THE UNIVERSITY OF CALGARY

Dynamically Set Lead Times In MRP Environments

by

Roger Mattar

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL ENGINEERING

CALGARY, ALBERTA

JULY, 1997

0-612-24720-1

Abstract

Material Requirements Planning logic is widely used in the manufacture of discrete goods in batch production environments. MRP is often criticised for promoting excessive work-in-progress inventory. This is related to the use of static planned lead times, which are set to accommodate worst case scenarios.

In this research, it is proposed that lead times are set dynamically taking current shop loads and batch sizes into account. The updated lead times are used to set more valid release and due dates for jobs.

A literature review is provided. An MRP system is interfaced with the simulation of a production environment. The performance of the system running under static lead times is compared to the same when dynamic lead times are used. Results indicate that when shop load fluctuates, dynamic lead times can improve delivery performance. However, at very high loads, the relationship used to set lead times is not responsive enough.

Acknowledgements

Many thanks go to

Dr. Van Enns, my supervisor, for his patience, guidance, the many tours, and for being a friend.

Dr. Gu and Dr. Balakrishnan of my examining committee for their effort in reviewing and examining this thesis.

My family; Tony, Samira and Johnny. Thanks for the food, advice, support, encouragement and love.

My extended family; Tante Hoda, Boss Zouzou, Louis, Sheila, Jamil, Nada, Sarah, Jad, Carla, Pierre, Monique, and Andrew.

The Department of Mechanical Engineering, the University Research Grants Committee and the Natural Sciences and Engineering Research Council for their generous financial support of this research.

Thank you also to all those who helped me in many ways, particularly Lynn Banach.

Table of Contents

List of Abbreviations

α	Exponential smoothing constant
AFM	Adaptive Forecasting Model
ANOVA	Analysis of Variance
BOM	Bill of Materials
C_i	Completion date of order i
CDMRP	Continuous Delivery Material Requirements Planning
CI	Confidence Interval
CNC	Computer Numerically Controlled
CRP	Capacity Requirements Planning
CV	Coefficient of Variation
d_i	Due date of order i
DBR	Drum-Buffer-Rope scheduling technique
EDD	Earliest Due Date dispatching rule
ESOFT	Exponentially Smoothed Operation Flowtime
FCS	Finite Capacity Scheduling
FMS	Flexible Manufacturing System
FT	Flowtime
GR	Gross Requirements
H_A	Alternate Hypothesis
H_0	Null Hypothesis
JIT	Just-In-Time

L_i	Lateness of order i
LFL	Lot For Lot batch sizing technique
LS	Lot Size
LT	Lead Time
MPS	Master Production Schedule
MRP	Material Requirements Planning
μ	Mean
OFT	Operation Flowtime
OLT	Operation Lead Time
PT	Percent Tardy
rand	Random number between 0 and 1
RCCP	Rough Cut Capacity Planning
σ	Standard deviation
SR	Scheduled Receipts
T_i	Mean tardiness of order i
TOC	Theory of Constraints
TWK	Total Work Content
TWKCP	Total Work Content on Critical Path
VBA	Visual Basic for Applications
WINS	Work in Systems
WIP	Work-in-Progress inventory

Chapter One

INTRODUCTION

Material Requirements Planning (MRP) systems are used in the manufacture of discrete goods in batch production environments. Developed in the 1960's, MRP systems are still in wide use today despite some well recognised weaknesses. The introduction of other production planning and control systems has not dampened enthusiasm for MRP: many companies look for ways to adapt the MRP approach or enhance their existing systems (Vollmann et al., 1992).

It is generally recognised that production environments operating under MRP control tend to have high levels of work-in-progress (WIP) inventory and correspondingly long manufacturing lead times. These not only worsen a company's financial position but make the shop floor more congested and difficult to coordinate. It is claimed that this poor performance is connected to the way manufacturing planned lead times are set. Over the years, this has led to suggestions that lead times be set dynamically (e.g. Hoyt, 1978).

The objective of this research is to test the hypothesis that dynamically setting manufacturing planned lead times improves the performance of MRP-controlled production environments. Planned lead times for purchased products continue to be statically set. Dynamic lead times are expected to adjust to shop conditions and maintain the validity of the planned lead times.

The remainder of this chapter is an introduction to MRP systems. A review of the literature is provided in chapter two. The production environment assumed is defined in chapter three and the fourth chapter describes development of the software. An experimental plan is discussed and outlined in chapter five. In chapter six the results are presented and analysis of statistical tests is undertaken. Conclusions drawn from this research are offered in chapter seven.

Some terms used extensively throughout the thesis are defined below:

- Lead time: the time allowed for an order to progress through the shop floor (completion time - release time = lead time). Lead time is planned, and is also referred to as *flow allowance* or *planned lead time.*
- Flowtime: the actual time that it takes for an order to progress through the shop floor (from time of release

into shop until time of completion). Flowtime is not planned, and is referred to in some of the literature as *actual lead times.*

- Lead time and flowtime can be calculated or measured for a single operation. In such cases, we refer to the lead time per operation and flowtime per operation.

1.1 Introduction to MRP

The development of MRP did not become possible until the advent of commercial computers in the mid-1950's. As the power of computing became recognised, existing inventory control systems (which were based on assumptions inappropriate for manufacturing environments) began to be questioned. An example of assumptions used in these systems (e.g re-order point, stock replenishment) is the idea of independent demand. Proponents of MRP argued that demand for components used in production was not independent but depended on the demand for the end item being produced. MRP was developed to exactly calculate the dependent demand for components. Demand is defined as dependent when it derives from the demand for another product. Dependent demand can be calculated and need (should) not be forecast. Demand is defined as independent when it is not directly related to demand for any other items.

Independent demand has to be forecast. The demand for automobiles from a manufacturer may be classified as independent. Demand by the manufacturer for stereo cassette players for the same automobiles should be classified as dependent demand. The purchase of a stereo cassette player, by a customer, from the manufacturer, as a replacement, however, is classified as independent demand.

1.1.1 Material Requirement Logic

In actual production environments, there are many components which are common to several end items. It was recognised that demands for the same components by multiple parent parts or assemblies should be jointly considered. Low-level coding was developed in response to some of these concerns. All bills of materials are analysed and the lowest level in the product trees at which a component appears is identified. This low-level code is added to component records. When determining gross requirements for all components, MRP processes all component records by level, highest first. The processing of a record is therefore delayed until all requirements for the component from higher levels have been established (Orlicky, 1975).

In many cases production or procurement of components involves expensive setups or long delivery times. To offset the effect of costly setups and long delivery times, different lot sizing rules were developed and used. Lot sizes would be set to arrive at a compromise between low costs and short batch times. Such lot sizing techniques include economic order quantity, least total cost, and lot for lot. Full explanations of these and other lot sizing rules can be found in Melnyk and Piper (1985), Lunn and Neff (1982) and Fogarty et al (1991). Some additional principles of MRP systems are outlined in the next sections. These are taken from Orlicky (1975).

1.1.2 Time Phasing Logic

The time intervals allowed to manufacture a component (or for it to be delivered) are called planned lead times. Lead times are made up from estimates including queueing, setup, processing and moving times. They are used to calculate planned lead time offsets for each component.

For example, an order for a desk is to be shipped at the end of week 19, and the assembly lead time is one week. The components for the desk (legs, desktop) must then be ready by the end of week 18. If it takes two weeks for delivery of the legs, then the order must be placed two weeks prior to the end of week 18 (i.e by the end of week 16). The desktop can be manufactured in only one week. MRP will therefore release an order authorising production to commence at the end of week 17. An important feature here is the use of backward scheduling. This, together with the time phasing, provides coordination of parts going into assembly. In the above example, the legs are started at a different time to the desktop such that they both arrive at the same time for assembly. This coordination reduces inventory (& hence costs) and improves work flow.

1.1.3 MRP Prerequisite Information

The following points summarise the essential pre-requisites for using MRP:

- existence of a master production schedule (MPS). The MPS tells the MRP how much and when to produce what end items
- each inventory item uniquely identified by part number
- existence of a bill of materials (BOM). The BOM identifies each manufactured item's components. BOM structure often reflects production procedure
- availability of inventory records (may include part

number, batch size, inventory status, product supplier, lead time) for all items

- availability of inventory status and planning factors (e.g lot sizes)
- integrity of data in files

1.1.4 MRP Assumptions

Below are listed several important assumptions for operating MRP systems:

- planned lead times are specified for all inventory items
- every inventory item goes into and out of stock (even if only momentarily)
- all components of an assembly are needed at the time of assembly order release
- discrete disbursement and usage of component materials
- process independence of manufactured items. This means an order for any item may be started and finished and not be dependent on any other order for purposes of completion

1.1.5 MRP Applicability

The applicability of using an MRP system to generate component release plans can be determined as follows:

- end item requirements are stated in the MPS. Gross component requirements and their timing are derived by the MRP from this MPS and the BOM's for the end items
- discrete manufacturing process
- any level of product complexity
- any discrete item subject to dependent demand

1.2 The MRP System

This section looks at the objectives of MRP systems. Inputs and outputs are listed. The MRP planning and control system is illustrated.

1.2.1 Objectives

The objective of all MRP systems is to determine the appropriate amount and timing of gross and net material requirements. This information is used to generate correct action pertaining to purchasing and production. Actions are either new ones or revisions of old ones. Revisions will frequently modify information on order quantity, release and due dates.

Net requirements are always related to time and are covered by planned or open orders. Planned orders are one of the outputs of an MRP system. They indicate a time in the future when an order should be placed with a supplier or a work order released to the shop floor. Planned orders become open orders (also known as scheduled receipts) when the planner releases the order to the shop floor or to a supplier.

