChain

De grundläggande funktionerna i och användningsområdena för mjukvarulösningen PipeChain beskrivs i detta kapitel. Informationen som presenteras har inhämtats under löpande samtal med anställda på företaget samt i företagsinterna dokument.

Mjukvaran med det självbetitlade namnet PipeChain är ett program vars yttersta syfte är att förbättra ett företags försörjningskedja ur ett lönsamhets- och konkurrensperspektiv. Programmet erbjuder flera gränssnitt för hantering på såväl ledningsnivå som operationell nivå. En av PipeChains viktigaste egenskaper är att det är ett av få program som faktiskt har som huvudsyfte att koppla ihop olika företag, en förutsättning för att arbeta ur ett VMI-perspektiv.

Den metod för lagerstyrning som används i programmet kan närmast beskrivas som tidsfasad beställningspunkt och sker i något som kallas *PipeChain Replenishment Engine*. Förutom att programmet självt kan räkna på lagerstyrning enligt denna metodik kan även vid önskemål beräkningar genomföras och planeras utanför programmet, typiskt i företagets eget ERP-system, för att sedan importeras till PipeChain. Programmet används i detta senare fall bara som en sammankoppling mellan företagen.

4.1 Typer av flöden för vilka PipeChain kan användas

PipeChain hanterar fyra olika typer av flöden. Vilken typ av flöde som är aktuellt beror på om det företag som installerar PipeChain (i fortsättningen benämnt klienten) är leverantör eller kund i den relation som PipeChain ska hantera samt om klienten har en VMI-relation med sina leverantörer respektive kunder. De fyra flödena benämns Purchase Order Collaboration (POC), Sales Order Collaboration (SOC), Vendor Managed Inventory (VMI) samt Customer Vendor Managed Inventory (C VMI) och definieras enligt tabell 6.

Tabell 6 Olika flöden som kan styras med PipeChain

Typ av flöde	Klientens roll	VMI-relation
POC	Kund	Nej
SOC	Leverantör	Nej
VMI	Kund	Ja
C VMI	Leverantör	Ja

4.1.1 Purchase Order Collaboration (POC)

POC innebär ett traditionellt tillvägagångssätt där köparen lägger inköpsorder till leverantören och begär leverans enligt sina behov. Genom att använda PipeChains POC-lösning kan klienten hantera transaktioner med alla sina leverantörer elektroniskt.

Inköpsrekvisitioner kan beräknas antingen i klientens ERP-system eller direkt i PipeChain. I det första fallet skickas informationen till PipeChain och publiceras i ett web-gränssnitt. I det andra fallet utförs beräkningarna i *Replenishment Engine*. Därefter publiceras inköpsrekvisitionerna i webbgränssnittet samt skickas till klientens ERP-system. Informationen som PipeChain tar emot från ERP-systemet kan, beroende på funktionerna i

det aktuella systemet, exempelvis vara i form av inköpsorder med återfyllnadskvantitet samt prognoser av efterfrågan per leverantör, eller t.ex. i form av leveransscheman. I det senare fallet delar PipeChain upp leveransschemat i inköpsorder samt prognos.

4.1.2 Sales Order Collaboration (SOC)

SOC innebär ett traditionellt tillvägagångssätt där köparen lägger inköpsorder till leverantören och begär leverans enligt sina behov. Genom att använda PipeChains SOC-lösning kan klienten hantera transaktioner med alla sina kunder elektroniskt. I SOC-lösningen placerar kunden antingen en inköpsorder som läggs manuellt i ett webbgränssnitt (webbshop), eller direkt i kundens eget ERP-system. Den lagda inköpsordern skickas automatiskt till PipeChain.

4.1.3 Vendor Managed Inventory (VMI) och Customer Vendor Managed Inventory (C VMI)

I en VMI-relation ligger ansvaret för återfyllnad på leverantören istället för att kunden skickar order. Detta möjliggörs genom att kunden delar med sig av information angående lagernivåer och verkligt behov i tid och kvantitet, vilket leder till att processen styrs av efterfrågan istället för av order.

Det är i normalfallet klienten (d.v.s. kunden i en VMI-relation och leverantören i en C VMI-relation), som har PipeChain installerat på sin server eller på en värdserver. Motparten har i normalfallet webbtillgång till klientens PipeChain, men kan också ha en egen installation av PipeChain. Kunden arbetar inte aktivt i PipeChain, men kan använda sig av programmet för att övervaka flödet.

4.2 PipeChains beräkningsmodell

PipeChains beräkningar av inköpsrekvisitioner grundar sig på ett tidskoncept, till skillnad från de flesta andra system som baserar beräkningarna på antal enheter. Tidskonceptet är mer dynamiskt eftersom lagernivån följer efterfrågemönstret. Detta anses vara en av fördelarna med PipeChains beräkningsmodell. En annan är att parametrarna är enkla att underhålla för en normal användare. Indata till PipeChains beräkningar är lager, prognoser i någon form (se stycke 4.4 Prognoser i PipeChain) samt de logistiska parametrar som satts upp. Resultatet av beräkningarna skickas sedan in i ERP-systemet.

4.2.1 Parametrar

I ett VMI-samarbete definieras ett antal parametrar i leveransavtalet (*the delivery agreement*) som sedan används för att utföra beräkningarna i PipeChains *Replenishment Engine*. Även i de fall av POC-lösningar där PipeChains *Replenishment Engine* används definieras dessa parametrar. Parametrarna är de olika tidsmåttens ledtid, säkerhetstid, maxtid och kritisk tid samt även minsta leveranskvantitet (*minimum delivery quantity*), multipelkvantitet (*multi unit quantity*) samt mottagningschema (*reception schedule*).

Tidsparametrar

- Ledtid (LT) = tiden från det att godset avsänds från leverantören tills det mottages av kunden.
- Säkerhetstid (ST) = den minsta täcktid för lagret som kunden begär för en viss produkt. Denna är begränsad till maximalt maxtiden.
- Maxtid (MT) = den maximala täcktid för lagret som kunden begär för en viss produkt.
- Kritisk tid = täcktid som är användardefinierad för att vara en varning när lagret är nära att ta slut, kan t.ex. motsvara den minsta tid som krävs för att med specialhantering kunna undvika brist.

Övriga parametrar

- Minsta leveranskvantitet (*minimum delivery quantity*): Föreslagen kvantitet kommer vara lika med eller större än denna.
- Multipelkvantitet (*multi unit quantity*): En framräknad kvantitet kommer alltid vara en multipel av denna kvantitet.
- Mottagningsschema (*reception schedule*): Definierar kundens mottagningsdagar och leverantörens avsändningsdagar.

4.2.2 Grundläggande beordringsstrategi

De grundläggande beräkningarna i PipeChains *Replenishment Engine* bygger på tidsfasad beställningspunkt (materialbehovsplanering utan nedbrytning). Denna beräkning går till på följande sätt:

1. Utgå från aktuell lagerposition
2. Räkna med hjälp av prognos ut tiden t då lagerpositionen = 0
3. Beräkna mottagningstid: $t - ST$
 - Backa säkerhetstiden från tiden t
4. Beräkna beställningstid: $t - ST - LT$
 - Backa ledtiden från mottagningstiden
5. Beräkna orderkvantitet = Efterfrågan under MT – lagerposition vid tiden $(t - ST)$
 - Lagerposition efter inleverans ska täcka prognostiserat behov under maxtiden

Utöver dessa beräkningar tar PipeChain hänsyn till praktiska omständigheter som exempelvis definieras med hjälp av parametrarna minsta leveranskvantitet, multipelkvantitet samt mottagningsschema.

Säkerhetstiden

Parametern säkerhetstid har alltid ett positivt värde, d.v.s. den lägsta möjliga säkerhetstiden är noll. En negativ säkerhetstid skulle i praktiken innebära att man planerar att ta emot en leverans först efter det att lagret prognostiserats vara slut, d.v.s. att man i steg 3 ovan istället för att backa en viss tid går framåt en viss tid. Det maximala värdet som säkerhetstiden kan sättas till är maxtiden. Detta beror på att även om en längre säkerhetstid används kommer

detta inte att få någon ytterligare påverkan på utfallet. Eftersom den maximala lagernivån normalt sett motsvarar maxtidens behov kan den i steg 2 uträknade tidpunkten ligga högst en maxtid framåt i tiden. Om säkerhetstiden är längre än maxtiden kommer man i steg 3 således hamna på en dag före innevarande dag, vilket innebär att beställningen i praktiken måste läggas på innevarande dag, alltså samma dag som om säkerhetstiden hade varit lika med maxtiden. Att sätta en viss säkerhetstid och maxtid innebär i förlängningen att beställningskvantiteter och tidpunkter för beställning hålls inom vissa ramar. Detta gör att fall där det är omöjligt att nå en viss servicenivå kan uppstå.

4.3 Lagerstatus och servicenivå i PipeChain

I PipeChain definierar olika färger det aktuella läget i lagret som åskådliggörs grafiskt i programmet med en lagerstatusindikator (*the duration meter*). De fyra färgerna definieras enligt följande:

- Röd
 - Lagersaldot är noll. Ifall en kritisk tid angivits: lagersaldot täcker mindre än den kritiska tiden.
 - Det finns för lite eller ingenting i lager.
- Gul
 - Lagersaldot täcker mindre än säkerhetstiden.
 - Det finns för lite i lager, även om det på kort sikt räcker. En beställning måste läggas, om det inte redan gjorts.
- Grön
 - Lagersaldot täcker en tid som ligger mellan säkerhetstiden och maxtiden.
 - Det finns lagom mycket i lager.
- Blå
 - Lagersaldot täcker mer än maxtiden.
 - Det finns för mycket i lager.

Servicenivån kan beräknas på några olika sätt i PipeChain, beroende på kundens önskemål. Måttet grundas alltid på andel dagar som lagret har en viss färg; den färg eller de färger som av kunden definierats som godtagbara. Det finns fem möjliga varianter på vilka färger som kan anses vara acceptabla:

1. Grön
2. Grön + blå
3. Grön + gul
4. Grön + blå + gul
5. Grön + blå + gul + röd då lagret ej är tomt (d.v.s. röd definieras med en kritisk tid)

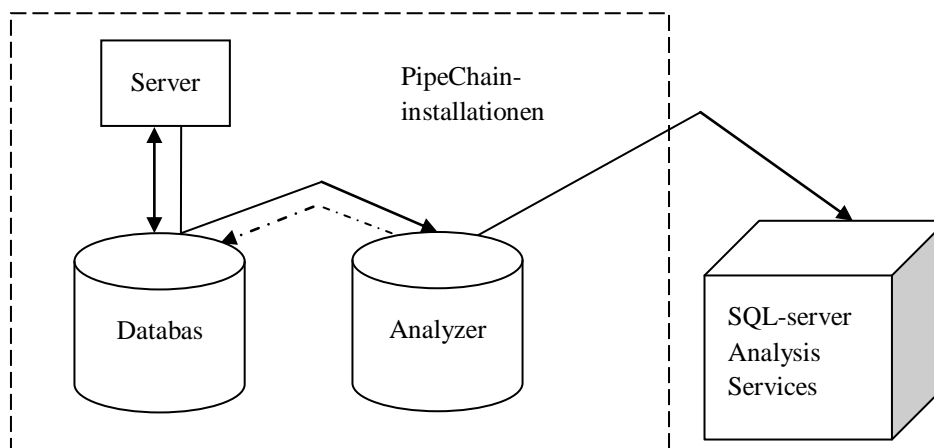
Den förvalda inställningen i PipeChain vid servicenivåberäkning är att andelen godtagbara dagar beräknas bland de senaste 28 dagarna. Antalet dagar kan dock varieras efter kundens önskemål upp till maximalt ett år.

4.4 Prognoser i PipeChain

Flödesmätaren är ett verktyg som alltid ingår i PipeChain. Den bygger på exponentiell utjämning av det historiska utflödet och ger bara ett värde per dag som utdata. I den exponentiella utjämningen tas extremvärden som ligger tre standardavvikelse över respektive under medelvärdet bort. Det är mycket få företag som faktiskt använder flödesmätaren, men när företaget inte har något prognosverktyg kan de använda detta verktyg. Företag som använder sin produktionsplan som indata i PipeChain låter dock ofta flödesmätaren ta vid där planen slutar, för att få en prognos för en längre tidshorisont. När användaren gör egna prognosberäkningar utanför PipeChain läggs dessa in i en flödesmodell. Indata till en flödesmodell kan, förutom prognoser, vara produktionsplan, en kombination av produktionsplan och prognos eller dylikt. I PipeChain finns möjligheten att ovanpå en flödesmodell eller flödesmätaren lägga en prognosprofil som motsvarar ett säsongsmönster i efterfrågan. Detta ger en mer detaljerad och bättre flödesmodell. Alla flödesmodeller sparas dagligen till databasen *PipeChain Analyser* där även prognostillförlitligheten beräknas.

4.5 Sambandet mellan de olika komponenterna i PipeChain-installationen

PipeChain-installationen innefattar PipeChain-servern som arbetar mot en databas samt *PipeChain Analyser* som är en databas. PipeChain-servern och databasen kommunicerar med varandra. Från databasen skickas sedan information till *Analyzern*. I vissa fall skickar den sistnämnda tillbaka information till databasen. Från *Analyzern* skickas sedan information till en SQL-server kallad *Analysis Services*. Här kan användaren sedan få tillgång till informationen på ett hanterbart sätt, via pivottabeller och lagringsstrukturer som integrerar fakta i olika dimensioner för att möjliggöra snabb avläsning, s.k. kuber. Sambandet mellan komponenterna åskådliggörs i figur 3.



Figur 3 Sambandet mellan komponenterna i PipeChain-installationen.

5 Dynamiska modeller för beräkning av säkerhetstid

I detta kapitel beskrivs alla de modeller för dynamisk uppdatering av säkerhetstiden som tagits fram under studien och därefter undersökts.

5.1 Förutsättningar för samtliga modeller

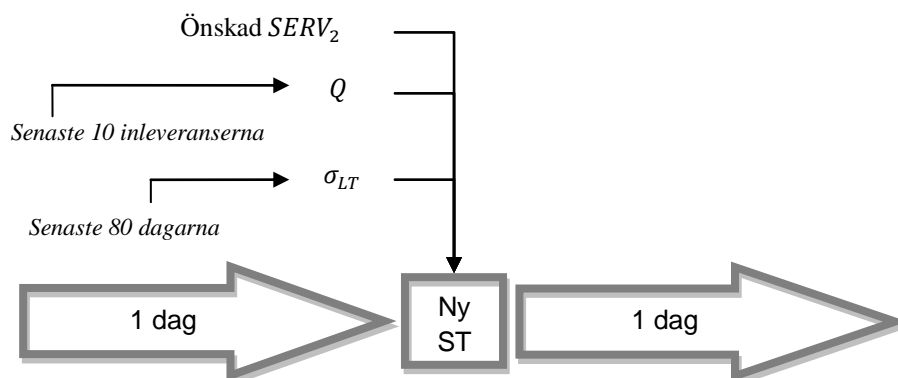
Samma antaganden som för originalmodellen gäller även för alla dynamiska modeller, se avsnitt 3.4.1 Övergripande undersökningsmodell. Således är utgångspunkten i samtliga metoder för uppdatering av säkerhetstiden att lagerstyrningen utförs med den variant av tidsfasad beställningspunkt som används i PipeChain. Alla servicenivåer uttrycks som $SERV_2$.

Lagerstyrningen i alla dynamiska modeller, liksom i originalmodellen, bygger på periodisk inspektion. De förändringar i formlerna som uppstår till följd av periodisk inspektion är dels i beräkningen av standardavvikelsen och dels i bestämningen av nästa inleveransdag (se avsnitt 2.2.3 Tidsfasad beställningspunkt). I de fall standardavvikelsen används i de dynamiska modellerna beräknas den över både ledtid och inspektionsperiod (d.v.s. ledtiden plus 1), istället för bara ledtiden. Då inleveransdagen ska beräknas ska hänsyn egentligen tas till att lagret inte bara ska täcka ledtiden och säkerhetstiden, utan också i medeltal hälften av inspektionsperioden, d.v.s. en halv dag extra. Eftersom systemet med tidsfasad beställningspunkt endast tillåter hantering av hela dagar måste denna extra halvdag avrundas antingen till noll eller en dag. Vidare påverkar denna förändring en helt annan del av lagerstyrningssystemet än beräkningen av säkerhetstid. Även i originalmodellen skulle denna hänsyn egentligen tas likaväl som i de dynamiska modellerna. Eftersom originalmodellen ej ska förändras, utöver hur säkerhetstiden sätts, utelämnas detta hänsynstagande även i de dynamiska modellerna. Beräkningar för att ta hänsyn till överdrag (se avsnitt 2.2.4 Överdrag) är desamma som de beräkningar som måste göras p.g.a. att inspektionen sker periodiskt med perioden en dag. Således tas ingen särskild hänsyn till överdrag.

5.2 Traditionell algoritm för beräkning av säkerhetslager

När simuleringarna körs inleds varje dag med att prognosen uppdateras. I denna modell uppdateras säkerhetstiden direkt därefter, varje dag. Detta motsvarar i praktiken att säkerhetstiden såväl som prognosen räknas om varje natt. Även andra uppdateringsintervall, d.v.s. att säkerhetstiden endast uppdateras med ett antal dagars mellanrum, har testats och refuserats då resultaten visade att vid efterfrågan med systematiska förändringar försämrades modellens förmåga att styra in mot önskad servicenivå i takt med att uppdateringsintervallet blir längre.

I denna modell har de traditionella beräkningarna av säkerhetslager utifrån en förutbestämd servicenivå använts. Beräkningarna utförs med hjälp av ekvationerna (2.16), (2.17) och (2.18) i avsnitt 2.3.2 Säkerhetslager baserat på $SERV_2$. De ingående variablerna är således orderkvantiteten Q , standardavvikelsen för prognosfelet under ledtiden σ_{LT} och den önskade servicenivån $SERV_2$. En principskiss av modellen kan ses i figur 4.



Figur 4 Principskiss för modellen traditionell algoritm.

Standardavvikelse

Beräkningarna inleds med att standardavvikelsen för prognosfelet under ledtiden tas fram. Standardavvikelsen beräknas på 80 värden av prognosfel per dag, som sedan ledtidjusteras enligt ekvation (2.24). Korrelation kan uppstå då exponentiell utjämning används som prognos, men förenklingen att konstanten c (i ekvation (2.22)) satts till 0,5 vid ledtidjusteringen har gjorts. Empiriska undersökningar av denna modell har visat att då enbart slumpmässiga variationer förekommer i efterfrågan uppstår ingen skillnad i resultatet om standardavvikelse beräknas på historisk efterfrågan respektive prognosfel. I de fall det finns systematiska variationer i efterfrågan är det dock mer rimligt att beräkna standardavvikelsen av prognosfelet än av historisk efterfrågan, eftersom efterfrågans standardavvikelse blir mycket hög i en period med starkt uppåtgående eller nedåtgående trend. Testsimuleringar visar också att utfallet blir bättre om prognosfel ligger till grund för standardavvikelsen. Följaktligen används prognosfelets standardavvikelse i denna modell.

Orderkvantitet

Nästa steg är att ta fram parametern Q . I ett vanligt beställningspunktssystem är Q en fast och förbestämd beställningskvantitet. När styrningen sker med tidsfasad beställningspunkt varierar istället beställningskvantiteten. För att få ett rättvisande Q att använda i beräkningarna beräknas den genomsnittliga kvantiteten från de tio senaste beställningarna. Genom att ta med så pass många beställningar i beräkningen erhålls en utjämnad kvantitet, som inte i så hög grad påverkas av tillfälliga avvikelser.

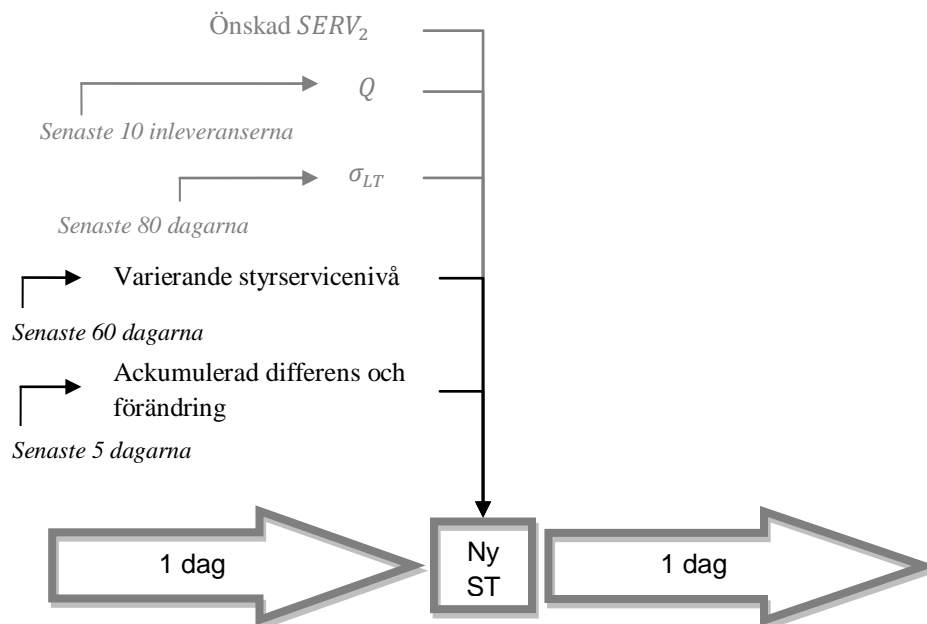
Från säkerhetslager till säkerhetstid

När säkerhetslagret beräknats enligt ekvation (2.16) beräknas säkerhetstiden som så många dagar som prognosen behöver ackumuleras för att uppnå det framräknade säkerhetslagret. Eftersom prognosen i de flesta fall inte kommer att gå jämnt ut med säkerhetslagret innebär detta att säkerhetstiden oftast kommer att motsvara en prognossumma som överstiger säkerhetslagret något. Således får man i praktiken ett något större säkerhetslager än önskat. Ett alternativ är att då den överskjutande delen av säkerhetslagerkvantiteten (d.v.s. det som

återstår efter att alla hela dagars prognos räknats bort från säkerhetslagerkvantiteten) överstiger en halv dags förbrukning avrundas säkerhetstiden uppåt, medan den avrundas nedåt då den överskjutande delen är mindre än en halv dag. Detta alternativ har refuserats eftersom det innebär att i ungefär hälften av fallen erhålls en säkerhetstid som motsvarar en mindre kvantitet än vad som egentligen räknats fram. Då huvudmålet måste anses vara att undvika brist har valet gjorts att inte göra denna avrundning.

5.3 Traditionell algoritm med justeringar

Denna modell bygger på samma traditionella beräkningar som den föregående. För att förbättra modellens prestation krävs någon form av justering utöver den grundläggande algoritmen. I den här modellen introduceras två övergripande sätt att utföra sådana injusteringar av säkerhetstiden, dels att styra mot en varierande servicenivå och dels att göra direkta ingripanden i den framräknade säkerhetstiden baserat på den senaste veckans prestation. Dessa justeringar har tagits fram genom empirisk testning. Testsimuleringar visar att de båda justeringarna var för sig förbättrar den grundläggande algoritmens instyrning mot önskad servicenivå, men störst förbättring uppnås då båda justeringarna används tillsammans. De två metoderna kan ses som komplement till varandra då den första justeringsmetoden, att styra mot varierande servicenivå, tar hänsyn till förändringar i prestationen över lite längre tid medan den andra justeringsmetoden går in och gör en injustering baserat på den senaste veckan ifall den första justeringsmetoden inte har reagerat tillräckligt snabbt på förändringarna. En principskiss av modellen, där de delar som är identiska med modellen *traditionell algoritm* är gråa, visas i figur 5.



Figur 5 Principskiss för modellen traditionell algoritm med justeringar.

Justeringarna, i synnerhet den andra som innebär att göra direkta ingripanden i den framräknade säkerhetstiden, kan ställas in på en mängd olika sätt. I detta skede har valet gjorts att ta fram en metod som passar både jämn och varierande efterfrågan så bra som möjligt, för att möjliggöra användning av samma metod för olika artiklar samt användning då man inte med säkerhet vet hur efterfrågan kommer att utvecklas. Inställningarna har tagits fram genom empirisk testning. De skulle kunna undersökas i mycket större utsträckning och därigenom optimeras. De presenterade inställningarna kan dock visa på hur principen fungerar, för att i ett senare skede optimeras. Då man senare eventuellt optimerar inställningarna kan man välja att göra det generellt, eller specifikt för olika typer av efterfrågan.

5.3.1 Styrning mot varierande servicenivå istället för önskad servicenivå

Denna justeringsmetod innebär att den servicenivå som används som inparameter i ekvation (2.16) varierar beroende på den tidigare uppnådda servicenivån, istället för att den förbestämda målservicenivån används.

Först tas den senast beräknade servicenivån fram, vilken är beräknad över de senaste 60 dagarna. Denna servicenivå jämnas ut med hjälp av exponentiell utjämning med utjämningskonstant 0,2 för att undvika att tillfälliga extremvärden i erhållen servicenivå får för stor inverkan på styrningen. Denna utjämning har genom testsimuleringar visats ge förbättrade resultat för efterfrågefall med systematiska variationer. För jämna efterfrågefall kan ingen skillnad utläsas mellan att använda och inte använda utjämningen. Den exponentiella utjämningen tas, på grundval av dessa observationer, med i modellen. Därefter beräknas differensen mellan den utjämnade erhållna servicenivån och målservicenivån. Om erhållen servicenivå har överträffat målservicenivån blir den varierande styrservicenivån nu målservicenivån minus differensen, d.v.s. har servicenivån överträffats med en procentenhet blir den nya styrservicenivån nu en procentenhet lägre än målservicenivån. Om istället erhållen servicenivå är lägre än målservicenivån sätts styrservicenivån till målservicenivån plus differensen, d.v.s. en procentenhets underprestation leder till att styrservicenivån ligger en procentenhet över målservicenivån. En begränsning nedåt sätts så att styrservicenivån aldrig kan bli lägre än två procentenheter under målservicenivån. Denna begränsning har tagits fram genom testkörningar, men vad som är den optimala begränsningen har inte utretts.

5.3.2 Justering av säkerhetstiden grundat på senaste veckans prestation

Efter att en ny säkerhetstid beräknats grundat på den traditionella algoritmen med en varierande styrservicenivå, görs en kontroll av läget för servicenivån den senaste veckan. Dels beräknas den ackumulerade differensen mellan dag för dag erhållen och önskad servicenivå (ack diff), och dels beräknas ett ackumulerat värde av den förändring av servicenivån som varje dag skett jämfört med föregående dag (ack för). Båda dessa mätvärden beräknas som procentenheter. Beräkningarna visas i ekvationerna (5.1) och (5.2).

$$\text{ack diff} = \sum_{d=\text{idag}-5}^{\text{idag}} (S_d - S_{\text{mål}}) \quad (5.1)$$

$$\text{ack för} = \sum_{d=\text{idag}-5}^{\text{idag}} (S_d - S_{d-1}) \quad (5.2)$$

där S_d är erhållen servicenivå dag d och $S_{\text{mål}}$ är önskad servicenivå.

Beroende på storleken av den ackumulerade skillnaden respektive den ackumulerade förändringen görs eventuellt en korrigering av den nya säkerhetstiden jämfört med den tidigare säkerhetstiden (se avsnittet Korrigeringar nedan) samt en bestämning av utjämningskonstanten α . Den nya korrigerade säkerhetstiden, α samt den senast använda säkerhetstiden används därefter till att med exponentiell utjämning beräkna ett värde på säkerhetstiden. Exponentiell utjämning används här för att undvika för stora variationer i säkerhetstiden. Det är denna säkerhetstid som slutligen avrundas till heltal och används som den nya säkerhetstiden.

Korrigeringar

Vid korrigeringen av säkerhetstiden finns fyra olika korrigeringsnivåer: *ökning*, *undvik minskning*, *oförändrat* samt *minskning*. I tabell 7 redovisas gränserna i ackumulerad differens och ackumulerad förändring för de olika korrigeringarna. Dessa gränser har tagits fram genom manuella studier av det historiska utfallet under några simuleringskörningar. Vad som egentligen är de optimala gränserna är således inte utrett.

Tabell 7 Gränser (i procentenheter) för korrigering av säkerhetstiden.

	ack diff < 0	0 < ack diff < 0,5	0,5 < ack diff
ack för < -1	Ökning	Ökning	Ökning
-1 < ack för < -0,5	Ökning	Undvik minskning	Undvik minskning
-0,5 < ack för < 0	Undvik minskning	Oförändrat	Oförändrat
0 < ack för < 0,5	Undvik minskning	Oförändrat	Minskning
0,5 < ack för < 1	Oförändrat	Minskning	Minskning
1 < ack för	Minskning	Minskning	Minskning

Om den senaste veckans prestation ligger inom gränsvärdena för en viss korrigeringsnivå kommer följande åtgärder ske i respektive nivå:

- *Ökning*: Säkerhetstiden höjs upp till maxnivån, d.v.s. maxtiden, ifall algoritmen försöker sänka eller behålla gårdagens säkerhetstid.
- *Undvik minskning*: Gårdagens säkerhetstid behålls ifall algoritmen försöker sänka säkerhetstiden.

- *Oförändrat*: Säkerhetstiden som räknats fram med algoritmen behålls utan korrigering.
- *Minskning*: Gårdagens säkerhetstid behålls ifall algoritmen försöker höja säkerhetstiden. Dessutom kontrolleras om säkerhetstiden varit 1 under den senaste veckan (de fem senaste dagarna). I så fall sätts säkerhetstiden till 0, eftersom säkerhetstiden 1 har visats leda till överprestation.

Utjämningskonstanten α sätts i fallen *ökning* och *minskning* till 0,4 och i övriga fall till 0,1. Den högre konstanten för *ökning* och *minskning* motiveras av att man i dessa fall vill tvinga fram en större korrigering, medan man i de övriga fallen inte behöver göra en lika stor förändring av säkerhetstiden. Även dessa värden har tagits fram i testkörningar och ger godtagbara resultat, men de optimala nivåerna på utjämningskonstanten har inte utretts.

De inställningar som valts empiriskt i denna justeringsmetod ger stark förbättring av resultaten för efterfrågefall med endast slumpmässiga variationer. För efterfrågefall med systematiska variationer ger inställningarna också stor förbättring mot att inte ha någon justering, men det finns andra inställningar som ger lite bättre resultat. Dock är denna skillnad så liten att dessa inställningar kan bortses ifrån, i syfte att ta fram en generell metod som passar både jämn och varierande efterfrågan.

5.4 Traditionell algoritm med justeringar och tidsbaserad avvikelse

I denna modell utförs beräkningarna på samma sätt som i föregående modell, *traditionell algoritm med justeringar*, men σ_{LT} i beräknas inte som vanlig standardavvikelse, utan som en så kallad tidsbaserad avvikelse som presenterats av Krupp, se avsnitt 2.4.2 Tidsbaserad avvikelse vid säkerhetslagerberäkning. Beräkningen skiljer sig dock från Krupps så tillvida att det i denna modell är en tidsbaserad standardavvikelse (σ_{TB}) och inte ett tidsbaserat MAD-värde som används. Standardavvikelsen beräknas, liksom i de två föregående modellerna, på 80 dagars historiska prognosfel. Den tidsbaserade standardavvikelsen tas fram som i ekvation (5.3).

$$\sigma_{TB} = \sqrt{\sum_{i=\text{idag}-1}^{\text{idag}-1} \left(\frac{PF_i - \overline{PF}}{P_i} \right)^2 / \text{antal perioder}} \cdot P_{\text{idag}+1} \quad (5.3)$$

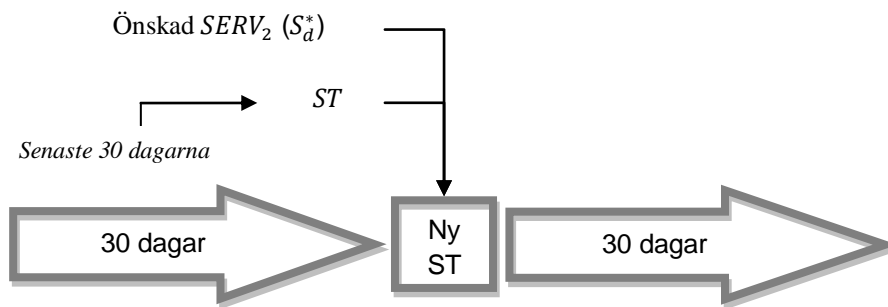
där PF_i är prognosfelet för dag i ,
 \overline{PF} medelvärdet av prognosfelen och
 P_i prognosen för dag i .

När prognosen bygger på exponentiell utjämning är alla framtida prognosvärden identiska och i dessa fall ger denna modell ett utfall som ligger mycket nära utfallet från modellen som använder en vanlig standardavvikelse. Det är endast då prognosen är mer avancerad

och består av unika värden för varje kommande dag som denna modells särart verkligen framkommer.

5.5 Simulering av lämplig säkerhetstid

Denna modell är helt fristående från övriga. Liksom i övriga modeller anges här en viss målservicenivå ($S_{mål}$) som man vill uppnå. I jämna optimeringsintervaller (OI), körs för den artikel man undersöker en simulering över historiska data. I denna studie har OI satts till 30 dagar, ett värde som eventuellt skulle kunna optimeras. Denna s.k. optimerings-simulering syftar till att bestämma vilken säkerhetstid man borde ha haft under föregående period för att nå $S_{mål}$. Genom att köra optimeringssimuleringen med olika säkerhetstider erhålls också olika servicenivåer (s) som kan jämföras med $S_{mål}$ i syfte att avgöra vilken säkerhetstid som ger minst avvikelse från målet. När den lämpligaste säkerhetstiden (ST^*) fastställts sätts säkerhetstiden (ST) för artikeln till denna nivå för att användas under kommande period. Det görs med andra ord ett antagande om att den säkerhetstid som var lämplig under föregående period också är lämplig under kommande period. En principskiss över modellen kan ses i figur 6.



Figur 6 Principskiss för modellen simulering.

$S_{mål}$ är det yttersta målet om servicenivå för artikeln. För att uppnå vidare styrning av systemet används dock vid optimeringen en dynamisk målservicenivå (S_d^*) som justeras beroende på hur artikelns servicenivå ser ut vid ett givet tillfälle. Om t.ex. en artikel visar en erhållen servicenivå (S) på 93 procent vid ett visst tillfälle och $S_{mål}$ är 95 procent kommer algoritmen att försöka rätta till detta genom att sätta S_d^* till 97 procent under kommande period. På motsvarande sätt används en lägre S_d^* ifall S är högre än $S_{mål}$. Genom användandet av dynamisk målservicenivå tar algoritmen automatiskt hänsyn till efterfrågetyper där man systematiskt måste sikta mot en högre eller lägre servicenivå för att nå sitt mål. För att undvika alltför dramatiska variationer av den dynamiska målservicenivån utjämnas S_d^* med exponentiell utjämning med utjämningskonstant 0,2. Vilket som är det optimala värdet på denna konstant är ännu inte utrett, men empirisk prövning har visat att förekomsten av exponentiell utjämning förbättrar prestationen.

Eftersom det sällan går att åstadkomma en exakt servicenivå ges möjligheten att definiera vilket intervall på servicenivån som är acceptabelt genom användandet av toleranser. En tolerans neråt (τ_n) på 0,01 och en tolerans uppåt (τ_u) på 0,02 innebär t.ex. vid en önskad målservicenivå på 95 procent att servicenivåer mellan 94 procent (S_{min}^*) och 97 procent (S_{max}^*) är acceptabla och att uppnådda servicenivåer i detta intervall inte leder till vidare optimeringssimuleringar. Detta är de toleranser som har tagits fram genom empirisk testning. Det faktum att τ_n är mindre än τ_u grundas på att underprestation har ansetts mindre acceptabelt än överprestation. Eftersom S_d^* teoretiskt kan nå värden både betydligt under och över $S_{mål}$ måste eventuellt S_{min}^* och S_{max}^* justeras så att inte orimliga värden används. Den övre gränsen kan naturligtvis inte vara över 100 % och den undre gränsen för acceptabel servicenivå sätts aldrig heller till mindre än S_{min}^* även om en längre period av väldigt hög servicenivå annars ur ett teoretiskt perspektiv skulle innebära att man kunde sikta väldigt lågt för att över en artikels livslängd totalt sett komma nära $S_{mål}$. Det antas dock orimligt att någonsin medvetet erbjuda lägre servicenivå än S_{min}^* .

Ifall många val av säkerhetstid ger samma bästa servicenivå i optimeringssimuleringen, strävar algoritmen efter att välja en lägre säkerhetstid vid överprestation, dvs. $S > S_{mål}$, och en högre säkerhetstid vid underprestation, dvs. $S < S_{mål}$. Ifall det inte går att åstadkomma en servicenivå som uppfyller $S_{min}^* \leq s \leq S_{max}^*$ ger algoritmen den säkerhetstid som kommer närmast S_d^* . Även i detta senare fall tas hänsyn till överprestation och underprestation. Om det t.ex. var omöjligt att erhålla mer än 75% servicenivå oavsett vilket värde som valdes på ST under en optimeringsperiod men $S_{mål} = 95\%$ kommer algoritmen att välja ett högre ST . Säkerhetstiden kan som minst bli 0 och som mest maxtiden för den aktuella artikeln. I tabell 8 sammanfattas alla beteckningar som används i denna modell.

Tabell 8 Beteckningar för parametrar och variabler i modellen simulering av lämplig säkerhetstid.

OI	Optimeringsintervall, så många tidsenheter det ska förflyta mellan varje optimeringssimulering och eventuellt fastställande av ny säkerhetstid
$S_{mål}$	Den målservicenivå som från början definierats som mål för aktuell artikel
S_d^*	Den dynamiskt justerade målservicenivå man just nu önskar uppnå
τ_u	Tolerans uppåt för vilken servicenivå som kan accepteras
τ_n	Tolerans neråt för vilken servicenivå som kan accepteras
S_{min}^*	Den undre gränsen för vilken servicenivå som anses acceptabel
S_{max}^*	Den övre gränsen för vilken servicenivå som anses acceptabel
S	Aktuell erhållen servicenivå vid ett visst mättillfälle
s	Servicenivå som uppnåtts vid en optimeringssimulering
ST	Aktuell säkerhetstid för artikeln
ST^*	Den säkerhetstid som man efter en simulering över historiska data konstaterar att man borde ha haft för att komma så nära $S_{mål}$ som möjligt

6 Resultat

I detta kapitel presenteras resultaten från samtliga simuleringar, såväl med genererad som verklig efterfrågan. De dynamiska modellernas prestation jämförs med originalmodellen. Resultaten från två specialtester med förbättrad prognos och tillåtande av negativ säkerhetstid presenteras också. Under rubrikerna "Modellernas prestation vid klassificering" presenteras löpande resultat utifrån ett klassificeringsperspektiv, d.v.s. utifrån efterfrågetyp och ledtid.

De dynamiska modellerna *traditionell algoritm*, *traditionell algoritm med justeringar* samt *simulering* har testats med hjälp av simuleringar, med både genererad och verklig efterfrågedata, enligt beskrivningen i avsnitt 3.4.2 Undersökningsupplägg. Resultaten från dessa simuleringar presenteras i avsnitt 6.1 till 6.3. I stycke 6.2.1 Test med bättre prognos redovisas en undersökning av den dynamiska modellen *traditionell algoritm med justeringar och tidsbaserad avvikelse* med en mer följsam prognos än exponentiell utjämning. Eftersom det endast är då det finns en mer avancerad prognos som den tidsbaserade avvikelsen ger en betydande skillnad mot den vanligt beräknade avvikelsen testas inte den tidsbaserade avvikelsen i de grundläggande simuleringfallen. Avslutningsvis presenteras i avsnitt 6.4 resultat från några simuleringar där negativa säkerhetstider tillåts.

För att få en jämförelse mellan de dynamiska modellerna och originalmodellen vid genererad efterfrågan används två referensfall: *original(optimal)* respektive *original(statisk)*, se avsnitt 3.4.2 Undersökningsupplägg.