1.2.2 Inputs and Outputs

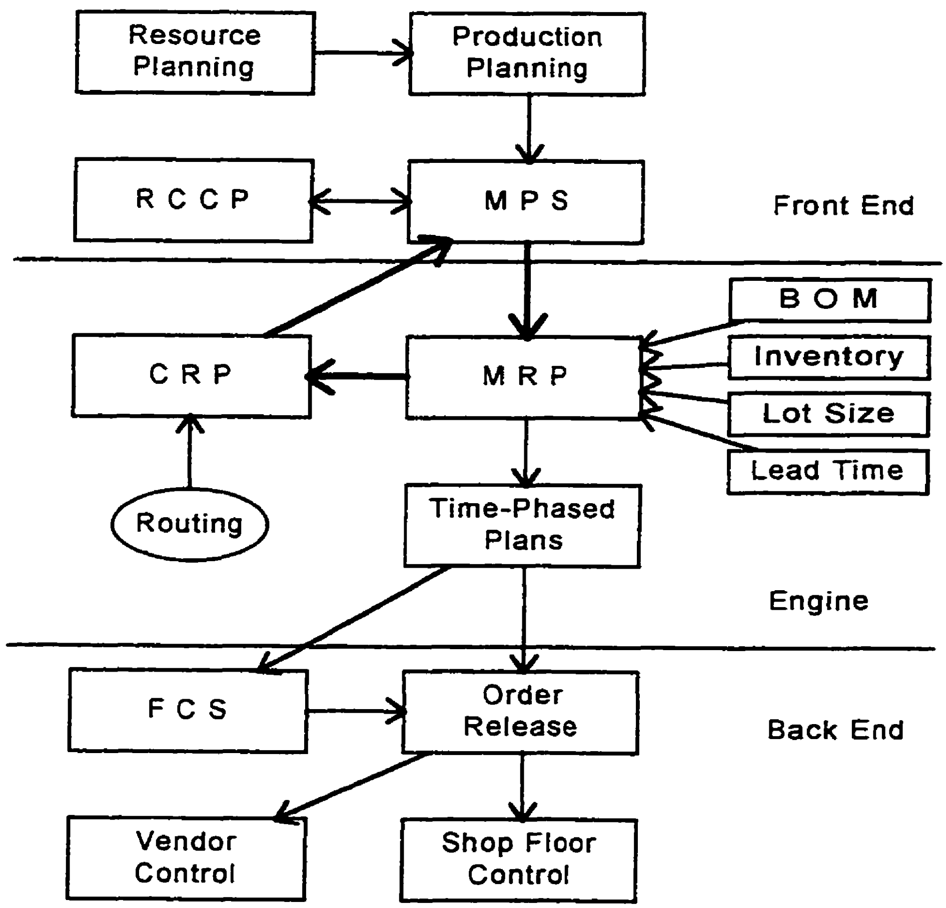
There are five main inputs into an MRP system:

- master production schedule
- inventory records
- lot sizing rules
- bills of materials
- planned lead time data

The outputs of an MRP system include:

- component order release and rescheduling notices
- order cancellation notices
- planned orders scheduled for release into the future
- item status analysis backup data
- inventory forecasts

Figure 1.1 illustrates in more detail how MRP functions within the overall framework of a production planning system. The front end in figure 1.1 represents the longer term planning portion. Resource and production planning take a long term view. They determine company needs for the foreseeable future.



Adapted from Vollmann, Berry, Whybark (1992) and Enns (1995a)

Figure 1.1 MRP within the production planning and control hierarchy

The MPS can then be set for an extended period using guidelines set by marketing and production management. The MPS is checked for feasibility using a rough cut capacity planning (RCCP) tool. The engine portion represents the MRP system and its associated inputs and outputs. The CRP module is described in section 2.3. The routing file for a part describes how to produce that part (which machine, tooling, setup times, etc). Time-phased plans take the form of orders, their release dates and due dates. The back end in figure 1.1 depicts the day-to-day shop floor and vendor control system. A finite capacity scheduling (FCS) module may be run, using the plans generated by MRP. FCS is discussed in section 2.3.

1.2.3 Record Processing

The basic MRP record is displayed in table 1.1 This record displays the following:

- gross requirements (GR): anticipated future usage for the item during each time bucket
- scheduled receipts (SR): existing replenishment orders for the item due in at the beginning of each time bucket
- projected on hand: current inventory in period *i* and future inventory status for the item at the end of each time bucket
- planned order release: planned replenishment orders for item at the beginning of each time bucket
- net requirements: Max(GR-SR-On Hand Inventory , 0)

LT=		P	1	2	3	4	5	6	7	8
	Gross Requirements									
	Scheduled Receipts									
LS=	Projected On Hand									
	Net Requirements									
	Planned Order Releases									

Table 1.1 The Basic MRP Record

A simple example is shown in table 1.2 below. A product (P101) is assembled from several components, including two units of component C102. A customer orders 120 units of P101 to be delivered in period 3. The planned order for P101 is released 2 time buckets in advance (period 1). This is called lead time offsetting. The planned orders for the parent (P101) become the gross requirements for the component (C102). Records for other components that would be required for assembly of P101 would be filled in exactly the same way. Thus the components are coordinated to arrive together for assembly. For more complicated product structures, the techniques used are exactly the same. The difficulty lies in coordinating several large end items with common parts simultaneously.

In actual production environments, most end items involve several assembly stages. Part numbers will change several times, as different stages of assembly are reached. At each of these stages, a work order may be required to allow a part to continue through the shop.

LT=	**Part: P101**	**PD**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
2	Gross Requirements		25	15	120		70		15	
	Scheduled Receipts									
LS=	Projected On Hand	50	25	10	0	0	0	0	0	0
1	Net Requirements				110		70		15	
	Planned Order Releases		110		70		15			

LT=	**Part: C102**	**PD**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	Gross Requirements		220		140		30			
	Scheduled Receipts		40							
LS=	Projected On Hand	225	45	45	25	25	35	35	35	35
40	Net Requirements				95		25			
	Planned Order Releases			120		40				

Table 1.2 A Simple Example of MRP Records

In this research, the product structures used are simple enough and involve no commonality of parts. They are described further in chapter three. Therefore once an order is released to the shop floor, assembly is assumed to be authorised at the time all components become available. Although MRP systems do not work in this way in real life, this method will not have a significant impact until product complexity increases substantially.

1.3 Alternatives to MRP

Production planning and control systems can be divided into three broad classes: 'push' systems, which include MRP, 'pull' systems like just-in-time (JIT), and those based on the theory of constraints (TOC), such as drum-buffer-rope (DBR) scheduling. It could be argued that DBR is a hybrid 'push'-'pull' system. Browne et al. (1996) offer a good comparison of these three systems.

Pull systems maintain a constant level of WIP on the shop floor. An order cannot be released until another has finished. Push systems release orders as necessary in order to have them completed by their due date. The level of WIP fluctuates.

1.3.1 Pull Systems

The Kanban production control system has received significant attention recently. A great deal of benefit has been gained in many production environments from the emergence of JIT systems using Kanban production control. The improvements that needed to be implemented to make Kanban feasible, such as setup time, batch size and process variability reduction, can be beneficial to any company. Due to the very small buffer size between stations, however, Kanban requires a very stable production environment, where demand can be smoothed out. For example, Toyota Motor Corporation's production plan covers one year and is updated monthly (Vollmann et al., 1992). In many environments, demand patterns cannot be smoothed out this well meaning that buffer sizes must be much higher. In this respect, MRP systems have proven to be more universally applicable than JIT systems, since they cope better with variability.

1.3.2 Drum-Buffer-Rope Scheduling

Drum-buffer-rope (DBR) is a scheduling tool based on TOC ideas developed by Goldratt (Goldratt and Cox, 1992). DBR assumes the existence of a bottleneck resource and acknowledges that the throughput of the facility will be dictated by that of the bottleneck. DBR then advocates placing priority on keeping the bottlenecks busy. Work is fed into the system at a rate consistent with the bottleneck resource's throughput. The objective is to keep the bottleneck resource busy while minimising inventory flowing to the bottleneck. It could be argued that with DBR, work is pulled through to the bottleneck, and pushed downstream from the bottleneck.

DBR, like JIT, forces people to examine what is really going on and find ways to improve the situation. Both JIT and DBR stress continuous improvement. A big benefit of TOC is that it has challenged the traditional thinking which encourages non-bottleneck resources to keep producing unwanted parts just to maintain high efficiencies. Instead, it puts forward the concept that lot sizes at non-bottleneck resources may be lowered (even with additional setups that will be incurred) since there is capacity left over (Enns, 1995a). This dynamic variation of lot sizes leads to lower inventories and flowtimes.

It is anticipated that dynamically setting lead times in MRP would have the same effect as DBR, since lead times at the bottleneck resource will instinctively be higher to reflect the longer queues. Perhaps it is time that MRP systems are developed to take account of these ideas and move forward into the 21st century.

Chapter Two

LITERATURE REVIEW

This chapter presents a review of topical literature. It is divided into four sections. Literature on flowtime prediction is presented in the first section. This is followed by a review of load-oriented manufacturing control. Section three looks at the development of MRP systems. The final section considers agile manufacturing systems.

2.1 Flowtime Prediction

Flowtime prediction pertains to the ability to accurately forecast how long an order will take to progress through a workstation or the entire shop floor. Flowtime prediction is important as it would allow planned lead times to be adjusted in order to maintain their validity. The use of valid lead times allows the MRP to compute valid release and due dates. Most of the literature on flowtime prediction and due date setting concentrates on job shop environments (Conway et al., 1967). After this literature is reviewed, some research which considers assembly is examined.

Flowtime estimates can be static or dynamic. Static estimates may be found from queueing analysis or steady-state simulation results and are constant over time. Dynamic estimates may be obtained using various inputs as predictors. For example, regression equations which incorporate shop status and job information may be developed. They can be derived to predict actual lead times (flowtimes) using historical data, and may include terms like *total work in shop* and *order batch size*. As their values fluctuate, the predicted actual lead time will change too. The estimates of actual lead times from these models may be used to set planned lead times.