De mätvärden som presenteras i detta kapitel är servicenivåns avvikelse från riktvärdet, servicenivåns medelvärde, servicenivåns standardavvikelse, totalt medellager samt antal enheter i lager per procentenhet av den uppnådda servicenivån. För beskrivning av hur värdena har tagits fram, se avsnitt 3.4.2 Undersökningsupplägg.

För alla mått utom antal enheter i lager per procentenhet av den uppnådda servicenivån har hypotestest utförts. Inom respektive tabell är i de flesta fall de olika modellernas värden på dessa mått signifikant skilda från varandra på signifikansnivån 1 procent. De skillnader som inte är signifikanta redovisas efter hand som dessa resultat presenteras. Icke signifikanta skillnader förekommer främst för de lågrörliga artiklarna med genererad efterfrågan utan systematiska variationer. När inget annat anges är följaktligen skillnaderna signifikanta på nivån 1 procent.

6.1 Genererad efterfrågan med endast slumpmässiga variationer

I detta avsnitt presenteras resultaten från artiklar med en jämn efterfrågan, d.v.s. en efterfrågan med enbart slumpmässiga och inga systematiska variationer.

6.1.1 Högrörliga artiklar

Här presenteras resultaten för högrörliga artiklar, d.v.s. artiklar med ankomstintensitet 50, 10 respektive 3 per dag. Generella iakttagelser av resultaten för målservicenivån 95 procent kan göras i bilaga 1, där genomsnittsvärden för 30 simuleringar presenteras. Ju längre ledtiden är i förhållande till maxtiden, desto svårare är det att uppnå en hög servicenivå oavsett använd modell. Sett till de olika efterfrågestrukturerna kan konstateras att ju mer högrörlig artikel (d.v.s. ju högre ankomstintensitet) desto lättare är det att uppnå en hög servicenivå. Också denna tendens gäller för samtliga modeller.

Simuleringarna visar att antalet inleveranser ökar med de dynamiska modellerna jämfört med när en statisk säkerhetstid används. Vissa perioder, då den dynamiska säkerhetstiden är nära eller lika med maxtiden, uppkommer inleveranser varje dag, medan de statiska modellerna under samma period har en relativt jämn fördelning av inleveransdagar. I takt med att antalet inleveranser ökar minskar den genomsnittliga inleveranskvantiteten.

Målservicenivå 95 procent

Sett till servicenivåns avvikelse från riktvärdet och servicenivåns medelvärde i tabell 9 överträffar ingen metod *original(optimal)*. Samtliga dynamiska metoder är dock bättre än *original(statisk)* och de mer avancerade dynamiska metoderna är bättre än *traditionell algoritm*. Förutom den totalt sett uppnådda servicenivån är det dock även viktigt att se till servicenivåns standardavvikelse. När det gäller denna faktor överträffas *original(optimal)* av såväl *traditionell algoritm* som *traditionell algoritm med justeringar*.

Det faktum att *original(statisk)* har en mycket låg standardavvikelse beror på att servicenivån i många fall tangerar 100 procent långa perioder, vilket ger en liten standardavvikelse. Eftersom själva servicenivån dock ligger långt ifrån målet måste *original(statisk)* ändå ses som sämre än övriga modeller.

Traditionell algoritm med justeringar är bättre på att sikta in sig mot målservicenivån än *traditionell algoritm*, dock uppstår större variation i servicenivån då justeringarna används. *Simulering* är den av de dynamiska modellerna som närmast uppnår servicenivån, men även denna lider av bristen att standardavvikelsen är hög.

Tabell 9 Servicenivå för genererad efterfrågan med endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 95 %.

Modell	Servicenivåns avvikelse från riktvärde	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,119	97,27%	1,769
Original (statisk)	1,196	98,85%	0,857
Traditionell algoritm	0,354	97,73%	1,658
Traditionell algoritm med justeringar	0,252	97,56%	1,716
Simulering	0,234	97,38%	1,852

Tabell 10 visar hur de olika modellerna presterar i fråga om lagernivå. Det står klart att även ur detta hänseende är *original(optimal)* den som presterar bäst. Dock får samtliga dynamiska modeller anses ha en medellagernivå som trots en viss skillnad nästan ligger i paritet med *original(optimal)*. Det bör särskilt uppmärksammas att den något lägre medellagernivån för *original(optimal)* jämfört med *traditionell algoritm* i princip utjämnas av att *traditionell algoritm* har en något högre medelservicenivå. Detta illustreras av att det endast är en liten skillnad mellan antal lagerenheter per procentenhet uppnådd servicenivå i tabell 10. *Original(statisk)* framstår åter som klart sämst av modellerna. Den modell av de dynamiska som lyckas hålla nere lagernivån mest är *traditionell algoritm*.

Tabell 10 Medellager för genererad efterfrågan med endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 95 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	85 809	1 443
Original (statisk)	112 945	1 884
Traditionell algoritm	86 142	1 447
Traditionell algoritm med justeringar	87 380	1 469
Simulering	86 737	1 459

Sammanfattningsvis kan konstateras att *original(optimal)* ger bäst resultat, med reservation för att servicenivåns standardavvikelse är högre än i de flesta andra fall. Rangordningen av de dynamiska modellerna är inte självklar då olika modeller presterar bäst utifrån olika aspekter.

Målservicenivå 98 procent

Förhållandena mellan modellerna är desamma som för målservicenivån 95 procent, förutom att *traditionell algoritm med justeringar* i detta fall är något bättre än *simulering* på att styra in mot målservicenivån (se tabell 11).

Traditionell algoritm med justeringar och *simulering* uppvisar värden som ligger nära *original(optimal)*. Skillnaden mellan servicenivåns medelvärde för *original(optimal)* och *traditionell algoritm med justeringar* är inte signifikant. Skillnaden mellan *original(optimal)* och *simulering* är för samma mått signifikant först på nivån 10 procent. Skillnaden mellan servicenivåns standardavvikelse för *original(optimal)* och *traditionell algoritm med justeringar* är inte signifikant på nivån 1 procent, dock på nivån 5 procent.

Tabell 11 Servicenivå för genererad efterfrågan med endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 98 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,130	98,17%	1,489
Original (statisk)	0,782	98,86%	0,873
Traditionell algoritm	0,326	98,54%	1,350
Traditionell algoritm med justeringar	0,184	98,16%	1,538
Simulering	0,206	98,22%	1,622

Tabell 12 visar att även med målservicenivån 98 procent blir förhållandena mellan de olika metodernas medellager ungefär desamma som med målservicenivån 95 procent. Det bör observeras att *traditionell algoritm med justeringar* och *simulering* har ökat medellagret avsevärt jämfört med då målservicenivån är 95 procent jämfört med övriga metoders ökning.

Tabell 12 Medellager för genererad efterfrågan med endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 98 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	86 654	1 454
Original (statisk)	112 974	1 885
Traditionell algoritm	87 513	1 466
Traditionell algoritm med justeringar	90 835	1 525
Simulering	90 633	1 520

Modellernas prestation vid klassificering

Den rangordning av hur bra modellerna är på att styra in mot önskad målservicenivå som presenteras i tabell 9 respektive är ungefär densamma när man betraktar resultaten uppdelat för varje ankomstintensitet (se bilaga 4 tabell 1 och tabell 3). Det som varierar är vilken som är bäst av *traditionell algoritm med justeringar* och *simulering*. Detsamma gäller när resultaten visas per ledtid (se bilaga 4 tabell 2 och tabell 4), dock med skillnaden att vid den längsta ledtiden är *traditionell algoritm* bättre än de båda andra dynamiska modellerna på att styra in mot målservicenivån.

Förmågan att styra in mot önskad servicenivå har en tendens att försämrans mer för *traditionell algoritm med justeringar* samt *simulering* än för övriga modeller för lägre ankomstintensiteter. Detta innebär att även om dessa två modeller fortfarande är betydligt bättre på att styra in mot målservicenivån även vid lägre ankomstintensiteter, så är inte

skillnaden lika påtaglig som vid höga ankomstintensiteter. Denna tendens gäller framför allt för målservicenivån 95 procent.

I takt med att ledtiden blir längre blir *optimal(statisk)*, *traditionell algoritm med justeringar* och *simulering* sämre på att styra in mot målservicenivån. Denna tendens finns i viss grad även hos *original(optimal)*, om än inte lika tydligt. Däremot finns ingen tydlig trend hos *traditionell algoritm*, som tvärtom övriga verkar prestera som bäst vad gäller instyrning då ledtiden är lång.

6.1.2 Lågrörliga artiklar

De lågrörliga artiklarna är genererade på samma sätt som de högrörliga. Rörligheten ligger även här på tre olika nivåer, med intensiteterna 1/2, 1/10 respektive 1/40 per dag. Generella iakttagelser som kan göras i bilaga 2 är att det krävs långa maxtider för att uppnå en tillräckligt hög servicenivå när efterfrågan är så här låg och ojämn. Det är de många fallen med korta maxtider som gör att medelvärdena för servicenivån i tabellerna i detta avsnitt är så låga. Högre servicenivåer uppnås också då ledtiden är lång.

Resultaten från de trettio olika simuleringarna skiljer sig i hög grad från varandra. Medelvärdet av uppnådda servicenivåer för en viss modell kan variera tio till tjugoprocentenheter mellan de olika simuleringarna, trots att en simulering är 6000 dagar lång. Standardavvikelseerna av servicenivåmedelvärdena är generellt en faktor tio större för de lågrörliga artiklarna än för de högrörliga.

Målservicenivå 95 procent

Även för de lågrörliga artiklarna är *original(optimal)* den modell som styr in närmast målservicenivån då denna är 95 procent, se tabell 13. Rangordningen för samtliga modellers instyrning är densamma som för de högrörliga artiklarna.

Tabell 13 Servicenivå för genererad efterfrågan med endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 95 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,200	89,87%	9,199
Original (statisk)	1,101	89,71%	8,749
Traditionell algoritm	0,790	89,03%	9,462
Traditionell algoritm med justeringar	0,624	89,25%	9,682
Simulering	0,494	89,32%	9,708

Servicenivåns medelvärde och standardavvikelse är mycket lika för de olika modellerna. I många fall är skillnaderna inte signifikanta. I tabell 14 redovisas vilka skillnader som inte är signifikanta, vilket anges som ”nej”, och vilka som är signifikanta, vilket anges med aktuell signifikansnivå. Det framgår tydligt att skillnaderna mellan de dynamiska modellerna inte är signifikanta.

Tabell 14 Signifikans i skillnaderna mellan modellernas uppnådda servicenivåers medelvärde och standardavvikelse.

	Original (statisk)	Traditionell algoritm	Traditionell algoritm med justeringar	Simulering
Original (optimal)	Medel: Nej Stdav: 10%	Medel: 1% Stdav: Nej	Medel: 5% Stdav: 10%	Medel: 5% Stdav: 5%
Original (statisk)		Medel: 1% Stdav: 1%	Medel: 10% Stdav: 1%	Medel: Nej Stdav: 1%
Traditionell algoritm			Medel: Nej Stdav: Nej	Medel: Nej Stdav: Nej
Traditionell algoritm med justeringar				Medel: Nej Stdav: Nej

Även medellagernivåerna är mycket lika för de olika modellerna, som synes i tabell 15. Skillnaden mellan *original(optimal)* och *traditionell algoritm* är inte signifikant. Det är inte heller medellagerskillnaden mellan *traditionell algoritm med justeringar* och *simulering*. Liksom för de högrörliga artiklarna bygger *original(statisk)* upp ett större lager än övriga modeller och *traditionell algoritm* är den av de dynamiska modellerna som genererar lägst lagernivå.

Tabell 15 Medellager för genererad efterfrågan med endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 95 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	2 148	38
Original (statisk)	2 392	42
Traditionell algoritm	2 140	38
Traditionell algoritm med justeringar	2 227	40
Simulering	2 239	40

Målservicenivå 98 procent

Den stora skillnaden mot både målservicenivån 95 procent och de högrörliga fallen är att *traditionell algoritm* har en lägre avvikelse från servicenivåns riktvärde än *traditionell algoritm med justeringar*, se tabell 16. Dock är skillnaden mellan dessa två värden inte signifikant. I övrigt stämmer rangordningen av de olika modellernas instyrning överens med tidigare fall.

Tabell 16 Servicenivå för genererad efterfrågan med endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 98 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,171	90,18%	8,916
Original (statisk)	1,036	89,14%	9,181
Traditionell algoritm	0,692	89,33%	9,438
Traditionell algoritm med justeringar	0,730	89,10%	9,869
Simulering	0,452	89,60%	9,607

Även med denna högre målservicenivå saknas det i hög grad signifikanta skillnader mellan de olika modellernas servicenivåmedelvärden och standardavvikelser. Det går inte att se några tydliga mönster i jämförelsen mellan tabell 14 och tabell 17, men även i detta fall är det i stor utsträckning mycket lika värden för de dynamiska modellerna (även om det förekommer två signifikanta skillnader på nivån 5 procent).

Tabell 17 Signifikans i skillnaderna mellan modellernas uppnådda servicenivåers medelvärde och standardavvikelse.

	Original (statisk)	Traditionell algoritm	Traditionell algoritm med justeringar	Simulering
Original (optimal)	Medel: 1% Stdav: Nej	Medel: 1% Stdav: 5%	Medel: 1% Stdav: 1%	Medel: 5% Stdav: 1%
Original (statisk)		Medel: Nej Stdav: Nej	Medel: Nej Stdav: 1%	Medel: 10% Stdav: 5%
Traditionell algoritm			Medel: Nej Stdav: 5%	Medel: Nej Stdav: Nej
Traditionell algoritm med justeringar				Medel: 5% Stdav: Nej

Original(statisk) har enligt tabell 18 fortfarande högst medellager. Numera har både *traditionell algoritm* och *traditionell algoritm med justeringar* lägre lagernivå än *original(optimal)*. Jämförelse mellan tabell 16 och tabell 18 visar dock att den lägre medellagernivån leder till lägre servicenivå. Skillnaden i lagernivå mellan *original(optimal)* och *traditionell algoritm med justeringar* är inte signifikant. Det är inte heller skillnaden mellan *original(statisk)* och *simulering*.

Tabell 18 Medellager för genererad efterfrågan med endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 98 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	2 243	39
Original (statisk)	2 368	41
Traditionell algoritm	2 199	39
Traditionell algoritm med justeringar	2 240	40
Simulering	2 384	42

Modellernas prestation vid klassificering

Vad gäller prestation vid olika ledtider (se bilaga 4 tabell 6 och tabell 8) presterar samtliga modeller bättre vid långa ledtider. Sett till olika ankomstintensiteter (se bilaga 4 tabell 5 och tabell 7) blir de dynamiska modellerna sämre på att styra in ju lägre ankomstintensiteten är. För originalmodellerna kan ingen tydlig tendens iaktas gällande prestationens beroende av ankomstintensiteten.

6.2 Genererad efterfrågan med systematiska variationer

I detta avsnitt presenteras resultaten från artiklar där efterfrågan innehåller olika former av systematisk variation. I bilaga 3 framgår att det för lång och kort säsong är omöjligt att uppnå målservicenivån vid längre ledtider, om man inte sätter maxtiden mycket högt. För veckosäsongen är det i princip alltid möjligt att uppnå målservicenivån.

Precis som i fallet med högrörliga artiklar utan systematiska variationer sker fler men mindre inleveranser med de dynamiska modellerna än med de statiska. Man kan särskilt se ett samband mellan att efterfrågan har en uppåtgående trend och att inleveranser sker tätare, ibland varje dag, eftersom den dynamiska säkerhetstiden i dessa perioder närmar sig maxtiden.

Målservicenivå 95 procent

Då efterfrågan innehåller systematiska variationer är det inte längre *original(statisk)* utan *traditionell algoritm* som uppvisar sämst resultat i uppnådd servicenivå såväl vad gäller instyrning som standardavvikelse, se tabell 19. *Original(optimal)* ger fortfarande bäst instyrning, men även *traditionell algoritm med justeringar* och *simulering* ger relativt bra instyrning. Av dessa styr *simulering* in närmast, dock med högre standardavvikelse. Skillnaden i servicenivåns medelvärde mellan *original(optimal)* och *simulering* är inte signifikant.

Tabell 19 Servicenivå för genererad efterfrågan med systematiska variationer, målservicenivå 95 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,098	91,95%	6,223
Original (statisk)	1,430	93,19%	5,218
Traditionell algoritm	1,764	90,53%	6,917
Traditionell algoritm med justeringar	0,867	92,30%	5,977
Simulering	0,340	91,96%	6,417

I tabell 20 kan utläsas att *traditionell algoritm* ger betydligt lägre medellager än övriga modeller, även utslaget per procentenhet uppnådd servicenivå. Dock måste detta ställas i relation till att *traditionell algoritm* enligt tabell 19 uppnår en alldeles för låg medelservicenivå och har stor avvikelse till riktvärdet. Förutom *traditionell algoritm* är *original(optimal)* den modell som ger lägst medellager. Även *traditionell algoritm med justeringar* har en låg medellagernivå.

Tabell 20 Medellager för genererad efterfrågan med systematiska variationer, målservicenivå 95 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	127 139	2 309
Original (statisk)	146 383	2 598
Traditionell algoritm	117 849	2 181
Traditionell algoritm med justeringar	129 247	2 341
Simulering	137 201	2 484

Sett till helheten av uppnådda servicenivåer och medellagernivåer kan *optimal(original)*, *traditionell algoritm med justeringar* och *simulering* anses vara de bästa modellerna med viss fördel för *traditionell algoritm med justeringar* framför *simulering* p.g.a. den låga medellagernivån och standardavvikelsen. Det är i detta fall inte helt självklart att *original(optimal)* är bättre än *traditionell algoritm med justeringar* och *simulering*.

Målservicenivå 98 procent

Förhållandena mellan modellerna i tabell 21 och tabell 22 är desamma som för målservicenivå 95 procent, förutom att *original(optimal)* nu har något lägre standardavvikelse för servicenivån än *traditionell algoritm med justeringar*. Dessutom är i detta fall skillnaden i medellager mellan *original(optimal)* och *traditionell algoritm med justeringar* inte signifikant (dock på nivån 10 procent).

Tabell 21 Servicenivå för genererad efterfrågan med systematiska variationer, målservicenivå 98 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (optimal)	0,119	93,14%	5,441
Original (statisk)	1,020	93,19%	5,218
Traditionell algoritm	1,756	91,11%	6,680
Traditionell algoritm med justeringar	0,780	92,73%	5,738
Simulering	0,358	92,81%	6,022

Tabell 22 Medellager för genererad efterfrågan med systematiska variationer, målservicenivå 98 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (optimal)	133 300	2 389
Original (statisk)	146 383	2 598
Traditionell algoritm	120 489	2 219
Traditionell algoritm med justeringar	133 694	2 411
Simulering	148 348	2 652

Modellernas prestation vid klassificering

Den rangordning för hur bra modellerna styr in mot målservicenivån som framgår i tabell 19 och tabell 21 är även giltig då lång respektive kort säsong studeras separat (se bilaga 4 tabell 9 och tabell 13). För veckosäsong är rangordningen nästan samma, *original(statisk)* kommer dock före *traditionell algoritm*.

För lång och kort säsong gäller generellt att *traditionell algoritm* och *traditionell algoritm med justeringar* har en tendens att prestera bättre vid korta ledtider än vid långa (se bilaga 4 tabell 10, tabell 11, tabell 14 och tabell 15). Detta gäller även *original(statisk)* när målservicenivån är 98 procent. Då den är 95 procent är dock *original(statisk)* lite bättre vid långa ledtider. *Simulering* tenderar att vara bättre vid kortare och längre ledtider än vid medellånga. För veckosäsong (se bilaga 4 tabell 12) och målservicenivå 95 procent gäller att all instyrning blir sämre i takt med att ledtiden blir längre. För 98 procent (se bilaga 4 tabell 16) gäller att instyrningen är bäst vid korta och långa ledtider. Rangordningen för vilka modeller som presterar bäst beroende på ledtiden varierar inte i så hög grad och inte efter något så generellt mönster att man kan dra några allmängiltiga slutsatser utifrån det.

6.2.1 Test med bättre prognos

Här presenteras resultaten från tester med en bättre prognos än exponentiell utjämning (för beskrivning av prognosen se stycke 3.4.2 Undersökningsupplägg). Fokus ligger dels på modellen *traditionell algoritm med justeringar* och *tidsbaserad avvikelse* (nedan kallad

enbart *tidsbaserad*), som jämförs med *traditionell algoritm med justeringar*, och dels på att jämföra hur originalmodellen presterar med en bättre prognos kontra en sämre.

Målservicenivå 95 procent

Då resultaten för modellerna *original(statisk)* och *traditionell algoritm med justeringar* i tabell 23 och tabell 24 jämförs med resultaten i tabell 19 och tabell 20 framgår det tydligt att modellerna presterar avsevärt mycket bättre med den bättre prognosen. Servicenivån styrs in betydligt närmare målet, med en mycket lägre standardavvikelse. Dessutom är servicenivån i medeltal ansenligt mycket högre samtidigt som medellagernivån är mycket lägre. Samtliga dessa skillnader är signifikanta på 1 procents signifikansnivå.

Tillägget att använda tidsbaserad avvikelse till *traditionell algoritm med justeringar* ger en något lägre standardavvikelse hos servicenivån. Denna skillnad är dock inte signifikant på 1 procentsnivån, men däremot på nivån 5 procent. Avvikelsen från riktvärdet blir större för *tidsbaserad* än för *traditionell algoritm med justeringar*, vilket avspeglas i en högre medelservicenivå. Detta sker på bekostnad av att lager byggs upp, vilket sker i högre takt än servicenivån ökar enligt måttet antal lagerenheter per procentenhet uppnådd servicenivå.

Tabell 23 Servicenivå för genererad efterfrågan med systematiska variationer då en bättre prognos än exponentiell utjämning används, målservicenivå 95 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (statisk)	1,184	99,38%	0,734
Traditionell algoritm med justeringar	0,502	98,37%	1,479
Tidsbaserad	0,666	98,64%	1,429

Tabell 24 Medellager för genererad efterfrågan med systematiska variationer då en bättre prognos än exponentiell utjämning används, målservicenivå 95 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (statisk)	118 893	1 985
Traditionell algoritm med justeringar	91 929	1 548
Tidsbaserad	95 323	1 603

Målservicenivå 98 procent

Resultaten i tabell 25 och tabell 26 kan jämföras med tabell 21 och tabell 22. Dessa resultat uppvisar samma mönster som då målservicenivån är 95 procent. Skillnaden i servicenivåns standardavvikelse mellan *tidsbaserad* och *traditionell algoritm med justeringar* är i detta fall inte signifikant.

Tabell 25 Servicenivå för genererad efterfrågan med systematiska variationer då en bättre prognos än exponentiell utjämning används, målservicenivå 98 %.

Modell	Servicenivåns avvikelse från riktvärdet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original (statisk)	0,792	99,38%	0,734
Traditionell algoritm med justeringar	0,337	98,97%	1,184
Tidsbaserad	0,413	99,07%	1,173

Tabell 26 Medellager för genererad efterfrågan med systematiska variationer då en bättre prognos än exponentiell utjämning används, målservicenivå 98 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original (statisk)	118 893	1 985
Traditionell algoritm med justeringar	94 278	1 583
Tidsbaserad	98 604	1 654

6.3 Verklig efterfrågan

Vid tester med verklig efterfrågan nåddes de resultat som presenteras i tabell 27, tabell 28 och tabell 29. Alla de dynamiska modellerna lyckas minska servicenivåns avvikelse från riktvärdet samtidigt som servicenivåns medelvärde höjs och därmed närmar sig den önskade nivån på 98 procent medan servicenivåns standardavvikelse sänks. De dynamiska modellerna höjer förvisso medellagret med 6 till 20 procent jämfört med originalmodellen, men i tabell 29 framgår att de dynamiska modellerna samtidigt sänker underprestationen i uppnådd servicenivå. Jämfört med originalmodellen innebär detta en sänkning med 23 till 44 procent.

Tabell 27 Servicenivå för verklig efterfrågan, målservicenivå 98 %.

Modell	Servicenivåns avvikelse från målet	Servicenivåns medelvärde	Servicenivåns standardavvikelse
Original	0,015	96,64%	0,059
Traditionell algoritm	0,012	96,89%	0,055
Traditionell algoritm med justeringar	0,008	97,03%	0,052
Simulering	0,009	96,84%	0,057

Tabell 28 Medellager för verklig efterfrågan, målservicenivå 98 %.

Modell	Totalt medellager	Antal lagerenheter per procentenhet uppnådd servicenivå
Original	414	429
Traditionell algoritm	441	449
Traditionell algoritm med justeringar	466	477
Simulering	497	509

Tabell 29 Underprestation i uppnådd servicenivå för verklig efterfrågan, målservicenivå 98 %.

Modell	Total underprestation
Original PipeChain	0,425
Traditionell algoritm	0,326
Traditionell algoritm med justeringar	0,237
Simulering	0,292

6.4 Tester med negativa säkerhetstider

Då säkerhetstiden tillåts vara negativ ger *traditionell algoritm* generellt sett bättre instyrning mot målservicenivån i fall med mycket lång maxtid, vilket innebär att överprestationen i förhållande till målet minskar. Medellagret blir samtidigt lite lägre. De negativa säkerhetstiderna orsakar inte någon underprestation, d.v.s. fall där instyrningen var god redan innan negativa säkerhetstider tillåts påverkas ej. *Traditionell algoritm med justeringar* uppvisar ungefär samma tendenser, dock med skillnaden att sänkningen av överprestation i många fall sker till priset av en ökning av medellagret.

Tester med att tillåta negativa säkerhetstider ner till minus maxtiden för *simulering* visar att instyrningen mot den angivna servicenivån lyckas mycket bra för korta ledtider (2 och 5) förutom vid den längsta maxtiden (60). För längre ledtider uppstår däremot i allmänhet en systematisk underprestation som blir extra stor i kombination med en lång maxtid.

7 Analys

I detta kapitel analyseras de presenterade resultaten. Diskussion förs även kring hur en dynamisk säkerhetstid skulle fungera ur ett VMI-perspektiv.

Resultaten från simuleringarna med genererad efterfrågan visar att de dynamiska modellerna alla har en god instyrningsförmåga mot den önskade servicenivån och minskar avvikelsen till målservicenivån i relation till *original(statisk)*. Även medellagret blir i de allra flesta fall lägre. Detta gäller vid såväl 95 som 98 procents målservicenivå. För genererad efterfrågan med systematiska variationer presterar dessutom de dynamiska modellerna i paritet med *original(optimal)*. Då verklig data används uppnås en prestation i paritet med eller bättre än originalmodellens, vilket illustrerar att de dynamiska modellerna även kan fungera i praktiken.

7.1 Genererad efterfrågan med endast slumpmässiga variationer

Normalt sett kan inte de dynamiska modellerna överträffa *original(optimal)*, vilket indikerar att man vid en mycket hög kunskapsnivå om lagerstyrning och parametersättning samt vid en mycket förutsägbar efterfrågan utan systematiska variationer bör använda en statisk säkerhetstid. Eftersom *original(optimal)* använder en säkerhetstid som räknats ut som optimal i efterhand ger dock denna modell realistiskt bra resultat och i praktiken är de dynamiska modellerna ett bra och realistiskt alternativ för val av säkerhetstid även vid avsaknad av systematiska variationer.

Resultaten av hur *original(statisk)* presterar visar att en statisk säkerhetstid som inte är tillräckligt bra vald är klart sämre än att använda någon av de dynamiska modellerna, med hänsyn tagen till såväl instyrning mot en viss servicenivå som erhållen medellagernivå. Med hänsyn till vad som tidigare framgått om att företag ofta bestämmer en säkerhetstid på ganska vaga grunder är det rimligt att anta att *original(statisk)* ger en bra bild av hur säkerhetstiden många gånger sätts i praktiken. Det är därför högst troligt att det i många fall skulle vara rekommenderbart att istället använda en dynamisk säkerhetstid.

7.1.1 Högrörliga artiklar

Traditionell algoritm med justeringar och *simulering* ger mindre avvikelse från målservicenivån än *traditionell algoritm*. Den sistnämnda ger dock lägre medellager till en högre genomsnittlig servicenivå och kan därför anses vara den mest fördelaktiga modellen. Generellt sett ger en dynamiskt förändrad säkerhetstid höjda medellager för högrörliga artiklar. *Traditionell algoritm* justerar inte säkerhetstiden i lika stor utsträckning som de andra dynamiska modellerna och därför är det inte förvånande att denna modell genererar lägst medellager. *Traditionell algoritm* uppvisar också lägre standardavvikelse än de andra dynamiska modellerna, vilket indikerar en större stabilitet i erhållen servicenivå.

Det faktum att *traditionell algoritm* styr in bäst mot målservicenivån för ankomstintensitet 50 per dag och sämst för intensiteten 3 per dag bör kunna förklaras av att algoritmen grundar sig på antagande om normalfördelning, ett antagande som är mer sant ju mer

högrörlig artikeln är. Oavsett ankomstintensitet är dock de andra två dynamiska modellerna bättre på att styra in mot önskad servicenivå, även om skillnaden är mindre för högrörliga artiklar. När ledtiden är lång framstår *traditionell algoritm* ännu mer tydligt som den bästa dynamiska modellen, eftersom den då är den modell som styr in närmast målservicenivån.

Det är mycket svårt att avgöra vilken modell som presterar bäst av *traditionell algoritm med justeringar* och *simulering*. Skillnaderna mellan modellernas resultat är visserligen signifikanta, men de är ändå inte särskilt stora och varierar dessutom beroende på målservicenivå. Då målservicenivån är 95 procent är *simulering* lite bättre på att styra in mot riktvärdet, medan motsatt förhållande gäller vid en önskad servicenivå på 98 procent. Den modell som styr in sämst i respektive fall ger något högre medelvärde på servicenivån. Oavsett målservicenivå ger *traditionell algoritm med justeringar* den mest stabila servicenivån (d.v.s. lägst standardavvikelse) medan *simulering* ger lägst medellager. Valet mellan modellerna skulle därför kunna avgöras genom att man tar ställning till om man vill prioritera en låg medellagernivå eller en stabil servicenivå.

7.1.2 Lågrörliga artiklar

Eftersom skillnaderna mellan de olika modellernas resultat är små och i många fall inte signifikanta är det svårt att bedöma vilka modeller som presterar bättre än andra. Dock finns det en tydlig tendens att *original(optimal)* är den bästa modellen även för de lågrörliga artiklarna. Denna bedömning grundar sig på att servicenivåns avvikelse från riktvärdet är signifikant lägre än övriga modellers och att servicenivåmedelvärdet är det signifikant högsta jämfört med de flesta andra modeller, samtidigt som standardavvikelsen relativt sett är ganska låg. Visserligen ger *original(optimal)* inte längre lägst medellager, men skillnaderna mot de andra modellernas medellager är små och i flera fall inte signifikanta, varför detta kan anses försumbart jämfört med fördelarna i uppnådd servicenivå.

Optimal(statisk) får anses vara den sämst presterande modellen även för lågrörliga artiklar, eftersom instyrningen är signifikant sämre än övriga modellers. Medellagret för *original(statisk)* är dessutom signifikant högre än samtliga andra modellers trots att servicenivåns medelvärde är ungefär likvärdiga för de olika modellerna.

Det är mycket svårt att jämföra de olika dynamiska modellernas prestation, då skillnaderna dem emellan i många fall inte är signifikanta. *Simulering* har både för målservicenivå 95 och 98 procent signifikant lägst avvikelse från servicenivåns riktvärde. Denna fördel motverkas dock av att *simulering* ger högst medellager. Dock är skillnaden mellan *simulering* och *traditionell algoritm med justeringar* inte signifikant i 95-procentsfallet. En jämförelse mellan *traditionell algoritm* och *traditionell algoritm med justeringar* för avvikelser från servicenivåns riktvärde visar att i 95-procentsfallet är *traditionell algoritm med justeringar* signifikant bättre, medan *traditionell algoritm* visar något bättre instyrning i 98-procentsfallet, dock utan signifikans. *Traditionell algoritm* har signifikant lägre medellager än *traditionell algoritm med justeringar*, vilket möjligen skulle kunna visa på att *traditionell algoritm* är något bättre. Då skillnaderna i servicenivåns medelvärde och standardavvikelse i många fall inte är signifikanta mellan de olika dynamiska modellerna

kan ingen bedömning göras utifrån det perspektivet. Sammantaget visar detta på att inga säkra uttalanden kan göras angående vilken dynamisk modell som är bäst för lågrörliga artiklar. Däremot kan det konstateras att samtliga dynamiska modeller ger godtagbara resultat i jämförelse med originalmodellen.

De otydliga skillnaderna mellan modellernas prestation har troligen att göra med den stora variation som har konstaterats mellan olika simuleringskörningar (se avsnitt 6.1.2 Lågrörliga artiklar). Slumpen får helt enkelt så stor inverkan att det inte går att urskilja variationer som beror på modellernas prestationsförmåga i de stora slumpmässiga variationerna. Detta gör också att en närmare analys av prestationen sett till olika ledtidsfall och efterfrågestrukturer omöjliggörs. Den enda generella tendens som kan utläsas är att alla de dynamiska modellerna styr in sämre mot servicenivåns riktvärde ju lägre ankomstintensiteten är. För *traditionell algoritm* och *traditionell algoritm med justeringar* torde detta vara en naturlig reaktion på att de fungerar sämre ju mindre normalfördelad efterfrågan är. Dock är det anmärkningsvärt att de faktiskt ger så bra resultat som de gör för de lågrörliga artiklarna, då antagandet om normalfördelning inte är uppfyllt i dessa fall.

7.2 Genererad efterfrågan med systematiska variationer

En stor skillnad mot efterfrågestrukturerna utan systematiska variationer är att *traditionell algoritm* visar sig fungera dåligt vid systematiska variationer och uppvisar den högsta avvikelser från riktvärdet, den lägsta medelservicenivån och den högsta standardavvikelsen. Att den har lägst medellager är därför inte mycket värt. En möjlig förklaring till *traditionell algoritms* dåliga prestation kan vara antagandet om normalfördelad efterfrågan, som i fallet med systematiska variationer knappast är uppfyllt.

Sett till avvikelser från riktvärdet är *original(optimal)* den bästa modellen även för systematiska förändringar. Det är dock svårt att i övrigt särskilja prestationen för *original(optimal)* från *traditionell algoritm med justeringars* prestation. För målservicenivå 95 procent har *traditionell algoritm med justeringar* något högre medelvärde och något lägre standardavvikelse för servicenivån jämfört med *original(optimal)*, medan förhållandet är det omvända för målservicenivå 98 procent. Även medellagernivåerna är lika för de två modellerna även om *traditionell algoritm med justeringar* generellt ligger något över *original(optimal)*. För målservicenivå 98 procent är skillnaden dock inte signifikant på nivån 5 procent.

Simulering har en lägre avvikelse från riktvärdet än *traditionell algoritm med justeringar* (dock högre än *original(optimal)*) och servicenivåns medelvärde ligger i paritet med utfallen från både *traditionell algoritm med justeringar* och *original(optimal)*. Dock har *simulering* ett betydligt högre medellager och därmed är *traditionell algoritm med justeringar* den rekommenderade modellen.

Även om *original(optimal)* bör anses vara den bästa modellen även när systematiska variationer förekommer i efterfrågan ligger prestationen för *traditionell algoritm med justeringar* så nära *original(optimal)* att de nästan kan anses likvärdiga. Eftersom det är rimligt att anta att det i verkligheten är mycket svårt att sätta en statisk säkerhetstid som är

så bra som *original(optimal)* kan därför antagandet göras att *traditionell algoritm med justeringar* i praktiken alltid är att rekommendera då det förekommer systematiska förändringar.

Även när man ser till olika ledtidfall och de specifika efterfrågestrukturerna presterar de olika modellerna på ungefär samma sätt. Det går inte att iaktta några skillnader eller mönster som föranleder en annorlunda analys i specifika fall än den som gjorts ovan.

7.2.1 Test med bättre prognos

Testet med en bättre prognos visar att den tidsbaserade avvikelserna inte ger bättre resultat, utan istället presterar sämre än då en vanlig standardavvikelse används. I de fall som Krupp presenterar (se avsnitt 2.4.2 Tidsbaserad avvikelse vid säkerhetslagerberäkning) där tidsbaserad avvikelse ska ge ett bättre resultat har efterfrågan en mycket tydlig trend, exempelvis vid slutet av en produktlivscykel. I denna studie består de systematiska variationerna istället av säsongsvariationer och detta kan vara förklaringen till att dessa resultat skiljer sig markant från Krupps resultat. En annan förklaring skulle kunna vara att det krävs en ännu bättre prognos än den som använts i detta test för att den tidsbaserade avvikelsernas fördelar ska visa sig. Den förbättrade prognos som använts i denna studie är en uppsättning värden med samma karaktär, slumpmässighet och avvikelser som den data som använts som efterfrågan. Detta skulle kunna betyda att prognosavvikelserna stundtals blir kraftiga till den tidsbaserade metodens nackdel. Kanske är dock den främsta orsaken till skillnaden mellan resultaten i denna studie respektive Krupps studie att den tidsbaserade avvikelserna i denna studie tillämpas i ett system med tidsfasad beställningspunkt. Hänsyn tas redan utan den tidsbaserade avvikelserna till framtida prognostiserad efterfrågan och detta skulle kunna leda till att den tidsbaserade avvikelserna blir överflödiga och till och med motverkad.

Den generella slutsatsen av testet med bättre prognos är att hur mycket man än arbetar med att förbättra systemets prestation med hjälp av t.ex. dynamisk säkerhetstid är i slutändan ändå det viktigaste att man har en tillräckligt bra prognos.

7.3 Sammanfattande analys för genererad efterfrågan

Sammanfattningsvis kan konstateras att oavsett typ av efterfrågan presterar de dynamiska modellerna tillräckligt bra för att vara ett alternativ till en statisk säkerhetstid. Det är endast då man med säkerhet vet att det inte förekommer systematiska variationer i efterfrågan som en statisk säkerhetstid skulle kunna antas vara bättre än en dynamisk och även i detta fall kan en dynamisk modell antas ge likvärdiga eller bättre resultat eftersom det krävs en mycket välvald statisk säkerhetstid för att överträffa den dynamiska.

Med utgångspunkt i ovanstående analys kan en sammanfattning angående val av dynamiska modeller göras, se tabell 30. Den rekommenderade modellen har valts ut som den dynamiska modell som presterar bäst om så många mått som möjligt tas i beaktande. Dessutom har övriga modeller som också presterar bra angetts som lämpliga modeller. Dessa rekommendationer bygger på en samlad bedömning av de olika måtten. Det måste

påpekas att detta är en klar generalisering och att vilken modell som är bäst måste avgöras genom att beakta vilket mått som är viktigast för den artikel man vill styra.

Tabell 30 Översikt över dynamiska modeller.

Typ av efterfrågan	Rekommenderad modell	Andra lämpliga modeller	Olämplig modell
Inga systematiska variationer – högrörliga artiklar	<i>Traditionell algoritm</i>	<i>Traditionell algoritm med justeringar, simulering</i>	-
Inga systematiska variationer – lågrörliga artiklar	<i>Traditionell algoritm med justeringar, traditionell algoritm, simulering</i>	-	-
Systematiska variationer	<i>Traditionell algoritm med justeringar</i>	<i>Simulering</i>	<i>Traditionell algoritm</i>

Det som är intressant att notera i ovanstående tabell är att *traditionell algoritm* är en olämplig modell vid systematiska variationer. Om man inte känner till om systematiska variationer förekommer är det därför bättre att välja *traditionell algoritm med justeringar* eller *simulering* som är rekommenderade eller lämpliga modeller i samtliga olika fall. Vilken av modellerna som ska väljas måste avgöras genom att beakta vilket mått som är viktigast vid styrningen. En allmängiltig rekommendation då man vill vara beredd på eventuella systematiska variationer skulle kunna vara att använda *traditionell algoritm med justeringar*. Motiveringen till detta val är att skillnaderna mellan denna modell och *simulering* är små i fall utan systematiska variationer, medan *traditionell algoritm med justeringar* har konstaterats vara bäst då systematiska variationer uppstår.