One form of static flowtime prediction is queueing analysis. Very briefly, a system is treated as a network of queues. Using established relationships and utilisation levels, queue lengths and flowtimes may be determined. Planned lead times are calculated based on steady-state flowtime estimates and used in due date setting. A more complete account of this process is given in Enns (1993). Several tools are now commercially available which use queueing heuristics including *Queuing Network Analyser* (Whitt, 1983) and *MPX* (Network Dynamics, 1991 and Suri and de Treville, 1991)

Baker (1984) surveys sequencing and due date assignment rules in job shops. While none of the due date assignment rules considers shop status, he concludes that due dates should reflect work content. Recently due date setting has started to consider dynamic shop status. Bertrand (1983) uses time-phased workload information and time-phased capacity information to set due dates. Jobs are rescheduled for later periods when capacity is unavailable. A reduction in lateness variance is reported. A survey of due date assignment rules by Ragatz and Mabert (1984) concludes that rules which consider shop status and job information (e.g Jobs In Queue) perform better than those which only consider job information. Vig and Dooley (1991)propose a mixed estimate by combining static and dynamic estimates in a linear weighted form. The aim is to combine the accuracy (no bias) of static estimates with the precision (low variance) of dynamic estimates. The method reduced but did not eliminate bias. Chang (1996) develops a heuristic for dynamic job shop scheduling which estimates queue times and feeds them back to the scheduler for improved performance. The most significant factor is identified from samples using analysis of variance (ANOVA, see Devor et al., 1992). This is followed by construction of a rule based on this factor. His results show that the use of queue time estimates improve due date performance (mean tardiness and percent tardy). Cheng and Gupta (1989) survey due date assignment rules for job shops.

The literature reviewed which examined the use of shop status information supports the conclusion that making use of shop status information when setting due dates helps improve due date performance. Lawrence (1995) models flowtime prediction as a forecasting problem. The actual flowtime is made up of a flowtime estimate plus an error term. The error term is a random variable estimated using a method of moments. Job lead times and due dates are then calculated. The method works well in single server networks but performance deteriorates in more complex environments.

Goodwin and Goodwin (1982), study the relative impact of different operating policies on the performance of an assembly shop. They show that not all job shop research can be generalised to assembly systems. Fry et al. (1989) test several due date setting rules in an assembly environment. Nine different product structures (3 tall, 3 flat, 3 mixed) are analysed in a simulation using the earliest due date (EDD) sequencing rule. The due date setting rules include total work content (TWK), total work content on the critical path (TWKCP), work in system (WINS), and combinations of these three. As with the literature on job shops, they conclude that rules that consider shop status information (combination of TWK or TWKCP with WINS) work best. Enns (1995b) presents a forecasting approach to flowtime prediction. The adaptive forecasting model (AFM) developed considers shop loading, workload conditions and job characteristics in setting due dates. The model adjusts to changes, since data on due date performance is dynamically fed back to it. The model performs well under unbalanced shop loads. This model is then adapted by Enns (1995c) for assembly environments. Flowtime forecasts are determined for various production stages and are stacked according to product structure. As in MRP, release dates for components are obtained by backward scheduling from the end item due date. Lead times for operations are updated. Unlike MRP, this model does not generate assembly order releases, and net requirements are assumed to equal gross requirements. In other words, since all order releases are assumed to be dedicated to a specific end item requirement, lot-for-lot batch sizing is used and assembly is assumed to be authorised at the time all components are available. The model performs well under assembly conditions.

The next step in this evolutionary process is to link an actual MRP system to a shop floor emulated by a simulation model. If lead times are adjusted to reflect actual shop conditions, a dynamic MRP system which can respond to changes in the production environment should result.

2.2 Controlling Actual Lead Times (Flowtimes)

As with the literature on flowtime prediction in section 2.1, most of the literature on control of lead times also focuses on job shops. It is generally acknowledged that queueing times frequently make up 90% or more of actual lead times for a product. Hence it is important to control lead times for several reasons. Firstly, the time spent queueing is not productive beyond what is required to buffer against uncertainty and variability. Secondly, long queues lead to congested shop floors which are difficult to manage.

Wight (1970) identifies erratic order input and lack of control over output rates, together with lead time inflation as the reasons why many plants have very long backlogs (in some cases 1 year or more) when two weeks would normally suffice. Such backlogs have their origins in capacity bottlenecks and excessive work input. Wight proposes input/output control to remedy the situation. His ideas are based on the axiom that shop floor input must not exceed shop floor output capabilities. He also places responsibility for setting order priorities and order release squarely with the production control department and not the shop foreman. Erratic customer demand is smoothed out to maintain *planned* rates of input. Onur and Fabrycky (1987) develop and test an input/output control system for a job shop. Input and output are controlled for the whole shop, not individual work centres. Job release and capacity are controlled to improve shop due date performance.

Spearman et al. (1990b) divide lead time reduction strategies into five categories: elimination of variability; work flow smoothing (includes levelling of work loads); synchronisation of production (between fabrication and assembly, for instance); keep things moving (smaller batches at non-bottleneck work centres); and elimination of *unnecessary* WIP. They recognise the value of WIP at bottlenecks and observe that reduction of mean flowtime and flowtime variance reduces lead times. Spearman et al. (1990a) also propose a new control system called CONWIP (CONstant WIP) for use in flow lines. It allows WIP to collect in front of bottlenecks. They claim reduced levels of WIP when compared to JIT systems.

Bechte (1988) and Wiendahl (1995) propose a control system that is similar but more detailed than CONWIP. It is called load-oriented manufacturing control. Feedback from a job shop is evaluated. Actual lead times are compared to planned lead times. Order release is controlled to keep WIP inventory at a controlled level. This maintains actual lead times at a planned and predetermined level. Orders may be downloaded from an MRP system. In such cases, the lead times used in the MRP can be set equal to the planned and pre-determined level mentioned above. Since shop load is controlled the validity of these static lead times is better maintained.

Watson et al. (1993) use backward simulation to generate component release plans. Starting with due dates, jobs pass through a simulation model of the shop backwards (i.e. assembly operations become dis-assembly operations). The finish time in backward simulation is then recorded as the release date for the component. A forward simulation run is then done to check feasibility. These component plans (which would normally be generated by an MRP system) are then downloaded to a shop floor control system, in this case a simulation-based scheduler. The models for generating these plans are deterministic, much like those used in FCS. Only one replication needs to be made so the simulation is very fast. However, stochastic environment characteristics such as processing time variability, machine breakdown, and future job arrivals are not represented. Deterministic models are less realistic than stochastic models since schedules quickly become invalid as uncertainty is introduced.

Hoyt (1978, 1982) puts the blame for poor MRP performance on the improper setting of lead times, normally set to cover all scenarios. Planned lead times are supposed to indicate the time for a job to go through the shop floor. If they are wrong, the release and due dates calculated will also be wrong. This leads to a host of problems (Hoyt, 1982). Hoyt advocates setting the lead times dynamically for each work station using equation 2.1. Exponential smoothing of the two terms in the equation is suggested to reduce MRP nervousness and dampen fluctuations. The lead time file is then updated.

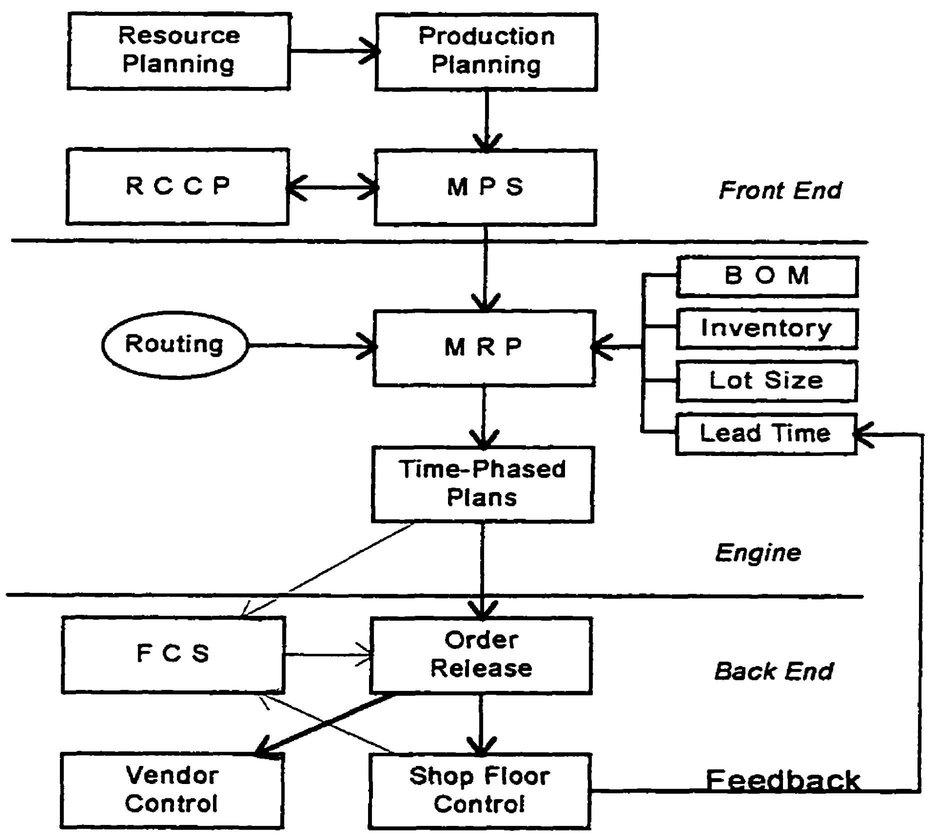
$$Actual\ Average\ LT = \frac{Average\ Queue\ for\ Period}{Average\ Output\ for\ same\ Period} \quad \textbf{(2.1)}$$

Such a calculation of lead time considers queue times, shop status, transfer times, setups, and almost any other factor.

2.3 Developments in MRP

Despite its weaknesses, MRP has developed into perhaps the dominant production planning system in North America today. Some of MRP's strengths include the ability to handle large volumes of data and many changes. Whybark and Williams (1976) identify four sources of uncertainty (combinations of demand or supply, and timing or quantities). They propose a safety lead time concept to cover for uncertainty in timing and safety stocks to cover uncertainty in quantities. MRP systems are insensitive to capacity, that is, they assume that what can be scheduled can be made. It is assumed that capacity considerations are taken care of in the formulation of the MPS. The earliest attempts to consider capacity were the rough cut capacity planning (RCCP) methods. They were designed to ensure MPS feasibility before the MRP generated its plans. This was to avoid unnecessary MRP runs, since computer time was expensive. These methods were only approximate. The next development was the closed-loop MRP which included a capacity requirement planning (CRP) module. Oden et al. (1993) offers an excellent review of CRP. The closed-loop is highlighted by the **bold** arrows in figure 1.1. The CRP module checks the plans generated by the MRP for feasibility. If infeasible, adjustments should made in the MPS and/or to capacity before the MRP is run again. If feasible, the time-phased MRP plans are released to the shop floor. Enns (1995a) identifies several problems with CRP. While a plan may be feasible in CRP, that is no guarantee the work can be completed within a specified time bucket. Work is placed in buckets specified by stacking lead time allowances during backward scheduling. If the lead times are invalid, work gets placed in the wrong buckets. There is also another problem related to lead time setting. As shop loads increase, average operation waiting times are also expected to increase due to longer queues. Since the lead times used by CRP remain unchanged, it does not anticipate changes in expected waiting times. The loading profiles generated by CRP become less realistic. CRP will not fix problems relating to capacity; it is up to the planners to manually fix the problem. CRP uses infinite loading assumptions, like MRP, so capacity is virtually ignored when the load report is produced. Capacity violations must be manually identified by the planners and fixed. Lastly, CRP is not a scheduling tool, since it does not determine specific start times for operations or even the sequence in which to process jobs competing for the same machine.

The most recent development has been finite capacity scheduling (FCS) systems. Wyman (1993), Roder (1993) and Enns (1995e, 1996b) describe FCS in greater detail. When FCS systems are run under MRP, detailed schedules for all operations are generated based on MRP release and due date outputs. These schedules can be displayed as Gantt charts. Detailed schedules also provide forward visibility, so problems are identified earlier. Several problems are still outstanding though. Any changes in shop conditions will render the schedule invalid. If the frequency of changes is high, forward visibility decreases precisely when it is required most. The other problem is that of lead times. If the lead times used by the MRP to calculate release and due dates are invalid, then the schedule generated by the FCS is also invalid. Static lead times, designed to cover all scenarios, are almost always invalid. Establishing a feedback loop to the MRP would allow lead times to be adjusted based on current work load conditions. The focus of this research is to try and



Adapted from Vollmann, Berry, Whybark (1992) and Enns (1995a)

Figure 2.1 Production planning and control with feedback on shop status to the MRP establish this feedback. Figure 2.1 illustrates one way the MRP system could look with such a feedback loop.

There have been many other developments. Many relate to the development of add-on modules to support other functional areas like finance, accounting, human resources and marketing. This has led to the acronym MRPII (manufacturing resources planning) to distinguish it from the basic MRP. Fogarty et al. (1991) offer a good description of MRPII. However, there are still some fundamental issues which seem to have been overlooked in the drive for a better MRP system. With very few exceptions the basic logic behind MRP has remained unchanged for over thirty years. One such exception is put forward by Piper and Kuik (1988). They suggest a continuous delivery MRP (CDMRP) system. Unlike conventional MRP systems where all input materials must be delivered prior to the start of a planned order, CDMRP allows components to be delivered in small transportation batches to the point of use. Whereas in conventional MRP the lead times are stacked according to product structure, lead times in CDMRP can overlap, with an overall reduction in both WIP inventory and product lead times. CDMRP works best when variability is minimised and setups are short, much like in JIT environments. CDMRP seeks to make production *flow* continuously.

2.4 Agile Manufacturing Systems

This section presents a general look at world manufacturing trends. From a technological point of view, there has been a trend towards automation in discrete part manufacturing. This trend started with Henry Ford in 1909 and has been accelerated by the advent of the computer in the 1950's. Robots were introduced in the 1960's and by the 1970's, computer numerical control (CNC) was a reality. In the 1980's flexible manufacturing systems (FMS) were becoming commonplace. From the production management point of view, there have been two world streams. In the 1950's and 1960's, throughput was the most important consideration for Western companies. This was due to the high levels of consumer demand following the end of the second world war. At the same time, Japanese firms were focusing on product quality in an attempt to gain a competitive edge and enter world markets. By the early 1980's, Japan had become an economic and manufacturing superpower. While Japanese and Western corporations boasted similar advanced manufacturing technology, Western companies were still engrossed in mass production while their Japanese counterparts were practising lean manufacturing. This was exemplified in the development and use of the JIT manufacturing philosophy. As market share was continuously being lost to the Japanese, a radical shakeup of Western thinking was required. Concurrent engineering was the next paradigm to take hold. Bedworth et al. (1991) offer an explanation of what concurrent engineering is all about. Briefly, inter-departmental personnel work together on a project and identify potential problems in design much earlier, when it would be easier, faster and cheaper to overcome such problems.

The current trend (and one that is not likely to change) is towards shorter product life cycles, more customisation, lower volumes, and rapid customer demand. The manufacturing enterprise of the past or the present is not going to be enough for many companies to survive. What is (will be) needed is an ability to make use of people's talents and respond quickly to changing conditions. Innovation will have to come from all parts of the company, not just the R & D department. Agile manufacturing (Kidd, 1994) is the term used to describe such abilities. Concurrent engineering, lean manufacturing or flexible manufacturing alone do not constitute agile manufacturing, yet these and other techniques, tools and methodologies must be present for a manufacturing entity to be agile. Agile manufacturing is not a tool, but a concept. It implies radical changes in the way manufacturing systems, and even whole organisations, are designed. Knowledge will become ever more important, and harnessing the knowledge of people in the entire organisation will be vital for survival.

Motivation for the development of MRP systems in the 1950's was lower inventory levels coupled with better delivery performance. The motivation today for dynamically setting lead times is still the same, that is to lower inventory levels and improve delivery performance. New information technology together with new ideas and approaches make this a possibility. This is consistent with the aims and ideas of agile manufacturing.

Chapter Three

THE EXPERIMENTAL PRODUCTION ENVIRONMENT

The objective of this chapter is to describe and justify the production environment assumed in the research. The first section describes the production facility, including the layout. The second section considers the products to be manufactured in the facility. Section three looks at the demand patterns for the products. Assumptions are stated in section four, and in section five a model of the production environment built using rapid modelling software is described.

3.1 The Production Facility

The production facility assumed is the same as that proposed by Enns (1996). There are four pre-assembly stations, one assembly station and two post-assembly stations. Each of the seven stations has one machine which is capable of performing a single type of operation. No task preemption is allowed, hence once a job is started, it must be finished before that machine becomes available to another job. There is no scrap and machines do not breakdown or require maintenance. The facility works one eight-hour shift per day, seven days a week. There is no overtime allowed. Queue lengths and work-in-progress (WIP) levels are not restricted. Transportation times between machines and stations are assumed to be zero. For every product at each operation, there is a fixed setup time followed by a processing time which is dependent on the batch size. There is variability in the processing time but not in the setup time. Figure 3.1 illustrates the assumed layout of the facility.

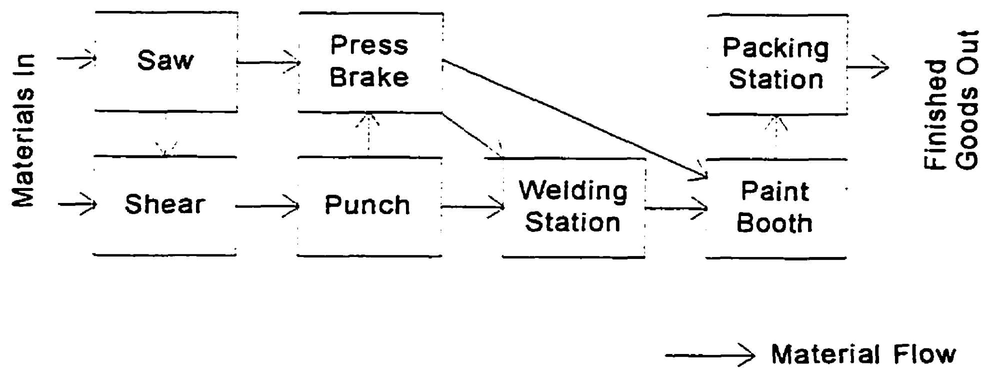


Figure 3.1 Layout of assumed production facility with possible material flow routes

3.2 Products for Manufacture

Two sets of products are identified for manufacture. Each set is made up of two finished products, named P1 and P2. There is no commonality of parts, hence the components that go into making P1 are not required for producing P2, and vice versa. The first set, called the *original* set, is taken from Enns (1996). Both P1 and P2 require use of the same machines, although their setup and processing times are different. Both P1 and P2 include one assembly stage. Two components (C1 and C2) are required to produce one unit of P1 and two other components are needed to manufacture a unit of P2. The product structures for P1 and P2 in the original set are shown in figure 3.2. This figure also contains additional information to the right of the part numbers. The first line indicates the machine required. Line two represents the production rate and line three the setup time.

Two product sets are considered in this research. This is done to test the effect of product structure, if any. Figure 3.3 illustrates the product structures for P1 and P2 in the second set of products, named the *modified* set. P1 includes one assembly stage and is made from one unit each of C1 and C2. P2 does not include any assembly. Additional information on the products in the modified set is also given in figure 3.3.