De iakttagelser som har gjorts i olika leddfall och för specifika efterfrågestrukturer skiljer sig i de flesta fall inte drastiskt från de övergripande iakttagelserna för respektive gruppering av fall (högrörliga artiklar, lågrörliga artiklar samt efterfrågan med systematiska variationer) och kan inte anses så allmängiltiga att de föranleder några särskilda rekommendationer i enskilda fall.

7.3.1 Instabilitet orsakad av dynamisk säkerhetstid

Ett problem som identifierats i studien är att det uppstår en ryckighet i beställningsförfarandet när en dynamisk säkerhetstid används. Detta visar sig bl.a. genom att det periodvis läggs beställningar varje dag. Det faktum att säkerhetslagret uttrycks som en tid innebär redan i sig självt att det faktiska säkerhetslagret i antal enheter ständigt kommer att variera utifrån aktuell efterfrågan och därmed att risken för ryckighet finns redan idag. Dock visar resultaten att originalmodellen med en statisk säkerhetstid ger ett betydligt jämnare inleveransmönster än de dynamiska modellerna.

Ifall täta beställningar och inleveranser innebär problem i ett visst flöde kan detta innebära att de dynamiska modellerna i sin nuvarande form inte är lämpliga att använda. Om täta

inleveranser är praktiskt genomförbart måste de eventuella nackdelar det kan medföra vägas mot de fördelar som påvisats hos de dynamiska modellerna. Möjligtvis skulle nackdelarna kunna motverkas genom ett närmare samarbete med leverantörer eller en ökad flexibilitet i godsmottagningen.

De täta inleveranserna uppstår när säkerhetstiden närmar sig maxtiden, vilket inträffar då de dynamiska modellerna räknat fram en hög säkerhetstid p.g.a. att det förekommit underprestation. En lösning skulle således kunna vara att sätta gränsen för den dynamiska säkerhetstiden något lägre än maxtiden. Detta innebär dock problem ifall maxtiden är mycket kort eftersom säkerhetstiden då i princip inte får något utrymme att variera. Problemet hade också kunnat undvikas om en längre maxtid användes. Om säkerhetstiden ofta räknas fram till att ligga nära maxtiden kan detta således tolkas som att en längre maxtid behövs.

7.4 Verklig efterfrågan

Den verkliga data som använts kommer från ett företag med en relativt hög mognadsgrad och en god kunskap om parametersättningen. Därför får det i detta fall betraktas som en framgång att få servicenivåavvikelser i samma storleksordning som originalmodellen. De dynamiska modellerna får här till och med lägre servicenivåavvikelse än originalmodellen, dock på bekostnad av ett lite högre medellager. Det faktum att de dynamiska modellerna uppvisar en tydligt mindre underprestation antyder dock att medellagerökningen bidrar till att undvika underprestation snarare än att bara uppnå samma servicenivå som originalmodellen fast med högre lager eller att ge en onödig överprestation av servicenivån. Detta tyder på att de dynamiska modellerna faktiskt presterar bättre än originalmodellen.

Vid jämförelse mellan de tre dynamiska modellerna framgår att *traditionell algoritm med justeringar* har bättre värden än *simulering* på samtliga mått. Jämfört med *traditionell algoritm* gäller detsamma med undantaget att *traditionell algoritm med justeringar* ger ett lite högre medellager. Denna skillnad är dock så pass liten att den i ljuset av övriga förbättringar bör kunna försummas. Slutsatsen blir därmed att *traditionell algoritm med justeringar* kan anses vara den bästa av de dynamiska modellerna. Dock är skillnaderna mellan modellernas resultat i många fall ganska små och dataunderlaget är för litet för att man ska kunna dra en säker slutsats.

7.5 Tester med negativa säkerhetstider

Traditionell algoritm presterar över lag bättre då negativa säkerhetstiden tillåts, eftersom överprestation i servicenivån sänks utan att underprestation uppstår i något annat fall. Dock är det bara då maxtiden är lång som detta inträffar, vilket snarast kan anses indikera att maxtiden är felbestämd. *Traditionell algoritm med justeringar* ger visserligen också mindre överprestation i servicenivån då negativa säkerhetstider tillåts, men med ett högre medellager. Detta ger ytterligare belägg för att negativa säkerhetstider inte bör användas.

Eftersom *simulering* som modell arbetar med att söka en säkerhetstid genom att studera historisk data kan modellen ibland finna att det över en viss period är lämpligt med en stor

negativ säkerhetstid. Detta kan bero på att det finns lager att ta av under just den undersökta perioden eller att det sker färre beställningar än vanligt under perioden. Att välja en stor negativ säkerhetstid baserat på föregående period och sedan applicera det på kommande period kan dock bli förödande om lagernivån vid den kommande periodens början är låg. Då uppstår plötsligt stora brister under den kommande perioden och servicenivån sjunker därmed kraftigt. Detta förklarar varför fall med en lång maxtid inte blir bra. Modellen kommer nämligen då och då kunna välja stora negativa säkerhetstider vilket får konsekvensen att vissa perioder får en mycket låg servicenivå vilket ger avtryck i den totala uppmätta servicenivån. Att längre ledtider också blir problematiska i samband med tillåtandet av negativa säkerhetstider är inte konstigt då efterfrågan under ledtiden i dessa fall blir större och därmed ökar även osäkerheten. Kombinerat detta med en negativ säkerhetstid kan bitvis perioder med stora brister uppstå.

Sammanfattningsvis kan konstateras att tillåtande av negativa säkerhetstider kan förbättra instyrningen mot en önskad servicenivå, i vissa fall i stor utsträckning. Behovet av en negativ säkerhetstid indikerar dock att något annat i parametersättningen är fel. En negativ säkerhetstid kan framstå som ologisk att arbeta med eftersom man i princip planerar in att man ska tillåta brister under ett visst antal dagar. Om de dynamiska modellerna kommer fram till att en negativ säkerhetstid ska väljas skulle detta, istället för att en sådan tid faktiskt väljs, kunna utnyttjas för att dynamiskt justera ner maxtiden. Om inte maxtiden ska justeras dynamiskt kan en av modellerna framräknad negativ säkerhetstid istället föranleda en varning om att maxtiden kanske är för hög. Vid integration i programvara kan denna varning presenteras för användaren.

7.6 Dynamisk säkerhetstid ur ett VMI-perspektiv

Matematiskt sett finns det inga hinder att använda de framtagna dynamiska modellerna även i ett VMI-samarbete. Ingående parametrar och beräkningsalgoritmer fungerar likadant oavsett de yttre praktiska förutsättningarna i lagerstyrningen. Ett VMI-samarbete skulle dessutom kunna antas innebära en fördel i implementationen av en dynamisk säkerhetstid. När leverantör och kund har ett nära samarbete och god kännedom om varandras interna förhållanden kan det finnas större acceptans för och förmåga att anpassa sig till de implikationer den ökade dynamiken medför. Det har framlagts att leverantörens flexibilitet ökar vid VMI (se avsnitt 2.6.3 Fördelar med VMI för leverantören), vilket skulle kunna innebära att leverantörens förmåga att hantera dynamiken i systemet är större i ett VMI-sammanhang. Dessutom har det konstaterats att den optimala återfyllnadsfrekvensen är högre och den optimala orderkvantiteten lägre i ett VMI-samarbete (se avsnitt 2.6.2 Fördelar med VMI för köparen). Eftersom detta stämmer väl med hur frekvens och kvantitet påverkas av de dynamiska modellerna är detta ännu ett tecken som tyder på att de dynamiska modellerna lämpar sig väl i VMI-sammanhang.

Den transparens som det innebär att befinna sig i ett VMI-samarbete kan till viss del liknas vid att ha tillgång till en bättre prognos. Det har i denna studie visats att prestandaförhållandet mellan framtagna dynamiska modeller och en statisk säkerhetstid består även vid

användandet av en bättre prognos och därmed kan dynamiska modeller ha en funktion även i ett VMI-sammanhang.

Det har visats att de dynamiska modellerna vid jämn efterfrågan inte kan överträffa en statisk säkerhetstid under förutsättning att denna är satt på ett genomtänkt och optimerat sätt. Det skulle kunna antas att företag som är del i ett VMI-samarbete med hög mognadsgrad har bearbetat sitt flöde i så hög grad att man också har väl bearbetade metoder för att ta fram parametrar. Detta skulle kunna medföra att säkerhetstiden redan optimerats och att en dynamisk modell därmed inte kan ge en ytterligare förbättring. I praktiken är det troligtvis dock så att även företag med ett välutvecklat VMI-samarbete inte optimerat sin säkerhetstid till den grad att det inte är värt att undersöka om en dynamisk modell kan förbättra systemet. Om efterfrågan dessutom är icke-stationär och variationerna inte är exakt kända på förhand så att man manuellt kan justera säkerhetstiden bör en dynamisk uppdatering av säkerhetstiden kunna medföra en förbättring även i ett VMI-samarbete.

8 Slutsats

I detta kapitel presenteras slutsatser och rekommendationer utifrån resultat och analys i denna studie. Dessutom redovisas några uppslag för framtida vidare studier.

Fyra modeller för dynamisk uppdatering av säkerhetstid har utvecklats. Dessa har visats ge en förbättrad prestation jämfört med en statisk säkerhetstid. Bedömningen bygger på att instyrning av servicenivå mot ett önskat värde har förbättrats utan att en oproportionerlig ökning av lagernivåerna har uppstått. Dessutom förutsätts att den statiska säkerhetstiden inte kan sättas till ett optimalt värde. Om däremot en statisk säkerhetstid som motsvarar ett teoretiskt optimalt värde kan sättas, kommer prestationen att överträffa de dynamiska modellerna. En sådan optimal tid antas dock mycket sällan vara möjlig att sätta i praktiken, eftersom efterfrågan är behäftad med stor osäkerhet och oförutsägbarhet.

Ett problem som har uppmärksammats med de dynamiska modellerna i denna studie är att de tidvis ger upphov till väldigt täta beställningar. Detta är något som inträffar då säkerhetstiden räknats fram till en nivå nära maxtiden. Om täta eller oregelbundna inleveranser utgör ett problem i ett visst flöde kan därför inte de dynamiska modellerna rekommenderas för dessa fall. När säkerhetstiden ofta sätts nära maxtiden kan det indikera att maxtiden är för kort och en lösning på problemet kan vara att tillåta en ökning av maxtiden i dessa fall.

Modellen *tidsbaserad avvikelse* är utvecklad ifrån och är nästintill identisk med modellen *traditionell algoritm med justeringar*. Eftersom *tidsbaserad avvikelse* inte ger någon förbättring jämfört med *traditionell algoritm med justeringar* har den refuserats. De tre kvarvarande dynamiska modellerna, *traditionell algoritm*, *traditionell algoritm med justeringar* och *simulering*, bedöms alla vara rekommenderbara. I analysen av modellerna har hänsyn tagits till såväl instyrningen av servicenivån, medelservicenivå, variationen i uppnådd servicenivå över tid och erhållen medellagernivå. Slutsatserna grundar sig på den samlade bedömningen av dessa mått, även om en viss modell kan vara bäst på ett specifikt mått. T.ex. ger *simulering* oftast minst avvikelse från målservicenivån, dock ofta till priset av högre lager.

Då efterfrågan är hög och jämn utan systematiska variationer är *traditionell algoritm* den modell som presterar bäst. Även de andra två modellerna ger godtagbara resultat. För lågrörliga artiklar utan systematiska variationer ger samtliga dynamiska modeller ungefär likvärdiga resultat. Då efterfrågan innehåller systematiska variationer är *traditionell algoritm med justeringar* den modell som ger bäst prestation. Även *simulering* ger godtagbara resultat, medan *traditionell algoritm* bedöms vara olämplig att använda i dessa fall. Den sammantagna slutsatsen är därför att *traditionell algoritm med justeringar* är att rekommendera då man inte känner till ifall systematiska variationer kan uppstå i efterfrågan, eller inte vet exakt när i tiden de kommer att inträffa. Denna slutsats förstärks även av resultaten från verklig data.

Utöver resultaten från den övergripande klassificeringen i tre olika efterfrågegrupper (högrörliga respektive lågrörliga artiklar med jämn efterfrågan, samt efterfrågefall med

systematiska variationer) har inga andra tydliga mönster i modellernas prestation kunnat observeras i olika specifika efterfrågefäll eller olika ledtidsfall.

De dynamiska modellerna antas fungera på motsvarande sätt som i denna studie även i ett VMI-flöde. Ett välutvecklat VMI-samarbete kan å ena sidan antas underlätta anpassningen till en dynamiskt föränderlig säkerhetstid. Å andra sidan kan det välutvecklade flödet eventuellt innebära att kontrollen över flödet redan är så hög att en automatiserad dynamik i säkerhetstiden inte tillför lika mycket som har visats i denna studie. Den generella slutsatsen är ändå att rekommendationerna ovan kan antas vara giltiga även i ett VMI-flöde

8.1 Framtida studier

Det finns flera olika områden inom vilka de dynamiska modellerna som tagits fram i denna studie kan och bör undersökas vidare. Av de förslag på vidare studier som presenteras nedan anser författarna att dynamisk maxtid är det förslag som är högst prioriterat.

Dynamisk maxtid

Den ursprungliga idén med dynamiska parametrar inkluderade både säkerhetstid och maxtid, även om huvudfokus låg på säkerhetstid. I denna studie har endast en dynamisk säkerhetstid undersökts, då det skulle blivit allt för komplext att undersöka dynamik i båda parametrarna samtidigt. Dock anser författarna att resultaten i denna studie tydligt pekar på att även en dynamisk maxtid måste utredas för att en dynamisk modell ska nå sin fulla potential.

Det kan antas att en fullständig dynamik i både maxtid och säkerhetstid kan riskera att leda till en ökad variation och oförutsägbarhet i systemet. Ett förslag för framtida studier är därför att den dynamiska säkerhetstiden ska vara grunden i dynamiken och att maxtiden vid behov ska justeras utifrån säkerhetstiden. I huvudsak föreslås två tillfällen då maxtiden bör justeras:

- Maxtiden bör sänkas då den optimala säkerhetstiden beräknats som negativ. På så sätt undviks överprestation i service och onödig uppbyggnad av lager.
- Maxtiden bör ökas när säkerhetstiden ställts in så nära maxtiden att beställningar genereras i princip varje dag. På så sätt kan servicenivån upprätthållas utan att beställningar sker orealistiskt ofta. Maxtiden behöver i synnerhet ökas i de fall den önskade servicenivån inte uppnås även om säkerhetstiden maximeras under nuvarande förutsättningar.

Maxgräns för säkerhetstiden

Ifall förslaget om dynamisk maxtid refuseras måste de dynamiska modellerna vidareutvecklas på något annat sätt för att undvika allt för täta inleveranser. Förslagsvis sker detta genom att en lämplig maxgräns väljs.

Mer verklighetstroga förutsättningar

I denna studie har en dynamiskt uppdaterad säkerhetstid beaktats i ett system med flera förenklingar gentemot verkligheten. Kan exempelvis en varierande ledtid, införandet av

leveransscheman eller batchstorlekar i ett verkligt system påverka så pass negativt vid användningen av en dynamisk säkerhetstid att de förbättringar som här redovisats inte blir giltiga i praktiken? Detta är områden som behöver utredas i vidare studier.

Reaktion på plötsliga variationer

Vissa av de dynamiska modellerna, i synnerhet *traditionell algoritm med justeringar*, har visat sig reagera väl på variationer i efterfrågan och skapa en följsamhet i lagerstyrningen så att servicenivån kan upprätthållas även när efterfrågan ökar utan förvarning. Dock har de efterfrågefall med systematiska variationer som undersökts innehållit relativt flacka variationer. Det bör utredas vidare ifall *traditionell algoritm med justeringar* även kan hantera mycket plötsliga och branta variationer i efterfrågan och om inte, ifall den kan vidareutvecklas för att klara även sådana fall.

Noggrannare utredning av parametrar

Den ackumulerade differensen och förändringen som används som styrparametrar i *traditionell algoritm med justeringar* har i denna studie inte kunnat utredas fullständigt. Genom att systematiskt ändra och testa olika inställningar för dessa hade förmodligen ännu bättre resultat kunnat uppnås. Även simulering hade säkerligen kunnat optimeras genom att vidare undersöka olika längder på optimeringsintervallet. I denna studie har ett optimeringsintervall på 30 dagar använts, det är dock möjligt att man hade nått bättre resultat om detta intervall minskades betydligt medan själva optimeringssimuleringarna fortfarande omfattade lika många dagar bakåt i tiden. Under optimeringssimuleringarna hade man även kunnat prova ut en lämplig maxtid genom att låta denna variera. Även andra parametrar i båda modellerna hade kunnat undersökas och optimeras ytterligare.

Ytterligare metoder för dynamisk uppdatering av säkerhetstid

Under litteraturstudien påträffades två ytterligare metoder för dynamiska säkerhetslager, som inte har undersökts i denna studie. Huruvida dessa kan användas tillsammans med PipeChains lagerstyrningssystem och leda till förbättrade resultat jämfört med denna studies dynamiska modeller kan utgöra föremål för framtida undersökningar.

Referenser

- Axsäter, S. *Inventory Control*. New York: Springer, 2006.
- Axsäter, S. *Lagerstyrning*. Lund: Studentlitteratur, 1991.
- Bryman, A. *Samhällsvetenskapliga metoder*. Malmö: Liber ekonomi, 2002.
- Claassen, M.J.T., van Weele, A.J. och van Raaij, E.M. Performance outcomes and success factors of vendor managed inventory (VMI). *Supply Chain Management: An International Journal*, vol. 13, nr. 6, 2008, s. 406-414.
- Emigh, J. Vendor-Managed Inventory. *Computerworld*, vol. 33, nr. 34, 23 augusti 1999, s. 52-53.
- Hill, R.M. Stock control and the undershoot of the reorder level. *Journal of the Operational Research Society*, vol. 39, nr. 2, 1988.
- Höst, M., Regnell, B. och Runeson, P. *Att genomföra examensarbete*. Lund: Studentlitteratur, 2006.
- Janssen, F., Heuts, R. och de Kok, T. Theory and methodology on the (R, s, Q) inventory model when demand is modeled as a compound Bernoulli process. *European journal of operational research*, vol. 104, nr. 3, februari 1998.
- Jonsson, P. och Mattsson, S-A. *Materialplaneringsmetoder i svensk industri – Användning och användningssätt*. PLAN, 2005, <http://www.plan.se/Jonsson%20och%20Mattsson%20-%20Materialplaneringsmetoder%20i%20svensk%20industri.pdf>
- Kanet, J. J., Gorman, M.F. och Stöblein, M. Dynamic planned safety stocks in supply networks. *International Journal of Production Research*, vol. 48, nr. 22, 2010, s. 6859–6880.
- Kotzab, H., Seuring, S., Müller, M. och Reiner, G. *Research Methodologies in Supply Chain Management*. Heidelberg: Physica-Verlag, 2005.
- Krupp, J. Safety stock management. *Production and Inventory management journal*, 3rd qtr, 1997, s. 11-18.
- Køhler Gudum, C. och de Kok, T. G. *A Safety Stock Adjustment Procedure to Enable Target Service Levels in Simulation of Generic Inventory Systems*. Köpenhamn: Copenhagen Business School, Department of Management Science and Statistics, Preprint No 1, 2002.
- Körner, S. och Wahlgren, L. *Statistisk dataanalys*. Lund: Studentlitteratur, 2000, s. 208
- Li, Q-Y. och Li, S-J. A dynamic model of the safety stock under VMI. *2009 International Conference on Machine Learning and Cybernetics*, vol. 3, 2009, s. 1304-1308.

- Mattsson, S-A. Bullwhipeffekter och andra konsekvenser av att använda förenklade lagerstyrningsmetoder. *PLAN's forsknings- och tillämpningskonferens*, 2008, s. 117-132.
- Mattsson, S-A. *Effektiv materialstyrning – en handbok för att lyckas*. Helsingborg: Permatron, 2010.
- Mattsson, S-A. *Efterfrågefördelningar för bestämmande av säkerhetslager*. Lund: Next generation innovative logistics (NGIL), 2007.
- Mattsson, S-A. *Materialstyrningsmodeller med hänsyn tagen till överdrag och olika efterfrågefördelningar*. Lund: Lunds universitet, 2007.
- Mattsson, S-A. och Jonsson, P. *Produktionslogistik*. Lund: Studentlitteratur, 2003.
- Mattsson, S-A. *Standardavvikelse för säkerhetslagerberäkningar*. Lund: Next generation innovative logistics (NGIL), 2007.
- Patel, R. och Davidson, B. *Forskningsmetodikens grunder*. Lund: Studentlitteratur, 2003.
- PipeChain. *Information*, 2010. <http://www.pipechain.com/o.o.i.s/689> (Hämtad 2011-03-04)
- PipeChain. *Innovation*, 2010. <http://www.pipechain.com/o.o.i.s/687> (Hämtad 2011-03-04)
- PipeChain. *Insight*, 2010. <http://www.pipechain.com/o.o.i.s/688> (Hämtad 2011-03-04)
- PipeChain. *PipeChain*, 2010. <http://www.pipechain.com/o.o.i.s/542> (Hämtad 2011-03-04)
- PipeChain. *Solutions*, 2010. <http://www.pipechain.com/o.o.i.s/686> (Hämtad 2011-03-04)
- Shao, J., Dong, S., Wu, L., Ma, T. och Wang, D. Research review on bullwhip effect controlling methods in a supply chain under uncertainty environments. *2008 3rd IEEE Conference on Industrial Electronics and Applications*, 2008, s. 1803-1808.
- Silver, E.A. och Peterson, R. *Decision Systems for Inventory Management and Production Planning*. New York: Wiley, 1979.
- Wallén, G. *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur, 1993.
- Wang, C., Ji, S., Shen, J. och Wei, W. Supply chain model in vendor managed inventory. *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, vol. 2, 2008, s. 2110-2113.
- Wu, J. och Zhai, X. The Value of Centralization in a Vendor-Managed Inventory System. *2007 International Conference on Management Science and Engineering*, 2007, s. 704-710.
- Wyland, B., Buxton, K. och Fuqua, B. Simulating the supply chain. *IIE Solutions*, vol. 32, nr. 1, 2000

Zizka, M. The Analytic Approach vs. the Simulation Approach to Determining Safety Stock. *Problems & Perspectives in Management*, nr. 3, 2005, s. 119-128.

Bilaga 1

Genererad efterfrågan utan systematiska förändringar, högrörliga artiklar, målservicenivå 95 procent. Medelvärden för servicenivå och medellager från 30 simuleringar à 6000 dagar.

Lambda = 50, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	99,57%	99,78%	99,89%	99,96%	<i>Servicenivå</i>
<i>Original(optimal)</i>	99,57%	99,78%	99,89%	99,96%	
<i>Original(statisk)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,57%	99,79%	99,89%	99,96%	
<i>Traditionell algoritm med justeringar</i>	99,57%	99,79%	99,89%	99,96%	
<i>Simulering</i>	99,57%	99,79%	99,90%	99,97%	
<i>Original(0)</i>	758,5	1446,0	2822,5	8352,4	
<i>Original(optimal)</i>	758,5	1446,0	2822,5	8352,4	
<i>Original(statisk)</i>	1028,5	1859,5	3236,8	8746,7	
<i>Original(maxtid)</i>	1238,0	2612,6	5361,7	16363,1	
<i>Traditionell algoritm</i>	758,5	1446,4	2827,1	8343,2	
<i>Traditionell algoritm med justeringar</i>	758,5	1446,4	2827,1	8343,2	
<i>Simulering</i>	758,8	1446,8	2826,9	8310,7	

Lambda = 50, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	99,06%	99,56%	99,80%	99,92%	<i>Servicenivå</i>
<i>Original(optimal)</i>	99,06%	99,56%	99,80%	99,92%	
<i>Original(statisk)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,06%	99,56%	99,80%	99,93%	
<i>Traditionell algoritm med justeringar</i>	99,06%	99,56%	99,80%	99,93%	
<i>Simulering</i>	99,11%	99,55%	99,80%	99,94%	
<i>Original(0)</i>	760,8	1449,7	2827,6	8355,2	<i>Medellager</i>
<i>Original(optimal)</i>	760,8	1449,7	2827,6	8355,2	
<i>Original(statisk)</i>	1030,7	2133,9	3925,4	9433,5	
<i>Original(maxtid)</i>	1239,1	2613,7	5362,9	16365,2	
<i>Traditionell algoritm</i>	760,9	1449,2	2827,8	8348,7	
<i>Traditionell algoritm med justeringar</i>	760,8	1449,2	2827,8	8348,7	
<i>Simulering</i>	768,4	1449,8	2829,9	8317,2	

Lambda = 50, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	98,17%	99,13%	99,59%	99,85%	<i>Servicenivå</i>
<i>Original(optimal)</i>	98,17%	99,13%	99,59%	99,85%	
<i>Original(statisk)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,66%	99,15%	99,60%	99,86%	
<i>Traditionell algoritm med justeringar</i>	98,20%	99,12%	99,60%	99,86%	
<i>Simulering</i>	98,31%	99,14%	99,59%	99,86%	
<i>Original(0)</i>	763,8	1454,0	2830,9	8357,5	<i>Medellager</i>
<i>Original(optimal)</i>	763,8	1454,0	2830,9	8357,5	
<i>Original(statisk)</i>	1034,2	2136,7	4202,3	10402,9	
<i>Original(maxtid)</i>	1241,0	2615,5	5364,8	16368,7	
<i>Traditionell algoritm</i>	877,4	1453,9	2832,4	8336,4	
<i>Traditionell algoritm med justeringar</i>	775,0	1453,8	2832,4	8336,4	
<i>Simulering</i>	778,1	1456,8	2839,5	8326,6	

Lambda = 50, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	96,17%	98,19%	99,16%	99,70%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,17%	98,19%	99,16%	99,70%	
<i>Original(statisk)</i>	99,95%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,36%	98,20%	99,16%	99,74%	
<i>Traditionell algoritm med justeringar</i>	96,92%	98,26%	99,16%	99,74%	
<i>Simulering</i>	96,85%	98,18%	99,16%	99,68%	
<i>Original(0)</i>	774,1	1459,9	2842,0	8360,0	
<i>Original(optimal)</i>	774,1	1459,9	2842,0	8360,0	
<i>Original(statisk)</i>	1039,6	2144,5	4212,4	12495,3	
<i>Original(maxtid)</i>	1244,5	2619,2	5368,8	16377,5	
<i>Traditionell algoritm</i>	904,8	1460,6	2840,6	8360,4	
<i>Traditionell algoritm med justeringar</i>	871,8	1505,0	2840,4	8360,4	
<i>Simulering</i>	830,6	1463,7	2857,2	8341,7	

Lambda = 50, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	92,16%	96,41%	98,17%	99,39%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,75%	96,41%	98,17%	99,39%	
<i>Original(statisk)</i>	98,87%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	99,83%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	96,75%	96,44%	98,16%	99,38%	
<i>Traditionell algoritm med justeringar</i>	94,90%	97,04%	98,30%	99,35%	
<i>Simulering</i>	95,75%	97,02%	98,25%	99,34%	
<i>Original(0)</i>	801,8	1489,0	2863,4	8369,0	
<i>Original(optimal)</i>	925,8	1489,0	2863,4	8369,0	
<i>Original(statisk)</i>	1054,3	2159,5	4230,9	12519,5	
<i>Original(maxtid)</i>	1252,6	2627,6	5379,2	16398,3	
<i>Traditionell algoritm</i>	925,8	1490,2	2855,8	8375,1	
<i>Traditionell algoritm med justeringar</i>	975,7	1712,3	3009,1	8447,5	
<i>Simulering</i>	1012,3	1666,1	2916,0	8337,4	

Lambda = 10, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	97,72%	98,88%	99,49%	99,85%	<i>Servicenivå</i>
<i>Original(optimal)</i>	97,72%	98,88%	99,49%	99,85%	
<i>Original(statisk)</i>	99,99%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,86%	98,88%	99,49%	99,85%	
<i>Traditionell algoritm med justeringar</i>	97,77%	98,87%	99,49%	99,85%	
<i>Simulering</i>	97,78%	98,89%	99,48%	99,84%	
<i>Original(0)</i>	153,6	291,3	568,4	1682,2	
<i>Original(optimal)</i>	153,6	291,3	568,4	1682,2	
<i>Original(statisk)</i>	206,7	374,0	651,7	1760,3	
<i>Original(maxtid)</i>	247,2	521,7	1071,5	3286,5	
<i>Traditionell algoritm</i>	180,0	291,3	568,4	1683,2	
<i>Traditionell algoritm med justeringar</i>	155,8	291,8	568,4	1683,2	
<i>Simulering</i>	154,3	291,7	568,5	1675,6	

Lambda = 10, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	95,67%	97,88%	98,94%	99,70%	<i>Servicenivå</i>
<i>Original(optimal)</i>	95,67%	97,88%	98,94%	99,70%	
<i>Original(statisk)</i>	99,88%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,09%	97,92%	98,96%	99,68%	
<i>Traditionell algoritm med justeringar</i>	96,64%	98,00%	98,99%	99,68%	
<i>Simulering</i>	96,40%	98,02%	99,01%	99,68%	
<i>Original(0)</i>	155,5	293,1	571,1	1683,9	
<i>Original(optimal)</i>	155,5	293,1	571,1	1683,9	
<i>Original(statisk)</i>	207,7	429,4	791,0	1904,0	
<i>Original(maxtid)</i>	247,6	522,4	1072,4	3289,5	
<i>Traditionell algoritm</i>	181,1	293,2	570,6	1686,0	
<i>Traditionell algoritm med justeringar</i>	173,7	303,2	573,1	1687,6	
<i>Simulering</i>	165,7	295,8	572,2	1676,4	

Lambda = 10, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	92,44%	96,35%	98,31%	99,42%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,93%	96,35%	98,31%	99,42%	
<i>Original(statisk)</i>	99,02%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	99,85%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	97,18%	97,65%	98,28%	99,41%	
<i>Traditionell algoritm med justeringar</i>	95,80%	97,06%	98,37%	99,43%	
<i>Simulering</i>	95,71%	96,83%	98,31%	99,43%	
<i>Original(0)</i>	159,0	297,0	575,0	1689,3	<i>Medellager</i>
<i>Original(optimal)</i>	183,9	297,0	575,0	1689,3	
<i>Original(statisk)</i>	209,9	431,6	848,2	2100,8	
<i>Original(maxtid)</i>	248,6	523,5	1073,9	3294,7	
<i>Traditionell algoritm</i>	185,7	311,7	574,8	1690,5	
<i>Traditionell algoritm med justeringar</i>	197,8	333,2	590,5	1695,1	
<i>Simulering</i>	194,7	312,8	579,0	1682,2	

Lambda = 10, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	87,00%	93,80%	96,96%	99,02%	<i>Servicenivå</i>
<i>Original(optimal)</i>	95,20%	93,80%	96,96%	99,02%	
<i>Original(statisk)</i>	95,20%	99,87%	100,00%	100,00%	
<i>Original(maxtid)</i>	97,93%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	95,52%	96,53%	96,97%	99,04%	
<i>Traditionell algoritm med justeringar</i>	95,41%	96,20%	97,44%	99,09%	
<i>Simulering</i>	93,90%	95,93%	97,31%	98,99%	
<i>Original(0)</i>	168,8	304,8	580,2	1696,2	<i>Medellager</i>
<i>Original(optimal)</i>	216,0	304,8	580,2	1696,2	
<i>Original(statisk)</i>	216,0	436,2	853,8	2535,2	
<i>Original(maxtid)</i>	252,1	526,4	1078,5	3306,0	
<i>Traditionell algoritm</i>	217,6	329,9	580,6	1693,4	
<i>Traditionell algoritm med justeringar</i>	224,3	375,8	654,2	1765,1	
<i>Simulering</i>	219,7	370,4	620,4	1697,3	

Lambda = 10, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	80,34%	89,35%	94,70%	98,44%	<i>Servicenivå</i>
<i>Original(optimal)</i>	91,16%	94,54%	94,70%	98,44%	
<i>Original(statisk)</i>	87,73%	98,38%	99,99%	100,00%	
<i>Original(maxtid)</i>	91,16%	99,63%	100,00%	100,00%	
<i>Traditionell algoritm</i>	90,75%	94,34%	94,84%	98,44%	
<i>Traditionell algoritm med justeringar</i>	90,46%	94,55%	96,54%	98,65%	
<i>Simulering</i>	90,19%	94,99%	96,52%	98,37%	
<i>Original(0)</i>	196,1	325,8	600,0	1722,4	
<i>Original(optimal)</i>	268,4	375,0	600,0	1722,4	
<i>Original(statisk)</i>	238,6	453,9	871,9	2558,3	
<i>Original(maxtid)</i>	268,4	538,6	1094,7	3332,7	
<i>Traditionell algoritm</i>	262,4	371,4	602,2	1722,4	
<i>Traditionell algoritm med justeringar</i>	262,4	417,8	726,2	1876,0	
<i>Simulering</i>	262,8	439,9	752,3	1798,7	

Lambda = 3, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	93,84%	97,06%	98,57%	99,50%	<i>Servicenivå</i>
<i>Original(optimal)</i>	93,84%	97,06%	98,57%	99,50%	
<i>Original(statisk)</i>	99,49%	99,98%	99,99%	100,00%	
<i>Original(maxtid)</i>	99,95%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	98,04%	97,20%	98,59%	99,50%	
<i>Traditionell algoritm med justeringar</i>	96,23%	97,53%	98,58%	99,50%	
<i>Simulering</i>	95,89%	97,34%	98,59%	99,51%	
<i>Original(0)</i>	49,1	91,4	176,8	517,0	
<i>Original(optimal)</i>	49,1	91,4	176,8	517,0	
<i>Original(statisk)</i>	64,8	116,2	201,1	546,1	
<i>Original(maxtid)</i>	76,0	159,9	328,6	1018,8	
<i>Traditionell algoritm</i>	56,8	92,0	177,2	517,1	
<i>Traditionell algoritm med justeringar</i>	57,4	97,2	180,4	519,8	
<i>Simulering</i>	55,2	92,9	176,7	516,5	

Lambda = 3, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	90,19%	95,37%	97,77%	99,25%	<i>Servicenivå</i>
<i>Original(optimal)</i>	94,77%	95,37%	97,77%	99,25%	
<i>Original(statisk)</i>	97,48%	99,97%	100,00%	100,00%	
<i>Original(maxtid)</i>	99,13%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	97,60%	97,83%	97,78%	99,23%	
<i>Traditionell algoritm med justeringar</i>	96,32%	96,86%	98,08%	99,25%	
<i>Simulering</i>	95,33%	96,56%	97,97%	99,28%	
<i>Original(0)</i>	51,6	94,3	179,4	523,4	
<i>Original(optimal)</i>	59,0	94,3	179,4	523,4	
<i>Original(statisk)</i>	66,9	134,7	245,8	590,6	
<i>Original(maxtid)</i>	77,8	161,7	330,9	1022,9	
<i>Traditionell algoritm</i>	66,9	101,9	179,4	524,0	
<i>Traditionell algoritm med justeringar</i>	66,5	107,3	191,5	531,9	
<i>Simulering</i>	65,1	103,2	183,9	522,4	

Lambda = 3, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	86,04%	93,11%	96,69%	98,92%	<i>Servicenivå</i>
<i>Original(optimal)</i>	95,58%	95,77%	96,69%	98,92%	
<i>Original(statisk)</i>	93,61%	99,60%	100,00%	100,00%	
<i>Original(maxtid)</i>	96,38%	99,95%	100,00%	100,00%	
<i>Traditionell algoritm</i>	95,92%	96,70%	96,69%	98,90%	
<i>Traditionell algoritm med justeringar</i>	95,24%	96,42%	97,56%	99,02%	
<i>Simulering</i>	93,52%	95,85%	97,28%	98,96%	
<i>Original(0)</i>	56,7	98,7	184,6	525,7	
<i>Original(optimal)</i>	77,4	106,2	184,6	525,7	
<i>Original(statisk)</i>	71,0	138,5	267,8	659,3	
<i>Original(maxtid)</i>	81,3	165,1	335,1	1029,8	
<i>Traditionell algoritm</i>	77,7	109,4	184,7	526,2	
<i>Traditionell algoritm med justeringar</i>	76,5	121,0	206,7	552,2	
<i>Simulering</i>	73,7	120,7	199,2	528,3	

Lambda = 3, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	81,85%	90,14%	95,16%	98,44%	<i>Servicenivå</i>
<i>Original(optimal)</i>	91,08%	94,28%	95,16%	98,44%	
<i>Original(statisk)</i>	87,91%	97,80%	99,95%	100,00%	
<i>Original(maxtid)</i>	91,08%	99,14%	100,00%	100,00%	
<i>Traditionell algoritm</i>	91,08%	95,49%	96,48%	98,51%	
<i>Traditionell algoritm med justeringar</i>	90,99%	95,82%	96,92%	98,65%	
<i>Simulering</i>	89,97%	94,94%	96,51%	98,40%	
<i>Original(0)</i>	67,9	109,0	194,8	540,8	
<i>Original(optimal)</i>	90,5	124,2	194,8	540,8	
<i>Original(statisk)</i>	80,8	148,2	278,5	807,3	
<i>Original(maxtid)</i>	90,5	173,3	344,8	1044,8	
<i>Traditionell algoritm</i>	90,5	128,7	202,4	542,0	
<i>Traditionell algoritm med justeringar</i>	90,2	139,2	229,5	592,2	
<i>Simulering</i>	87,9	141,7	234,8	574,9	

Lambda = 3, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	79,77%	87,17%	93,31%	97,88%	<i>Servicenivå</i>
<i>Original(optimal)</i>	85,95%	95,09%	95,60%	97,88%	
<i>Original(statisk)</i>	83,35%	94,40%	99,36%	100,00%	
<i>Original(maxtid)</i>	85,95%	96,45%	99,91%	100,00%	
<i>Traditionell algoritm</i>	85,95%	94,17%	95,63%	97,86%	
<i>Traditionell algoritm med justeringar</i>	85,95%	94,16%	96,37%	98,25%	
<i>Simulering</i>	85,86%	93,51%	95,99%	98,17%	
<i>Original(0)</i>	95,4	135,2	219,6	574,0	
<i>Original(optimal)</i>	114,3	179,0	237,5	574,0	
<i>Original(statisk)</i>	105,3	173,3	303,5	838,9	
<i>Original(maxtid)</i>	114,3	195,2	368,1	1077,2	
<i>Traditionell algoritm</i>	114,3	168,4	235,9	574,8	
<i>Traditionell algoritm med justeringar</i>	114,3	173,9	266,3	629,7	
<i>Simulering</i>	114,0	176,0	276,5	628,6	

Bilaga 2

Genererad efterfrågan utan systematiska förändringar, lågrörliga artiklar, målservicenivå 95 procent. Medelvärden för servicenivå och medellager från 30 simuleringar à 6000 dagar.