3.3 Demand Patterns

The demand patterns chosen result in fluctuating shop loads, since production is assumed to chase demand. Most production facilities are subject to fluctuating loads. It is this fluctuation which is thought to be a major contributor to the

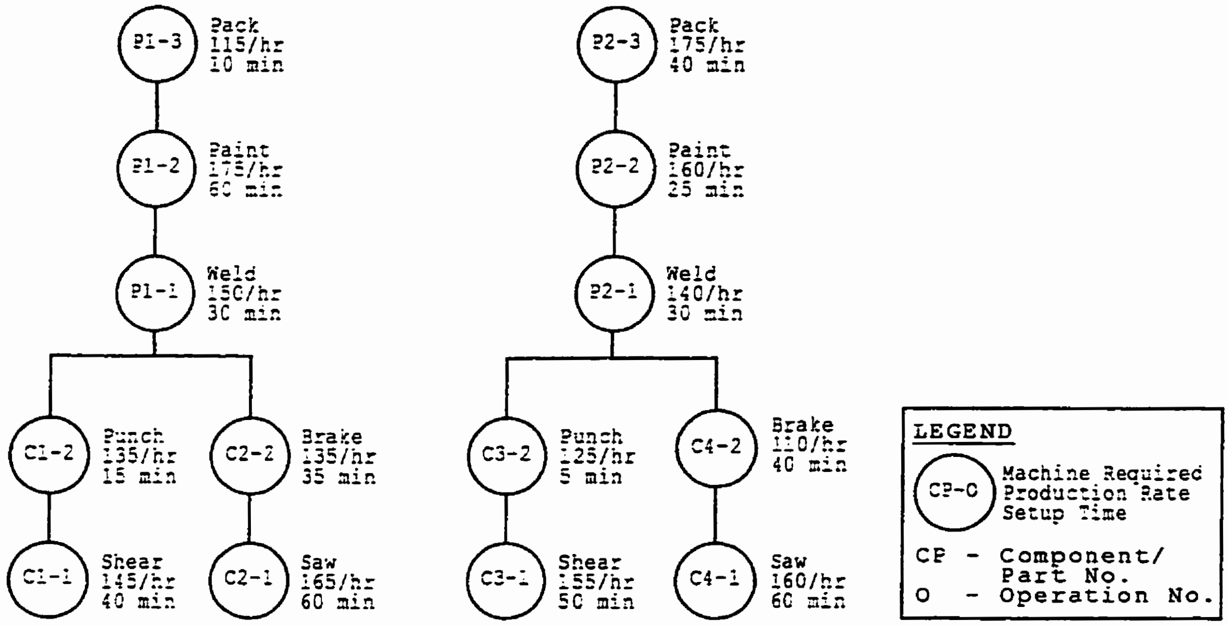


Figure 3.2 The original product set

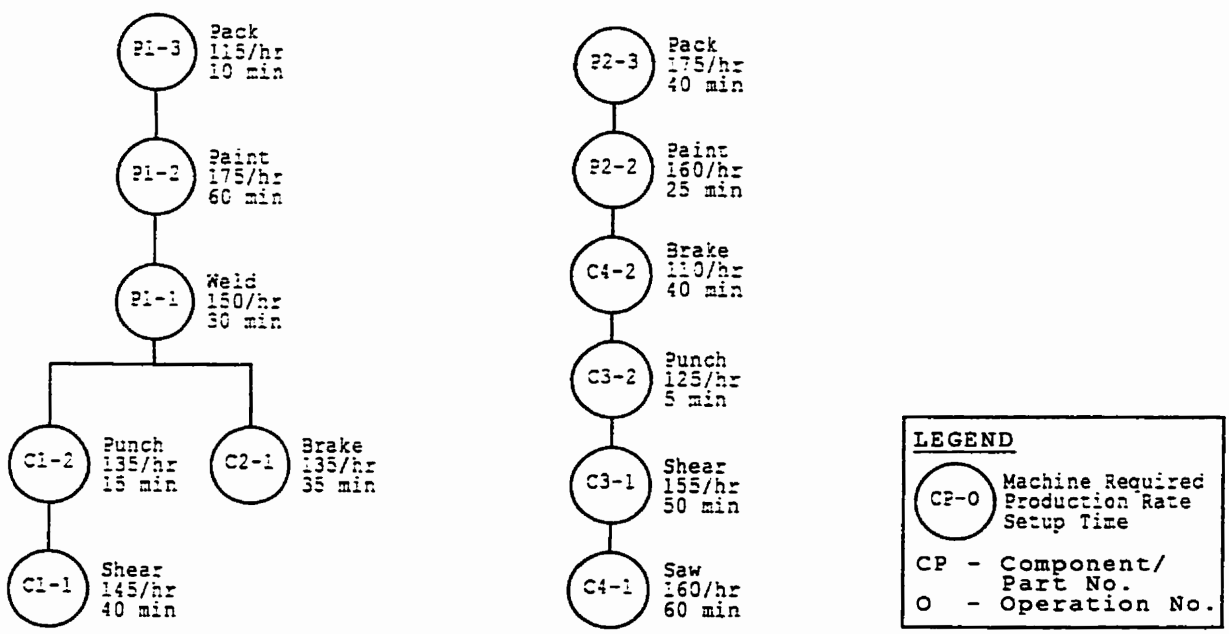


Figure 3.3 The modified product set

poor performance of many MRP systems. Shop loading and queueing times are highly correlated. Since queue times often account for 90% or more of total flowtime, shop loads are also highly correlated with flowtimes. Planned lead times are supposed to be based on flowtime. Hence if shop load fluctuates and lead times remain constant, the quality of release and due date output from the MRP would be expected to deteriorate. Excess inventory or tardy deliveries result.

A seasonal pattern is chosen whereby peak season and off-season *average* demand levels are 25% above and below the annual mean demand respectively. Figure 3.4 illustrates the

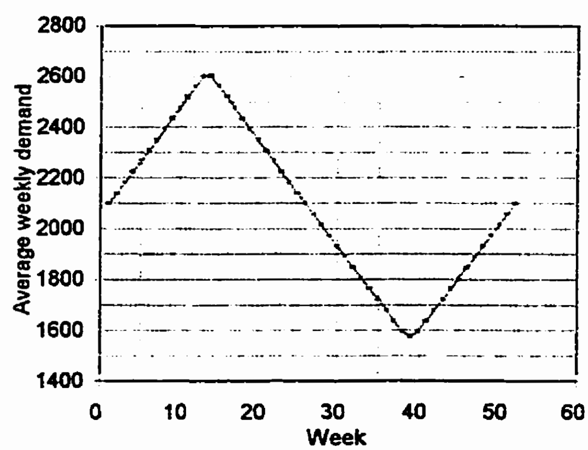


Figure 3.4 Demand pattern subjected to seasonality

expected seasonal demand pattern. Actual weekly demand is drawn from a normal distribution with a standard deviation of 100 and a mean which is equal to the expected demand for that week.

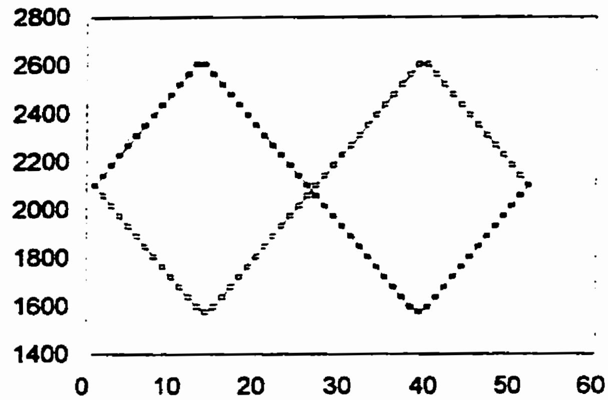


Figure 3.5 a Demand with staggered seasonality

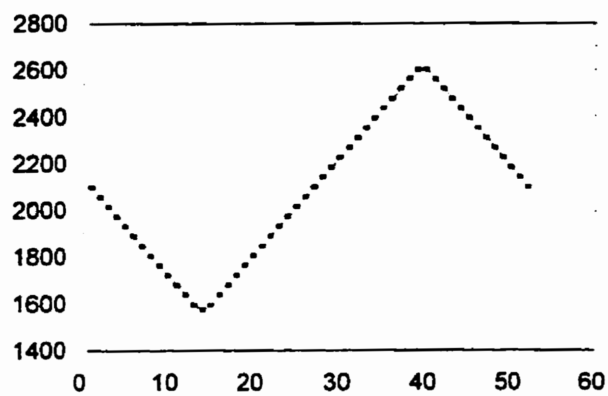


Figure 3.5 b Demand with synchronised seasonality

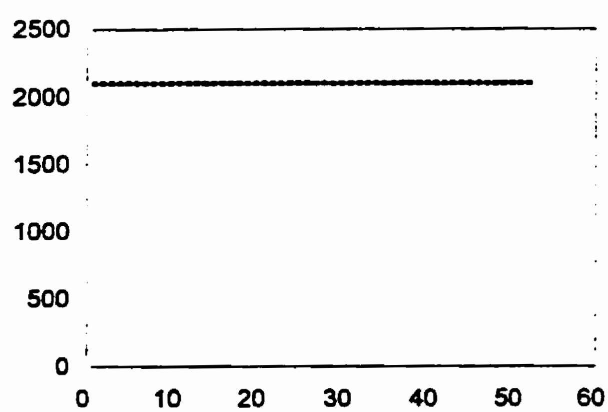


Figure 3.5 c Demand with no seasonality

Three different configurations are tested. In the first scenario, peak season for P1 coincides with the off-season for P2. This is termed *staggered seasonality*. In the second scenario, peak season and off-season for P1 and P2 occur simultaneously. This is termed *synchronised seasonality*. In the third, there is no seasonality in the demand. Figure 3.5 (parts a, b, c) illustrates the three demand pattern configurations.

Three levels of demand are chosen such that at the bottleneck workstation, peak season average utilisations are 80%, 95%, and 105% for a moderately loaded, heavily loaded, and overloaded shop floor.

3.4 Further Assumptions

The lot-for-lot (LFL) batch sizing approach is assumed. This choice is partly based on remarks by Orlicky (1975) that the LFL approach should be used whenever feasible, and that one discrete lot-sizing algorithm is about as good as another. Studies since then have not conclusively disproved this last remark. Melnyk and Piper (1985) showed that in MRP environments LFL works at least as well as other lot-sizing algorithms. Moreover, it is easily implemented and minimises inventory carrying costs (Orlicky, 1975).

A coefficient of variation (CV) of 0.3 is set for the processing times. All processing times are drawn from normal distributions with means, μ, calculated based on figures 3.2 and 3.3. The standard deviations, σ, are then calculated using equation 3.1.

$$CV = \frac{\sigma}{\mu} \tag{3.1}$$

The earliest due date (EDD) dispatch rule is used throughout the facility. It is simple, due date dependent, and offers good performance. There is a choice to make as to which due date to use in the EDD rule. It has been shown by Kanet and Hayya (1982) and Baker (1984) that operation-oriented priority rules perform better than order-oriented rules, in this case end item-oriented rules. Their studies, however, did not assume assembly conditions. The first option is to use the end item due date for all dispatching decisions. The second option is to use the component due date for operations prior to assembly and the end item due date for operations after assembly. MRP systems use the second option since the lot-for-lot rule is not always used when there is commonality of parts across the product line. In this case the end item into which components will go is unknown so an end item (or MPS) due date cannot be used. Under this scenario, MRP systems generate release dates for every stage of production.

The complicating factor in this research is that not all products have assembly requirements. To use the second option, an artificial 'component due date' would have to be created to allow fair competition for resources on the shop floor. A side experiment was set up to test the significance of using one option over the other. The simulation output is shown in appendix 5. It indicates that for the modified product set which includes the product with no assembly requirement, there is no significant difference between the two options. For the original product set, there is a difference in average mean tardiness of about 0.5 days. As the number of assembly stages increases, it is expected that the use of component due dates becomes more advantageous. Therefore, to keep comparisons as fair as possible, the end item due date option (option 1) is selected.

The MRP system sanctions the release of 1 batch per week of each product. The batch sizes are obtained from the master production schedule (MPS). Master production schedules are generated which reflect the seasonal nature of demand, the different levels of average shop loading and the random fluctuation in demand from week to week. Care is taken to ensure that the random numbers used to generate demand values are the same for all shop loads and seasons. In other words, common random numbers are used as a variance reduction technique.

3.5 Modelling the Production Environment

The production environment is emulated using discrete-event simulation. Discrete-event simulation is extremely versatile and can be used to model complex features. Discrete-event simulation and the process of building the model are discussed in chapter four. In order to do some rough preliminary analysis, however, a model of the production environment is constructed using rapid modelling software. The MPX package developed by Network Dynamics (1991) is used. Advantages of this particular package include the ability to handle assembly environments and the impressive graphics used in presenting output. Suri and de Treville (1991) provide an additional description of MPX. The purpose of building the model is to provide a quick check on the ability of the facility to handle the loads imposed on it. The rapid model in MPX also allows *what-if* scenarios, such as changes in the part structure, to be quickly tested. The calculations in MPX are based on queueing approximations. Hence the results will not be exact, but can instead be used as good approximations. The rapid modelling software used also cannot be used to model certain features like variable production rates through time. Finally, the MPX model can be used to help in verifying and validating the simulation model, as described in the next chapter.

Chapter Four

SOFTWARE DEVELOPMENT

This chapter discusses the process of interfacing a simulation of the production environment with an MRP system. The simulation model emulates production floor activity and the MRP acts in a production planning capacity. A major challenge in this research is that information has to be fed back and forth as plans are periodically generated. This is shown in figure 4.1.

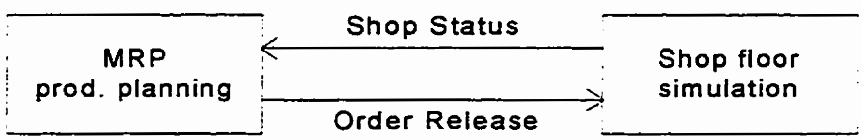


Figure 4.1 Two-way flow of information between shop floor and MRP system

Section one looks at developing the simulation model in SIMAN. The second section considers adapting a spreadsheet-based MRP system to the production environment assumed. Lead time adjustment is based on exponentially smoothed flowtime feedback and is described in this section. In section three the interface and running the system are discussed. The fourth and fifth sections describe the verification and validation efforts. Reader familiarity with the SIMAN V simulation language and with LOTUS™ 1-2-3® spreadsheets is assumed.

4.1 The Simulation Model

The production facility is emulated using a simulation model written in the SIMAN V language developed by Systems Modelling Corporation (Pegden et al., 1990). SIMAN V is chosen for reasons of availability and familiarity. Appendix 1 contains the coding of the model files to run both product sets. The models are divided into four main modules: initialisation, order read in, shop floor emulation and the data collection station. Sections 4.1.1 through 4.1.4 describe the important features of each module. Section 4.1.5 looks at the experiment file. Entities in the model represent BATCHES of parts, NOT individual parts.

4.1.1 Initialisation

Variables in the model, such as parts in current work-in-progress, are initialised at the start of each experiment. This helps the model attain steady state conditions much more quickly and reduces the warm up period. This in turn leads to better computing efficiency. Initialisation values are average values obtained from pilot runs. Initialisation occurs once, at the start of each experiment. A full list of variables is shown on the first page of each model file in Appendix 1.

4.1.2 Order Read In

Every week, one order each for products P1 and P2 is read in to the simulation from a text file. This is done using the READ and CLOSE blocks. Each text file contains batch information on order number, batch size, the lowest level component release dates, and end item due date. A complete list of an order's attributes is shown at the start of the model files in Appendix 1. Each order is then held up via a DELAY block until its release date, or is released immediately if the release date has already passed. Released orders (SIMAN entities) then pass on to the shop floor portion of the model.

4.1.3 Shop Floor Emulation

This portion of the model is made up of the logic for processing at the seven stations (machines). At each station arriving orders enter a queue and are held here until processed on the machine. The feedback mechanism used to update lead times is based on exponentially smoothing operation flowtimes by part type. This method is chosen among the alternatives described in section 2.1 for its simplicity and because many of the alternatives have not been previously tested in assembly environments. In addition, very few data elements need to be maintained. Exponential smoothing of operation flowtimes (ESOFT) is done as shown in equation 4.1.

$$ESOFT_{new} = \alpha(OFT) + (1-\alpha)\,ESOFT_{old} \qquad \textbf{(4.1)}$$

The rate of response when using exponential smoothing can be controlled by selection of appropriate smoothing constants. Selection of this constant is a compromise between responsiveness and stability. In this research, the smoothing constant α is set at 0.1. The flowtime per part (OFT) is calculated on the basis of batch queue times plus setup time plus processing times, divided by the batch size. Exponentially smoothed operation flowtimes *per part* are then calculated and written to a text file. This file contains only one number (the exponentially smoothed operation flowtime per part) and is updated whenever that particular operation finishes processing an order. The ESOFT term is later read by the MRP when it is updating the planned lead times. The order then moves to the next station in its visitation sequence. At the welding station, where assembly takes place, additional logic is used to emulate assembly. When batches of components enter this station, they are placed in a queue where they wait for the other matching components to arrive (e.g. a batch of C1 parts will wait until the matching batch of C2 parts with the same order number arrives, or vice versa). When both sets of components are available, the MATCH block allows them to proceed. The MATCH block provides authorisation to commence assembly. One entity representing a batch of components decrements the WIP counter and is disposed. The other entity proceeds as the assembled item (e.g. P1).

4.1.4 Data Collection

Orders that have been completed on the shop floor pass through the data collection module where data is collected on flowtime, mean tardiness, exponentially smoothed flowtimes, flow allowance (planned lead time) and percent tardy. These measures are explained in chapter five. The data is then written to text files via the WRITE block.

4.1.5 The Experiment File

This section summarises the main features in the experiment files, shown in full in Appendix 2. SIMAN V uses an experimental file to control various experimental inputs and outputs. This file is compiled and linked to the model file, which specifies the system logic, prior to execution. Variable arrays are used to store setup and mean processing times, and processing time standard deviations. The SEQUENCES element defines the station visitation sequences for all products and components. The ARRIVALS element is used to help load up the model at the start of each experiment. Tests showed that loading the model and initialising variables reduced the warm up period by a factor of 10.

The FILES element is used in conjunction with the READ, WRITE and CLOSE blocks to control file access. The DSTATS and TALLIES elements, generally used to assist in collection of data, are used to obtain average response values. This is described in section 4.3.

The REPLICATE element starts one long replication at time 36570. Data collected from this long replication is then truncated to obtain samples. The time units in the simulation are *days*. This is to allow the SIMULATION and the MRP to work in consistent time units. The MRP uses the LOTUS™ 1-2-3® date numbering system, where 36570 is equivalent to February 14, 2000 and 36571 is equivalent to February 15, 2000 and so on. Hence 365 simulation time units are equivalent to 1 year.

4.2 The MRP System

An evaluation of some commercially available MRP packages was carried out. The *Spreadsheet Resource Manager* from User Solutions Inc. (1996) is a spreadsheet-based MRP package which not only is the cheapest of those investigated but is shipped with all the source code. It has a bill of materials processor and is capable of forward or backward scheduling. A variety of reports can be run against the generated schedules. These include capacity load reports and Gantt charts. It is a macro-based package requiring user input at various stages. In order to allow interfacing, several features are added. These are described below.

4.2.1 Shop Floor Feedback

The observed exponentially smoothed operation flowtimes per part (ESOFT) that are written to text files by the simulation model (section 4.1.3) are read into the MRP. This is done via a LOTUS™ 1-2-3® macro. There are seven workstations, each working on one of two possible products. This results in 14 values to be read, each value representing the flowtime for a part at one machine. Each value is held in a separate file and is updated in SIMAN V independently of the other 13 values.

4.2.2 Updating Lead Times

After the MRP has read in the smoothed part flowtimes, the new order batch sizes are read in from the MPS. Operation planned lead times (OLT) are then calculated as in equation 4.2.

$$OLT = ESOFT * Batch\,Size \qquad \textbf{(4.2)}$$

The *ESOFT* term in equation 4.2 accounts for shop load and the *Batch Size* term accounts for the size of the new order. All 14 operation lead times are calculated in this way. The bills of materials (which, in this particular MRP system, contain lead time data) are then updated. Lead times are updated weekly just prior to the regeneration of the next weekly MRP plan.

4.2.3 MRP Explosion

An MRP explosion is the term used to describe the process which determines the required quantities and timing for the production or procurement of components and raw materials needed to build the end items on time. Prior to the MRP explosion, the due dates for product orders are known. The MRP is fed this information, together with part numbers and quantities required. Using backward scheduling (Oden et al., 1993), MRP computes operation start dates by subtracting the operation lead times (equation 4.2) from the due date. Starting with the end item due date at the final operation, this process continues backwards through the product structure until the lowest-level components have been processed.

4.2.4 Writing Out Data

When the MRP explosion has taken place and release and due dates have been calculated, certain information from the schedule is searched and recorded. This information (order number, batch sizes, release and due dates) is then written to text files.

It is definitely possible to extract all operation due dates for a particular order and write them out to a file. However in order to keep the system simple, only the end item due dates (along with order numbers, batch sizes, and release dates) are extracted. The data for P1 is written to one file and the data for P2 to another file.