Lambda =1/2, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	77,89%	87,77%	93,40%	97,94%	<i>Servicenivå</i>
<i>Original(optimal)</i>	88,40%	94,89%	95,51%	97,94%	
<i>Original(statisk)</i>	85,57%	94,89%	97,81%	99,32%	
<i>Original(maxtid)</i>	88,40%	98,61%	99,99%	100,00%	
<i>Traditionell algoritm</i>	88,05%	93,34%	94,25%	97,96%	
<i>Traditionell algoritm med justeringar</i>	88,27%	95,10%	96,16%	98,14%	
<i>Simulering</i>	87,75%	94,67%	95,96%	98,08%	
<i>Original(0)</i>	11,2	19,3	35,2	103,4	<i>Medellager</i>
<i>Original(optimal)</i>	14,7	23,6	37,2	103,4	
<i>Original(statisk)</i>	13,4	23,6	40,4	109,8	
<i>Original(maxtid)</i>	14,7	30,2	63,0	204,3	
<i>Traditionell algoritm</i>	14,4	22,0	36,2	103,2	
<i>Traditionell algoritm med justeringar</i>	14,6	25,0	42,4	113,3	
<i>Simulering</i>	14,5	25,2	40,7	104,7	

Lambda = 1/2, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
Original(0)	74,80%	85,18%	92,24%	97,51%	Servicenivå
Original(optimal)	83,42%	94,45%	95,34%	97,51%	
Original(statisk)	80,80%	94,45%	99,03%	99,76%	
Original(maxtid)	83,42%	96,67%	99,90%	100,00%	
Traditionell algoritm	83,42%	93,00%	94,27%	97,46%	
Traditionell algoritm med justeringar	83,42%	94,30%	96,20%	97,62%	
Simulering	83,10%	93,70%	95,54%	97,67%	
Original(0)	83,32%	93,40%	95,42%	97,63%	
Original(optimal)	13,0	21,2	37,6	107,0	
Original(statisk)	16,4	27,9	40,9	107,0	
Original(maxtid)	15,1	27,9	50,7	119,1	
Traditionell algoritm	16,4	31,7	64,8	207,1	
Traditionell algoritm med justeringar	16,4	25,7	39,6	106,9	
Simulering	16,4	28,5	45,8	117,8	

Lambda = 1/2, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
Original(0)	73,60%	83,44%	90,81%	97,08%	Servicenivå
Original(optimal)	80,10%	93,70%	94,55%	97,08%	
Original(statisk)	78,02%	91,14%	98,53%	99,88%	
Original(maxtid)	80,10%	93,70%	99,53%	100,00%	
Traditionell algoritm	80,10%	92,56%	94,46%	97,06%	
Traditionell algoritm med justeringar	80,10%	92,57%	95,77%	97,50%	
Simulering	79,92%	91,72%	95,20%	97,62%	
Original(0)	79,94%	91,34%	94,79%	97,67%	
Original(optimal)	16,2	24,5	41,3	112,3	
Original(statisk)	19,4	34,6	46,3	112,3	
Original(maxtid)	18,2	30,9	57,3	139,2	
Traditionell algoritm	19,4	34,6	68,0	211,9	
Traditionell algoritm med justeringar	19,4	31,8	45,6	112,3	
Simulering	19,4	33,1	52,2	124,7	

Lambda = 1/2, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	74,53%	82,78%	90,37%	96,53%	<i>Servicenivå</i>
<i>Original(optimal)</i>	79,12%	91,10%	95,33%	97,38%	
<i>Original(statisk)</i>	77,69%	88,71%	97,02%	99,99%	
<i>Original(maxtid)</i>	79,12%	91,10%	98,66%	100,00%	
<i>Traditionell algoritm</i>	79,12%	91,02%	94,66%	96,79%	
<i>Traditionell algoritm med justeringar</i>	79,12%	90,92%	95,48%	97,47%	
<i>Simulering</i>	79,10%	90,23%	94,61%	97,65%	
<i>Original(0)</i>	79,02%	89,43%	94,42%	97,67%	
<i>Original(optimal)</i>	23,0	32,0	49,6	120,9	
<i>Original(statisk)</i>	26,2	41,4	59,2	125,2	
<i>Original(maxtid)</i>	25,0	37,7	64,9	181,2	
<i>Traditionell algoritm</i>	26,2	41,4	75,3	221,9	
<i>Traditionell algoritm med justeringar</i>	26,2	41,1	57,1	121,2	
<i>Simulering</i>	26,2	41,1	62,7	136,6	

Lambda = 1/2, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	79,16%	84,49%	90,88%	96,93%	<i>Servicenivå</i>
<i>Original(optimal)</i>	81,76%	89,81%	94,91%	96,93%	
<i>Original(statisk)</i>	80,87%	88,04%	95,73%	99,95%	
<i>Original(maxtid)</i>	81,76%	89,81%	97,51%	99,99%	
<i>Traditionell algoritm</i>	81,76%	89,81%	95,18%	97,03%	
<i>Traditionell algoritm med justeringar</i>	81,76%	89,81%	95,41%	97,46%	
<i>Simulering</i>	81,75%	89,28%	94,41%	97,67%	
<i>Original(0)</i>	81,68%	88,83%	94,16%	97,43%	
<i>Original(optimal)</i>	38,3	48,2	68,4	143,5	
<i>Original(statisk)</i>	41,7	57,7	79,5	143,5	
<i>Original(maxtid)</i>	40,5	53,9	82,0	203,0	
<i>Traditionell algoritm</i>	41,7	57,7	92,5	243,1	
<i>Traditionell algoritm med justeringar</i>	41,7	57,7	79,6	143,4	
<i>Simulering</i>	41,7	57,7	82,4	156,5	

Lambda = 1/10, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	64,34%	75,36%	84,82%	95,06%	<i>Servicenivå</i>
<i>Original(optimal)</i>	70,00%	89,56%	95,49%	95,06%	
<i>Original(statisk)</i>	68,53%	83,80%	91,30%	96,92%	
<i>Original(maxtid)</i>	70,00%	89,56%	98,94%	100,00%	
<i>Traditionell algoritm</i>	69,79%	83,62%	88,04%	95,23%	
<i>Traditionell algoritm med justeringar</i>	69,98%	87,34%	92,06%	95,42%	
<i>Simulering</i>	70,00%	88,65%	93,76%	96,46%	
<i>Original(0)</i>	70,00%	88,53%	93,61%	95,82%	<i>Medellager</i>
<i>Original(optimal)</i>	5,0	7,8	13,0	36,5	
<i>Original(statisk)</i>	5,5	10,5	17,0	36,5	
<i>Original(maxtid)</i>	5,4	9,1	14,9	37,4	
<i>Traditionell algoritm</i>	5,5	10,5	21,2	68,1	
<i>Traditionell algoritm med justeringar</i>	5,5	9,1	14,1	37,2	
<i>Simulering</i>	5,5	10,0	16,8	40,4	

Lambda = 1/10, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	72,71%	80,85%	88,22%	96,30%	<i>Servicenivå</i>
<i>Original(optimal)</i>	76,50%	90,71%	95,52%	96,30%	
<i>Original(statisk)</i>	75,49%	88,99%	96,43%	98,96%	
<i>Original(maxtid)</i>	76,50%	90,71%	98,87%	100,00%	
<i>Traditionell algoritm</i>	76,48%	88,89%	92,39%	96,52%	
<i>Traditionell algoritm med justeringar</i>	76,50%	89,66%	93,99%	96,38%	
<i>Simulering</i>	76,49%	89,79%	94,28%	96,83%	
<i>Original(0)</i>	76,49%	89,57%	94,24%	96,44%	<i>Medellager</i>
<i>Original(optimal)</i>	6,5	9,5	14,7	38,7	
<i>Original(statisk)</i>	7,1	12,1	18,3	38,7	
<i>Original(maxtid)</i>	7,0	11,4	19,0	44,4	
<i>Traditionell algoritm</i>	7,1	12,1	22,9	70,1	
<i>Traditionell algoritm med justeringar</i>	7,1	11,5	16,7	39,5	
<i>Simulering</i>	7,1	11,8	18,6	43,7	

Lambda = 1/10, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	80,09%	85,41%	91,07%	96,66%	<i>Servicenivå</i>
<i>Original(optimal)</i>	82,29%	92,30%	96,05%	96,66%	
<i>Original(statisk)</i>	81,74%	91,25%	97,35%	99,73%	
<i>Original(maxtid)</i>	82,29%	92,30%	98,84%	100,00%	
<i>Traditionell algoritm</i>	82,29%	92,12%	95,34%	97,40%	
<i>Traditionell algoritm med justeringar</i>	82,27%	91,86%	95,44%	97,31%	
<i>Simulering</i>	82,23%	91,03%	94,93%	97,34%	
<i>Original(0)</i>	82,17%	90,82%	94,77%	97,34%	<i>Medellager</i>
<i>Original(optimal)</i>	9,2	12,2	18,1	41,4	
<i>Original(statisk)</i>	9,8	14,9	21,2	41,4	
<i>Original(maxtid)</i>	9,6	14,2	22,9	54,1	
<i>Traditionell algoritm</i>	9,8	14,9	25,8	73,6	
<i>Traditionell algoritm med justeringar</i>	9,8	14,7	20,8	43,4	
<i>Simulering</i>	9,8	14,7	21,8	47,9	

Lambda = 1/10, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	88,02%	91,27%	94,73%	98,32%	<i>Servicenivå</i>
<i>Original(optimal)</i>	89,19%	94,86%	95,40%	98,32%	
<i>Original(statisk)</i>	88,91%	94,24%	98,15%	99,97%	
<i>Original(maxtid)</i>	89,19%	94,86%	98,96%	100,00%	
<i>Traditionell algoritm</i>	89,19%	94,86%	98,12%	98,81%	
<i>Traditionell algoritm med justeringar</i>	89,19%	94,71%	97,00%	98,37%	
<i>Simulering</i>	89,03%	93,27%	95,91%	98,59%	
<i>Original(0)</i>	89,00%	93,21%	95,82%	98,62%	<i>Medellager</i>
<i>Original(optimal)</i>	14,6	17,8	24,5	50,6	
<i>Original(statisk)</i>	15,3	20,6	25,0	50,6	
<i>Original(maxtid)</i>	15,1	19,9	28,9	69,0	
<i>Traditionell algoritm</i>	15,3	20,6	31,8	80,8	
<i>Traditionell algoritm med justeringar</i>	15,3	20,6	29,0	53,4	
<i>Simulering</i>	15,3	20,4	27,8	54,9	

Lambda = 1/10, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
Original(0)	94,13%	95,70%	97,14%	98,95%	Servicenivå
Original(optimal)	94,68%	95,70%	97,50%	99,39%	
Original(statisk)	94,56%	96,95%	98,90%	99,99%	
Original(maxtid)	94,68%	97,23%	99,47%	100,00%	
Traditionell algoritm	94,68%	97,23%	99,39%	99,74%	
Traditionell algoritm med justeringar	94,68%	97,17%	98,27%	98,93%	
Simulering	94,38%	96,07%	97,49%	98,92%	
Original(0)	94,34%	96,06%	97,42%	98,86%	Medellager
Original(optimal)	26,4	30,0	37,5	69,2	
Original(statisk)	27,1	30,0	38,0	70,1	
Original(maxtid)	26,9	32,1	41,8	84,2	
Traditionell algoritm	27,1	32,9	44,8	96,0	
Traditionell algoritm med justeringar	27,1	32,9	44,2	73,8	
Simulering	27,1	32,8	40,8	70,9	

Lambda = 1/40, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
Original(0)	50,87%	62,21%	72,59%	87,73%	Servicenivå
Original(optimal)	51,90%	66,85%	84,70%	96,28%	
Original(statisk)	51,57%	65,09%	75,74%	89,45%	
Original(maxtid)	51,90%	66,85%	84,70%	98,70%	
Traditionell algoritm	51,68%	64,53%	73,65%	91,52%	
Traditionell algoritm med justeringar	51,89%	66,49%	78,53%	88,19%	
Simulering	51,90%	66,65%	81,58%	91,06%	
Original(0)	51,90%	66,71%	80,74%	90,06%	Medellager
Original(optimal)	2,6	4,8	9,4	28,8	
Original(statisk)	2,7	5,7	13,5	37,7	
Original(maxtid)	2,7	5,3	10,1	30,3	
Traditionell algoritm	2,7	5,7	13,5	52,3	
Traditionell algoritm med justeringar	2,7	5,2	9,6	32,5	
Simulering	2,7	5,7	11,3	29,3	

Lambda = 1/40, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	60,20%	68,99%	77,24%	90,35%	<i>Servicenivå</i>
<i>Original(optimal)</i>	61,06%	72,80%	87,47%	96,64%	
<i>Original(statisk)</i>	60,84%	72,30%	84,49%	94,15%	
<i>Original(maxtid)</i>	61,06%	72,87%	87,47%	98,81%	
<i>Traditionell algoritm</i>	61,01%	71,50%	79,20%	94,59%	
<i>Traditionell algoritm med justeringar</i>	61,03%	72,30%	83,60%	91,38%	
<i>Simulering</i>	61,06%	72,37%	84,82%	93,05%	
<i>Original(0)</i>	61,06%	72,24%	84,06%	92,47%	
<i>Original(optimal)</i>	4,1	6,6	11,4	32,2	
<i>Original(statisk)</i>	4,2	7,6	15,8	39,3	
<i>Original(maxtid)</i>	4,2	7,4	13,9	35,7	
<i>Traditionell algoritm</i>	4,2	7,6	15,8	54,9	
<i>Traditionell algoritm med justeringar</i>	4,2	7,2	11,9	36,9	
<i>Simulering</i>	4,2	7,5	13,7	33,5	

Lambda = 1/40, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	70,83%	77,42%	83,68%	93,00%	<i>Servicenivå</i>
<i>Original(optimal)</i>	71,40%	80,33%	90,99%	93,60%	
<i>Original(statisk)</i>	71,23%	79,71%	89,86%	97,19%	
<i>Original(maxtid)</i>	71,40%	80,33%	90,99%	98,91%	
<i>Traditionell algoritm</i>	71,38%	79,60%	86,12%	95,78%	
<i>Traditionell algoritm med justeringar</i>	71,36%	79,72%	86,87%	93,88%	
<i>Simulering</i>	71,40%	79,41%	87,31%	93,60%	
<i>Original(0)</i>	71,34%	79,31%	86,97%	93,79%	
<i>Original(optimal)</i>	7,0	9,8	15,2	36,5	
<i>Original(statisk)</i>	7,2	11,0	19,7	37,1	
<i>Original(maxtid)</i>	7,1	10,8	18,5	43,3	
<i>Traditionell algoritm</i>	7,2	11,0	19,7	59,2	
<i>Traditionell algoritm med justeringar</i>	7,2	10,7	16,2	41,7	
<i>Simulering</i>	7,2	10,8	16,9	38,6	

Lambda = 1/40, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	85,53%	89,30%	91,78%	95,69%	<i>Servicenivå</i>
<i>Original(optimal)</i>	85,91%	90,45%	95,18%	98,29%	
<i>Original(statisk)</i>	85,82%	90,26%	94,94%	99,08%	
<i>Original(maxtid)</i>	85,91%	90,53%	95,43%	99,17%	
<i>Traditionell algoritm</i>	85,91%	90,39%	93,29%	98,20%	
<i>Traditionell algoritm med justeringar</i>	85,75%	89,97%	93,19%	96,50%	
<i>Simulering</i>	85,78%	89,85%	93,23%	96,34%	
<i>Original(0)</i>	85,77%	89,94%	93,12%	96,49%	<i>Medellager</i>
<i>Original(optimal)</i>	13,9	17,3	22,8	44,3	
<i>Original(statisk)</i>	14,1	18,5	27,6	52,2	
<i>Original(maxtid)</i>	14,1	18,3	26,6	57,9	
<i>Traditionell algoritm</i>	14,1	18,5	28,0	67,9	
<i>Traditionell algoritm med justeringar</i>	14,1	18,4	24,6	51,1	
<i>Simulering</i>	14,0	18,0	24,4	46,2	

Lambda = 1/40, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	95,89%	96,87%	97,67%	98,63%	<i>Servicenivå</i>
<i>Original(optimal)</i>	95,89%	96,87%	97,67%	98,12%	
<i>Original(statisk)</i>	95,99%	97,02%	98,24%	99,52%	
<i>Original(maxtid)</i>	96,01%	97,11%	98,36%	99,52%	
<i>Traditionell algoritm</i>	96,01%	97,04%	97,88%	99,48%	
<i>Traditionell algoritm med justeringar</i>	95,95%	96,90%	97,78%	98,32%	
<i>Simulering</i>	95,93%	96,90%	97,79%	98,32%	
<i>Original(0)</i>	95,93%	96,88%	97,74%	98,32%	<i>Medellager</i>
<i>Original(optimal)</i>	29,9	33,7	39,8	64,4	
<i>Original(statisk)</i>	29,9	33,7	39,8	62,4	
<i>Original(maxtid)</i>	30,2	34,8	43,7	76,1	
<i>Traditionell algoritm</i>	30,2	35,1	45,0	85,6	
<i>Traditionell algoritm med justeringar</i>	30,2	35,0	42,8	71,0	
<i>Simulering</i>	30,1	34,0	40,5	62,8	

Bilaga 3

Genererad efterfrågan med systematiska förändringar, målservicenivå 95 procent. Medelvärden för servicenivå och medellager från 30 simuleringar à 6000 dagar.

Lång säsong, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	96,28%	98,15%	99,03%	99,64%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,28%	98,15%	99,03%	99,64%	
<i>Original(statisk)</i>	99,90%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	98,69%	98,16%	99,04%	99,62%	
<i>Traditionell algoritm med justeringar</i>	98,05%	98,58%	99,13%	99,62%	
<i>Simulering</i>	97,44%	98,28%	99,05%	99,63%	
<i>Original(0)</i>	307,8	586,9	1165,0	3561,4	<i>Medellager</i>
<i>Original(optimal)</i>	307,8	586,9	1165,0	3561,4	
<i>Original(statisk)</i>	414,3	748,8	1319,4	3819,3	
<i>Original(maxtid)</i>	496,6	1047,8	2153,6	7017,2	
<i>Traditionell algoritm</i>	321,5	587,1	1164,1	3582,4	
<i>Traditionell algoritm med justeringar</i>	325,4	607,6	1169,8	3582,4	
<i>Simulering</i>	346,0	590,8	1164,4	3577,9	

Lång säsong, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	88,98%	94,45%	97,13%	98,92%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,27%	94,45%	97,13%	98,92%	
<i>Original(statisk)</i>	96,27%	99,98%	100,00%	100,00%	
<i>Original(maxtid)</i>	98,99%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	93,45%	94,47%	97,09%	98,95%	
<i>Traditionell algoritm med justeringar</i>	96,29%	97,74%	98,29%	99,00%	
<i>Simulering</i>	93,46%	96,56%	97,61%	98,94%	
<i>Original(0)</i>	318,0	594,5	1178,0	3636,4	<i>Medellager</i>
<i>Original(optimal)</i>	418,0	594,5	1178,0	3636,4	
<i>Original(statisk)</i>	418,0	860,0	1593,0	4290,8	
<i>Original(maxtid)</i>	498,1	1049,1	2157,8	7101,2	
<i>Traditionell algoritm</i>	366,3	594,7	1177,0	3667,2	
<i>Traditionell algoritm med justeringar</i>	351,8	634,1	1232,3	3733,3	
<i>Simulering</i>	396,6	770,0	1299,7	3685,2	

Lång säsong, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	76,88%	86,73%	93,08%	97,50%	<i>Servicenivå</i>
<i>Original(optimal)</i>	84,83%	94,97%	94,73%	97,50%	
<i>Original(statisk)</i>	82,45%	96,58%	99,98%	100,00%	
<i>Original(maxtid)</i>	84,83%	99,53%	100,00%	100,00%	
<i>Traditionell algoritm</i>	80,74%	87,82%	93,09%	97,62%	
<i>Traditionell algoritm med justeringar</i>	82,92%	94,99%	96,84%	98,28%	
<i>Simulering</i>	83,85%	92,86%	95,24%	97,86%	
<i>Original(0)</i>	357,3	617,4	1190,7	3805,9	<i>Medellager</i>
<i>Original(optimal)</i>	520,0	816,2	1235,9	3805,9	
<i>Original(statisk)</i>	447,0	867,8	1710,6	4620,0	
<i>Original(maxtid)</i>	520,0	1052,7	2170,0	7255,4	
<i>Traditionell algoritm</i>	404,7	622,2	1192,9	3836,0	
<i>Traditionell algoritm med justeringar</i>	425,1	684,7	1277,0	4352,8	
<i>Simulering</i>	479,9	883,3	1568,0	4255,5	

Lång säsong, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	67,08%	73,53%	82,68%	94,34%	<i>Servicenivå</i>
<i>Original(optimal)</i>	73,99%	81,96%	94,98%	94,96%	
<i>Original(statisk)</i>	71,89%	79,72%	92,31%	100,00%	
<i>Original(maxtid)</i>	73,99%	81,96%	98,40%	100,00%	
<i>Traditionell algoritm</i>	70,63%	75,01%	82,66%	94,44%	
<i>Traditionell algoritm med justeringar</i>	73,13%	79,49%	90,91%	96,39%	
<i>Simulering</i>	73,90%	81,38%	91,89%	96,34%	
<i>Original(0)</i>	499,0	740,8	1281,0	4064,5	
<i>Original(optimal)</i>	646,1	1130,6	1935,3	4134,5	
<i>Original(statisk)</i>	580,1	963,3	1770,6	5684,1	
<i>Original(maxtid)</i>	646,1	1130,6	2231,6	7602,4	
<i>Traditionell algoritm</i>	551,1	762,6	1282,7	4109,4	
<i>Traditionell algoritm med justeringar</i>	604,3	864,3	1434,5	4176,7	
<i>Simulering</i>	641,6	1088,9	1954,8	6458,2	