4.3 Interfacing and Execution

The simulation runs in SIMAN V, which is an MS-DOS® program. The MRP runs in LOTUS™ 1-2-3®, which is a Windows™ program.

There are two basic methods to interface the two programs. In the first method, coding is developed in SIMAN V to launch the MRP regeneration cycle periodically. Developing the coding in SIMAN V results in the whole experiment running a little faster and the data collection system being straightforward. This option requires the use of an EVENT block in SIMAN V, which is visited by an entity whenever an MRP regeneration is required. The event block passes control to a user-coded event which executes a subroutine in FORTRAN or C. The main drawback to this method of interfacing is to locate (or create from scratch) a function which will execute a LOTUS™ 1-2-3® macro from a program running in an MS-DOS® shell.

In the second method, the interface is coded in LOTUS™ 1-2-3® and the simulation model launched after each MRP cycle. The whole experiment will run slightly slower. After an MRP cycle has completed, LOTUS™ 1-2-3® executes a SYSTEM call, which means it executes a command in MS-DOS®. This command launches the simulation in SIMAN. This method is much easier to code and is the one chosen. Command-line switches may be used to launch SIMAN V directly into the interactive debugger. A file with all the necessary commands is then read in and the commands automatically executed from within the interactive debugger. Placing *END* as the final command in the file causes the simulation to terminate. The following interactive debugger commands (briefly explained below) are placed in the command file called "snp.txt":

```
RESTORE "snapshot.snp"
GO UNTIL
SAVE "snapshot.snp"
END
```

The RESTORE command restores the simulation to the status when it was previously terminated. The status is kept in the file 'snapshot.snp'. GO UNTIL TNOW+7 causes the simulation to advance 7 time units (days). In these 7 days, orders will be read in, and shop status updated. The SAVE command writes system status to the file 'snapshot.snp'. END causes the simulation run to terminate, with control returning to the MRP.

When the SYSTEM call follows the MRP cycle, a command is issued to load the simulation in SIMAN V and the above commands are executed. Figure 4.2 shows this repeating cycle. The main drawback to this method is that data collection via the DSTATS and TALLIES into data files is not possible. The DSTATS element in SIMAN V collects time-dependent statistics for things like resource utilisation and queue lengths. The TALLY block and TALLIES element record observational data like

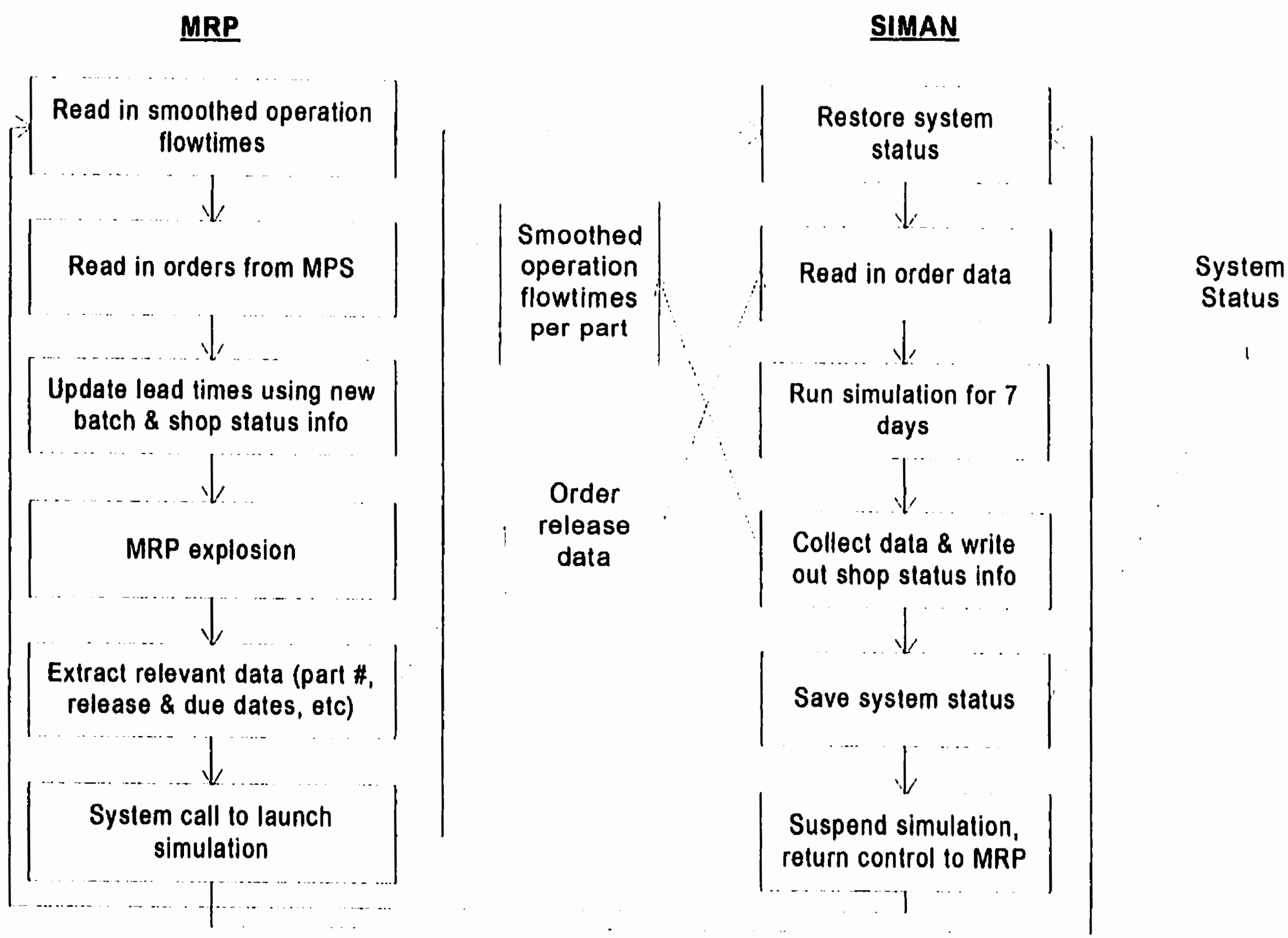


Figure 4.2 Schematic diagram of system operating cycle

average flowtime. Every time the simulation is launched, all the relevant output data files are reset. This necessitates use of the WRITE blocks, described in section 4.1.4, as an alternative method of output data collection. In conjunction with the FILES element, the WRITE block can be made to append to a file every time it accesses that file. The data collected in these files is then loaded into any statistical analysis package.

4.4 Model Verification

Verification is defined as the process of determining that a model operates as intended (Pegden, Shannon, Sadowski 1990). It involves making sure that syntax and logic errors are removed from the model. There are many ways to verify a model. Some of the methods used in this case are described below.

Syntax errors are identified by the SIMAN compilation and execution programs (Model.EXE, Expmt.EXE, Linker.EXE, and Siman.EXE) and are easily remedied. Such errors usually involve the omission of punctuation marks. Catching logical errors involved the use of the TRACE element, the interactive debugger and walkthroughs with persons familiar with the SIMAN V language. The TRACE element records the detailed movement of entities within a model. This is useful in locating flow-of-control and initialisation errors. The interactive debugger provides similar capabilities as the TRACE element but its interactive nature provides greater flexibility. The interactive debugger helped resolve a logic error involving the treatment of assembled parts after exiting the MATCH block in the welding station. The walkthrough sessions identified errors in the way statistics were calculated.

4.5 Model Validation

Law and Kelton (1991) define validation as being concerned with determining whether the conceptual simulation model (as opposed to the computer program) is an accurate representation of the system under study. There are several ways to validate a model, depending on the circumstances. In this case, the production environment has already been designed, tested and run using rapid modelling by Enns (1996). This is the least difficult type of case to validate. The tests carried out to try and validate the model are grouped under two broad categories as suggested by Pegden, Shannon, Sadowski (1990).

The first category are tests for reasonableness. A consistency check revealed that changing the random number seeds had a very small impact on the system's long-term performance. The duration of the working shift (8 hours) was inconsistent with the MPX model's shift. This anomaly was corrected. Shop floor resources (machines and labour) were de-activated. The resulting deterioration in system performance was as expected. A check for absurd conditions revealed the occurrence of negative WIP values. This was traced back to incorrect initialisation of WIP variables. The root of this problem was entities entering via the ARRIVALS element.

The second category tested model structure and data. Although the model was not animated, face validity was established during a walkthrough session with two other persons. Parameters such as means and standard deviations were adjusted to observe the sensitivity of the model to changes. As expected, small increases in variation cause a slight deterioration in model performance.

Two simple analytical models of the facility are built within a spreadsheet with the aim of predicting equipment utilisation levels for all machines. The models, one for each product set, are shown in Appendix 3. The simulation model is then run using the original product set and the modified product set. In each case, the error in the average shop floor utilisation between the analytical model and the simulation model is less than 0.5%.

The MPX model described in section 3.5 was used as a representation of the real system after it had been verified and validated. The predictive behaviour of the simulation model was tested as new input data was entered and the output results were consistent with those of the MPX model using the new data.

Having ascertained that the model as set up to process parts from the *original* product set was sufficiently validated, the *modified* product set was introduced and many of the tests described directly above were repeated. After being satisfied about the validity of the model, the design of experiments is started.

Chapter Five

EXPERIMENTAL DESIGN

This chapter focuses on the design of the experimental plan. The first section examines the issues considered. Section two looks at experimental factors considered and chosen. The performance measures used are defined and discussed in section three. Section four examines the strategy to carry out the experiments. The fifth section considers the analysis to be done on collected data. Finally, section six outlines the experimental plan.

5.1 Issues Considered

The issues identified and discussed are highly interdependent and should really be considered together. However in the interests of clarity, they are addressed separately.

5.1.1 Components of Flowtime

The flowtime for a job, defined as the time a job spends on a shop floor (Baker, 1984) can be split up into many components. In an ideal situation the flowtime will equal the processing time. In reality, parts have to be setup on machines prior to processing and inspected after processing. This leads to setup and inspection times. Reduction (or elimination) of setup times is an important part of JIT. Queueing times are generally believed to account for 80-90% of the total flowtime (Fogarty et al., 1991), hence they deserve the attention directed towards trying to reduce them. Batch times occur when processed parts must wait for the rest of the batch to be processed before moving on. Transfer time includes transportation between machines and stations, to and from storage, and sometimes includes time spent waiting for a transporter. In this research transfer times are ignored.