Lång säsong, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	56,29%	62,05%	69,08%	88,85%	<i>Servicenivå</i>
<i>Original(optimal)</i>	61,25%	69,69%	78,60%	94,73%	
<i>Original(statisk)</i>	59,35%	67,36%	75,50%	97,36%	
<i>Original(maxtid)</i>	61,25%	69,69%	78,60%	100,00%	
<i>Traditionell algoritm</i>	59,51%	63,22%	69,12%	89,16%	
<i>Traditionell algoritm med justeringar</i>	61,24%	68,40%	75,24%	91,17%	
<i>Simulering</i>	61,25%	69,69%	78,60%	94,39%	
<i>Original(0)</i>	1027,7	1270,6	1823,2	4886,5	
<i>Original(optimal)</i>	1166,7	1658,9	2787,4	5734,7	
<i>Original(statisk)</i>	1102,8	1488,0	2317,3	6335,1	
<i>Original(maxtid)</i>	1166,7	1658,9	2787,4	8392,9	
<i>Traditionell algoritm</i>	1104,3	1301,5	1836,5	4955,3	
<i>Traditionell algoritm med justeringar</i>	1166,6	1554,2	2175,5	6326,7	
<i>Simulering</i>	1166,7	1658,9	2787,4	6812,3	

Kort säsong, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	93,84%	96,82%	98,30%	99,35%	<i>Servicenivå</i>
<i>Original(optimal)</i>	93,84%	96,82%	98,30%	99,35%	
<i>Original(statisk)</i>	98,85%	99,90%	99,96%	99,99%	
<i>Original(maxtid)</i>	99,82%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	96,85%	96,85%	98,30%	99,36%	
<i>Traditionell algoritm med justeringar</i>	97,46%	98,29%	98,69%	99,33%	
<i>Simulering</i>	95,60%	97,58%	98,43%	99,36%	
<i>Original(0)</i>	311,8	597,1	1199,7	4079,7	
<i>Original(optimal)</i>	311,8	597,1	1199,7	4079,7	
<i>Original(statisk)</i>	415,2	754,6	1378,6	4796,8	
<i>Original(maxtid)</i>	496,0	1048,6	2214,9	8629,3	
<i>Traditionell algoritm</i>	336,0	597,7	1195,5	4112,4	
<i>Traditionell algoritm med justeringar</i>	334,6	621,5	1265,4	4056,1	
<i>Simulering</i>	417,9	684,8	1223,4	4079,8	

Kort säsong, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	83,99%	91,12%	95,24%	98,27%	<i>Servicenivå</i>
<i>Original(optimal)</i>	90,83%	94,91%	95,24%	98,27%	
<i>Original(statisk)</i>	89,24%	98,34%	99,96%	99,99%	
<i>Original(maxtid)</i>	90,83%	99,87%	100,00%	100,00%	
<i>Traditionell algoritm</i>	87,45%	92,10%	95,21%	98,30%	
<i>Traditionell algoritm med justeringar</i>	88,99%	96,61%	97,81%	98,41%	
<i>Simulering</i>	89,16%	93,85%	96,51%	98,37%	
<i>Original(0)</i>	335,4	618,5	1235,8	4249,3	
<i>Original(optimal)</i>	507,7	714,8	1235,8	4249,3	
<i>Original(statisk)</i>	430,6	868,2	1656,4	4878,4	
<i>Original(maxtid)</i>	507,7	1056,3	2264,4	8887,7	
<i>Traditionell algoritm</i>	384,8	622,0	1238,6	4386,3	
<i>Traditionell algoritm med justeringar</i>	411,9	674,1	1331,5	4501,6	
<i>Simulering</i>	470,9	900,0	1693,2	4280,3	

Kort säsong, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	75,80%	81,94%	89,01%	96,36%	<i>Servicenivå</i>
<i>Original(optimal)</i>	82,98%	89,60%	94,94%	96,36%	
<i>Original(statisk)</i>	81,28%	87,90%	96,74%	99,90%	
<i>Original(maxtid)</i>	82,98%	89,60%	99,85%	100,00%	
<i>Traditionell algoritm</i>	79,88%	83,19%	89,03%	96,25%	
<i>Traditionell algoritm med justeringar</i>	81,76%	87,58%	94,27%	97,28%	
<i>Simulering</i>	81,98%	88,21%	93,75%	97,18%	
<i>Original(0)</i>	412,1	690,6	1369,8	4737,4	
<i>Original(optimal)</i>	573,5	1115,4	1698,7	4737,4	
<i>Original(statisk)</i>	499,6	927,7	1850,1	5471,4	
<i>Original(maxtid)</i>	573,5	1115,4	2381,8	9329,6	
<i>Traditionell algoritm</i>	470,1	703,4	1357,5	4514,0	
<i>Traditionell algoritm med justeringar</i>	540,5	775,9	1449,1	5449,4	
<i>Simulering</i>	562,3	1092,9	2006,9	6310,3	

Kort säsong, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	66,11%	72,64%	78,93%	91,84%	<i>Servicenivå</i>
<i>Original(optimal)</i>	73,37%	80,58%	87,33%	94,99%	
<i>Original(statisk)</i>	71,05%	78,80%	85,06%	99,18%	
<i>Original(maxtid)</i>	73,37%	80,58%	87,33%	100,00%	
<i>Traditionell algoritm</i>	70,14%	73,84%	78,91%	92,07%	
<i>Traditionell algoritm med justeringar</i>	73,03%	78,55%	84,59%	94,89%	
<i>Simulering</i>	72,70%	80,34%	87,33%	94,71%	
<i>Original(0)</i>	703,3	993,8	1710,1	5274,1	
<i>Original(optimal)</i>	872,4	1455,7	2833,1	5631,7	
<i>Original(statisk)</i>	793,7	1252,5	2266,8	7377,4	
<i>Original(maxtid)</i>	872,4	1455,7	2833,1	10237,9	
<i>Traditionell algoritm</i>	784,0	1041,6	1710,4	5278,9	
<i>Traditionell algoritm med justeringar</i>	868,4	1234,4	1947,1	6294,4	
<i>Simulering</i>	869,5	1452,7	2833,1	6742,9	

Kort säsong, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	57,39%	61,18%	68,43%	84,84%	<i>Servicenivå</i>
<i>Original(optimal)</i>	59,41%	66,83%	78,28%	95,03%	
<i>Original(statisk)</i>	58,53%	64,56%	74,18%	95,53%	
<i>Original(maxtid)</i>	59,41%	66,83%	78,28%	100,00%	
<i>Traditionell algoritm</i>	58,55%	61,72%	68,51%	84,09%	
<i>Traditionell algoritm med justeringar</i>	59,36%	66,23%	75,37%	94,71%	
<i>Simulering</i>	59,41%	66,83%	78,28%	94,63%	
<i>Original(0)</i>	2008,8	2369,6	3200,2	6686,7	<i>Medellager</i>
<i>Original(optimal)</i>	2230,8	2937,8	4530,1	9285,7	
<i>Original(statisk)</i>	2128,7	2692,3	3861,9	9454,4	
<i>Original(maxtid)</i>	2230,8	2937,8	4530,1	12127,0	
<i>Traditionell algoritm</i>	2135,6	2436,1	3233,7	6477,1	
<i>Traditionell algoritm med justeringar</i>	2230,5	2851,6	3968,6	10709,7	
<i>Simulering</i>	2230,8	2937,8	4530,1	10334,1	

Veckosäsong, LT = 2	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	98,98%	97,97%	98,36%	99,50%	<i>Servicenivå</i>
<i>Original(optimal)</i>	98,98%	97,97%	98,36%	99,50%	
<i>Original(statisk)</i>	99,99%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,99%	99,77%	98,38%	99,50%	
<i>Traditionell algoritm med justeringar</i>	99,42%	98,30%	98,44%	99,50%	
<i>Simulering</i>	99,11%	98,07%	98,38%	99,48%	
<i>Original(0)</i>	537,3	864,0	1556,6	4532,1	<i>Medellager</i>
<i>Original(optimal)</i>	537,3	864,0	1556,6	4532,1	
<i>Original(statisk)</i>	608,9	1012,2	1771,8	4784,4	
<i>Original(maxtid)</i>	657,0	1394,3	2893,7	8972,1	
<i>Traditionell algoritm</i>	608,9	933,4	1557,6	4535,0	
<i>Traditionell algoritm med justeringar</i>	545,1	896,9	1586,9	4535,0	
<i>Simulering</i>	540,3	868,2	1556,1	4515,2	

Veckosäsöng, LT = 5	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	96,46%	97,52%	98,56%	99,48%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,46%	97,52%	98,56%	99,48%	
<i>Original(statisk)</i>	99,73%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	100,00%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	100,00%	99,40%	98,57%	99,43%	
<i>Traditionell algoritm med justeringar</i>	99,73%	97,83%	98,56%	99,43%	
<i>Simulering</i>	96,93%	97,65%	98,53%	99,45%	
<i>Original(0)</i>	521,6	846,6	1565,6	4554,0	
<i>Original(optimal)</i>	521,6	846,6	1565,6	4554,0	
<i>Original(statisk)</i>	621,5	1212,3	2167,2	5197,4	
<i>Original(maxtid)</i>	662,5	1406,3	2911,0	8994,8	
<i>Traditionell algoritm</i>	642,9	951,8	1566,8	4560,7	
<i>Traditionell algoritm med justeringar</i>	621,5	874,2	1582,7	4560,7	
<i>Simulering</i>	540,8	856,5	1571,3	4559,1	

Veckosäsöng, LT = 10	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	93,69%	96,33%	97,98%	99,33%	<i>Servicenivå</i>
<i>Original(optimal)</i>	93,69%	96,33%	97,98%	99,33%	
<i>Original(statisk)</i>	99,22%	100,00%	100,00%	100,00%	
<i>Original(maxtid)</i>	99,97%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,97%	99,77%	99,49%	99,32%	
<i>Traditionell algoritm med justeringar</i>	99,93%	98,85%	98,10%	99,31%	
<i>Simulering</i>	96,06%	97,01%	98,09%	99,35%	
<i>Original(0)</i>	491,8	850,9	1599,8	4622,9	
<i>Original(optimal)</i>	491,8	850,9	1599,8	4622,9	
<i>Original(statisk)</i>	646,8	1236,8	2379,4	5795,2	
<i>Original(maxtid)</i>	680,5	1431,0	2941,1	9032,8	
<i>Traditionell algoritm</i>	680,5	1020,1	1701,3	4611,4	
<i>Traditionell algoritm med justeringar</i>	670,4	990,6	1637,1	4624,9	
<i>Simulering</i>	567,0	902,5	1611,2	4598,0	

Veckosäsöng, LT = 20	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	92,11%	95,37%	97,48%	99,15%	<i>Servicenivå</i>
<i>Original(optimal)</i>	96,85%	95,37%	97,48%	99,15%	
<i>Original(statisk)</i>	98,19%	99,99%	100,00%	100,00%	
<i>Original(maxtid)</i>	99,53%	100,00%	100,00%	100,00%	
<i>Traditionell algoritm</i>	99,53%	99,97%	99,66%	99,14%	
<i>Traditionell algoritm med justeringar</i>	99,53%	99,78%	98,06%	99,23%	
<i>Simulering</i>	95,62%	96,61%	97,69%	99,19%	
<i>Original(0)</i>	572,8	931,9	1685,1	4718,8	<i>Medellager</i>
<i>Original(optimal)</i>	687,1	931,9	1685,1	4718,8	
<i>Original(statisk)</i>	709,1	1341,8	2476,0	7070,9	
<i>Original(maxtid)</i>	733,5	1489,1	3006,4	9109,3	
<i>Traditionell algoritm</i>	733,5	1203,6	1837,9	4714,4	
<i>Traditionell algoritm med justeringar</i>	733,5	1126,6	1788,7	4782,2	
<i>Simulering</i>	653,8	1032,9	1727,4	4717,1	

Veckosäsöng, LT = 40	MT = 5	MT = 10	MT = 20	MT = 60	
<i>Original(0)</i>	91,28%	94,84%	97,16%	99,10%	<i>Servicenivå</i>
<i>Original(optimal)</i>	94,91%	94,84%	97,16%	99,10%	
<i>Original(statisk)</i>	96,44%	99,88%	100,00%	100,00%	
<i>Original(maxtid)</i>	97,50%	99,99%	100,00%	100,00%	
<i>Traditionell algoritm</i>	97,50%	99,99%	99,80%	99,07%	
<i>Traditionell algoritm med justeringar</i>	97,50%	99,90%	99,32%	99,20%	
<i>Simulering</i>	95,08%	96,46%	97,60%	99,08%	
<i>Original(0)</i>	748,3	1108,8	1868,7	4955,9	<i>Medellager</i>
<i>Original(optimal)</i>	821,7	1108,8	1868,7	4955,9	
<i>Original(statisk)</i>	849,8	1500,5	2633,8	7261,1	
<i>Original(maxtid)</i>	865,4	1624,5	3149,5	9267,3	
<i>Traditionell algoritm</i>	865,4	1612,2	2123,0	4953,0	
<i>Traditionell algoritm med justeringar</i>	865,4	1505,3	2044,4	5078,8	
<i>Simulering</i>	820,4	1260,1	2033,1	5029,9	

Bilaga 4

Servicenivåns avvikelse från riktvärdet för respektive efterfrågestruktur samt för respektive ledtid. För efterfrågan utan systematiska variationer betecknas efterfrågestrukturen med lambda som motsvarar ankomstintensiteten per dag. För efterfrågan med systematiska variationer betecknas efterfrågestrukturen med namnet för den specifika säsongsvariationen.

Tabell 1 Endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 95 %.

Modell	Lambda = 50	Lambda = 10	Lambda = 3
Original (optimal)	0,017	0,041	0,041
Original (statisk)	0,223	0,441	0,518
Traditionell algoritm	0,066	0,125	0,143
Traditionell algoritm med justeringar	0,018	0,079	0,130
Simulering	0,024	0,072	0,113

Tabell 2 Endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 95 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,012	0,002	0,033	0,021	0,032
Original (statisk)	0,142	0,194	0,236	0,281	0,330
Traditionell algoritm	0,053	0,086	0,077	0,072	0,045
Traditionell algoritm med justeringar	0,018	0,043	0,043	0,059	0,064
Simulering	0,013	0,028	0,043	0,057	0,069

Tabell 3 Endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 98 %.

Modell	Lambda = 50	Lambda = 10	Lambda = 3
Original (optimal)	0,030	0,047	0,039
Original (statisk)	0,162	0,292	0,326
Traditionell algoritm	0,096	0,116	0,100
Traditionell algoritm med justeringar	0,043	0,066	0,068
Simulering	0,034	0,075	0,092

Tabell 4 Endast slumpmässiga variationer, högrörliga artiklar, målservicenivå 98 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,013	0,023	0,022	0,033	0,024
Original (statisk)	0,098	0,123	0,161	0,187	0,212
Traditionell algoritm	0,069	0,069	0,074	0,049	0,051
Traditionell algoritm med justeringar	0,020	0,023	0,026	0,038	0,071
Simulering	0,022	0,031	0,040	0,054	0,053

Tabell 5 Endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 95 %.

Modell	Lambda = 50	Lambda = 10	Lambda = 3
Original (optimal)	0,000	0,017	0,000
Original (statisk)	0,028	0,150	0,001
Traditionell algoritm	0,015	0,058	0,001
Traditionell algoritm med justeringar	0,004	0,040	0,000
Simulering	0,003	0,034	0,001

Tabell 6 Endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 95 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,017	0,000	0,000	0,000	0,000
Original (statisk)	0,105	0,055	0,017	0,002	0,000
Traditionell algoritm	0,026	0,033	0,013	0,002	0,000
Traditionell algoritm med justeringar	0,025	0,016	0,004	0,000	0,000
Simulering	0,022	0,014	0,002	0,000	0,000

Tabell 7 Endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 98 %.

Modell	Lambda = 50	Lambda = 10	Lambda = 3
Original (optimal)	0,000	0,017	0,000
Original (statisk)	0,027	0,095	0,000
Traditionell algoritm	0,024	0,067	0,000
Traditionell algoritm med justeringar	0,009	0,058	0,000
Simulering	0,005	0,037	0,000

Tabell 8 Endast slumpmässiga variationer, lågrörliga artiklar, målservicenivå 98 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,014	0,003	0,000	0,000	0,000
Original (statisk)	0,070	0,038	0,013	0,001	0,000
Traditionell algoritm	0,043	0,032	0,014	0,001	0,000
Traditionell algoritm med justeringar	0,050	0,012	0,005	0,000	0,000
Simulering	0,024	0,015	0,003	0,000	0,000

Tabell 9 Systematiska variationer, målservicenivå 95 %.

Modell	Lång säsong	Kort säsong	Veckosäsong
Original (optimal)	0,024	0,014	0,034
Original (statisk)	0,502	0,476	0,447
Traditionell algoritm	0,646	0,718	0,397
Traditionell algoritm med justeringar	0,294	0,245	0,313
Simulering	0,144	0,117	0,069

Tabell 10 Systematiska variationer, lång säsong, målservicenivå 95 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,000	0,018	0,003	0,001	0,003
Original (statisk)	0,068	0,102	0,114	0,120	0,097
Traditionell algoritm	0,024	0,021	0,133	0,232	0,235
Traditionell algoritm med justeringar	0,023	0,053	0,045	0,088	0,085
Simulering	0,013	0,036	0,037	0,051	0,006

Tabell 11 Systematiska variationer, kort säsong, målservicenivå 95 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,012	0,001	0,001	0,000	0,000
Original (statisk)	0,092	0,114	0,087	0,105	0,078
Traditionell algoritm	0,019	0,063	0,156	0,213	0,266
Traditionell algoritm med justeringar	0,043	0,062	0,049	0,052	0,039
Simulering	0,015	0,042	0,045	0,012	0,004

Tabell 12 Systematiska variationer, veckosäsong, målservicenivå 95 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,000	0,000	0,013	0,019	0,002
Original (statisk)	0,052	0,077	0,106	0,112	0,101
Traditionell algoritm	0,028	0,055	0,099	0,113	0,102
Traditionell algoritm med justeringar	0,008	0,036	0,076	0,096	0,096
Simulering	0,003	0,007	0,019	0,021	0,020

Tabell 13 Systematiska variationer, målservicenivå 98 %.

Modell	Lång säsong	Kort säsong	Veckosäsong
Original (optimal)	0,034	0,027	0,038
Original (statisk)	0,397	0,345	0,276
Traditionell algoritm	0,728	0,763	0,265
Traditionell algoritm med justeringar	0,318	0,248	0,214
Simulering	0,148	0,131	0,081

Tabell 14 Systematiska variationer, lång säsong, målservicenivå 98 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,012	0,019	0,002	0,001	0,000
Original (statisk)	0,051	0,068	0,078	0,120	0,080
Traditionell algoritm	0,024	0,058	0,165	0,253	0,228
Traditionell algoritm med justeringar	0,023	0,026	0,040	0,115	0,114
Simulering	0,008	0,031	0,052	0,046	0,011

Tabell 15 Systematiska variationer, kort säsong, målservicenivå 98 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,015	0,006	0,003	0,002	0,001
Original (statisk)	0,051	0,056	0,066	0,075	0,097
Traditionell algoritm	0,014	0,098	0,173	0,209	0,270
Traditionell algoritm med justeringar	0,024	0,026	0,052	0,074	0,071
Simulering	0,013	0,041	0,052	0,013	0,011

Tabell 16 Systematiska variationer, veckosäsong, målservicenivå 98 %.

Modell	LT = 2	LT = 5	LT = 10	LT = 20	LT = 40
Original (optimal)	0,000	0,006	0,014	0,009	0,009
Original (statisk)	0,051	0,057	0,059	0,050	0,058
Traditionell algoritm	0,044	0,054	0,060	0,060	0,047
Traditionell algoritm med justeringar	0,027	0,044	0,050	0,054	0,041
Simulering	0,017	0,015	0,016	0,016	0,017