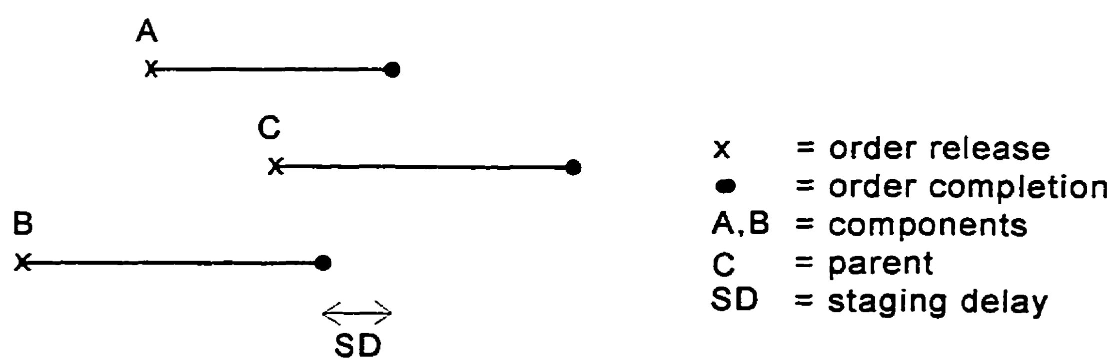


Figure 5.1 Diagram depicting staging delay

In assembly environments, two special forms of delay often occur. The first is called staging delay and is depicted in figure 5.1. It occurs when one component is ready for assembly but another component needed is not yet available, or when assembly is scheduled to have started but both components needed are unavailable. This can occur when lead times are set too tight, or there is an unexpected increase in orders.

In the second type of delay, known as order release delay and shown in figure 5.2, both components are ready to assembly, but the order release for assembly has not yet come into effect. This situation can arise when lead times are set too high in an attempt to cover all scenarios, or when there is a sudden drop in orders. When planners increase lead times to improve delivery performance (mean tardiness, % tardy), there will be more time allowed for operations to complete. As processing times will not change, jobs will simply wait longer for the next stage's order release. This leads to increased WIP levels. These last two delay components of flowtime illustrate the special need to maintain valid lead times.

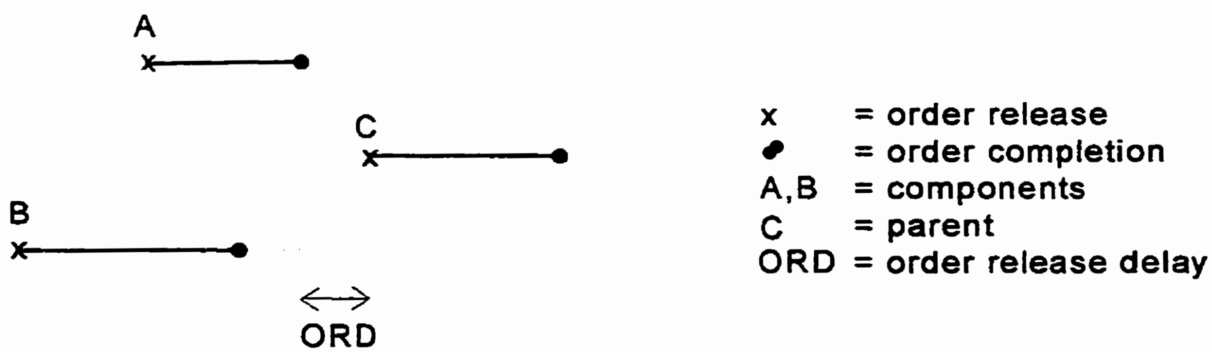


Figure 5.2 Diagram depicting order release delay

5.1.2 Lead Times

The importance of maintaining valid planned lead times cannot be overstated. Lead times are used to establish how much time is allowed for an order to pass through the shop floor. The size of the order will make a difference, as will the level of congestion on the shop floor. These lead times are used in calculating release dates and due dates. When setting static lead times, production planners use demand forecasts and historical estimates, as well as their intuition. While it is possible that general trends in customer demand can be predicted, it is almost impossible to forecast the exact quantity and timing of customer orders. Quantity uncertainty is offset by higher levels of finished goods inventory. Lead times as set by production planners are set to cover worst-case scenarios. They are almost always invalid. This means that MRP will set incorrect release dates for orders. WIP will build up on the shop floor, and actual lead times will get inflated due to higher queueing times.

One major weakness of MRP systems is the assumption of infinite capacity. Finite capacity scheduling (FCS) systems used with MRP are one recent attempt to overcome this weakness. Release and due dates are downloaded from the MRP and the FCS creates a schedule for the shop floor, taking capacity into account. The use of invalid lead times means the FCS is being fed invalid release and due dates with which to work. A fuller discussion on FCS is given in Roder (1993) and in Wyman (1993).

Finally, lead times are self-fulfilling (Hoyt, 1982) meaning that the more time is allowed for an order to complete, the longer that order will actually take to complete. This is illustrated in figure 5.2. The lead time is increased to reduce the probability of lateness. Components then spend more time waiting for an order release. Hence keeping lead times as long or as short as actually needed will keep the production planning and control system running more smoothly.

5.1.3 Flowtime Prediction

In order to determine the correct planned lead times, it is necessary to know how long an order will take to pass through the shop floor. While this is not possible, it is always possible to *predict* flowtimes. Several algorithms have been developed which are quite accurate in predicting flowtimes. Some are examined below. Lawrence (1995) uses a flowtime estimate plus an error term drawn from an error density estimated using the method of moments (Winkler and Hays, 1975). This method has not been tested in assembly environments. Enns (1995b) has developed an adaptive forecasting model (AFM). This model forecasts and sets internal due dates in a job shop. Internal due dates are set by the producer (not the customer) and are calculated as arrival time plus sum of setup & processing times plus sum of expected waiting times. Enns (1995c) has adapted the AFM to an assembly environment. Using backward scheduling, as in MRP, the model forecasts flowtimes with little or no bias. However, unlike MRP, it only generates release dates for bottom level components. Watson et al. (1993) use backward and forward simulation runs to generate release and due dates based on expected flowtimes. Up to three iterations may be required. Hoyt (1978, 1982) advocates using exponentially smoothed actual queueing time as a feedback mechanism to set lead times dynamically.

The selected approach is to exponentially smooth operation flowtimes and feed them back to the MRP to update lead times. As an operation finishes, the flowtime *per part* is exponentially smoothed and written to a text file. The operation lead time in the MRP is then set as described in section 4.2.2. It could be argued that, as with Hoyt's method,

same random numbers are used across all levels of loading as a variance reduction technique.

$$D_i = \mu_i + \sigma(\cos(2\pi rand)) * \sqrt{(-2\ln(rand))} \quad \textbf{(5.1)}$$

where D_i = demand for each product

rand = uniformly distributed random number between 0 and 1

5.1.5 Safety Factors

It is not unusual to find that lead times include safety time in many MRP systems. This is done in an attempt to improve on-time delivery. A safety lead time may be added to cover uncertainties in the timing of orders. Uncertainty pertains to possible changes by the customer and/or supplier (e.g. supplier puts back delivery date). The size of this safety lead time is critical. If it is too large, many orders will be released into the shop floor much too early, and this causes a congested shop floor and poor shop performance. It is also then much harder to set order priorities correctly. If the safety lead time is too small, orders will always struggle to keep up with the set due dates. The need for expediting increases and the formal production planning and control system can break down as shop foremen discard its outputs. Hoyt (1982) recommends building in a safety factor to avoid stockouts and decreased service levels. In a system with no bias, orders will only *on average* be completed on the due date. About half will finish early and half will be late. Since this research is essentially comparing two alternatives (static lead times and dynamical lead times), safety factors are deemed irrelevant if both cases are treated the same way. It also avoids the issue of how much safety time to add.

5.2 Experimental Factors

Four experimental factors are identified. A full factorial experiment is designed to test the significance of the four factors described below. Two of the factors have 2 levels each and the other two factors have 3 levels each. The first factor is demand seasonality. This factor is the primary mechanism for fluctuating shop loading within an experiment. This factor has three levels: no seasonality, staggered seasonality and synchronised seasonality, as described in section 3.3 and shown in figure 3.5.

The second factor is the average shop load within an experiment. The first level sets the peak bottleneck utilisation at 80%. This translates to an average weekly demand per product per week of 2100 units. For the second level, 95% utilisation translates to 2500 units, and for level three, 105% utilisation (i.e. overload) translates to 2800 units.

The third factor in the experiments is the product structures. At one level products from the modified product set (figure 3.3) are manufactured in the facility. At the other level the original product set (figure 3.2) is used.

The final factor is the type of lead time control employed. Static lead times represent one level and dynamically set lead times represent the other level. A side experiment was set up in which different values for the exponential smoothing constant, α, were tested. The selected value is a compromise between responsiveness and stability in the system. α is set to 0.1.

5.3 Performance Measures

In research studies flowtime and due date performance are important criteria. In actual manufacturing practice, meeting due dates tends to be more important than minimising flowtimes (Baker, 1984). Due date performance is therefore set as the primary performance criterion in this research. Mean tardiness is used to judge due date performance. It is defined (in equation 5.3) as zero or the amount by which a job is late, whichever is greater. The notation used is consistent with the list of abbreviations (page ix following table of contents).

Flowtime is defined as completion date less the release date. It is a secondary performance criterion. Lateness is defined as the completion date less the due date. Lateness (equation 5.2) may be positive or negative, mean tardiness may be positive or zero. Percent tardy (PT) measures the proportion of jobs that are late (equation 5.4).

$$L_i = C_i - d_i \tag{5.2}$$

$$T_i = \max(L_i, 0) \tag{5.3}$$

$$PT = \frac{Number\ of\ jobs\ where\ L_i > 0}{Total\ Number\ of\ Jobs} \tag{5.4}$$

Many other measures are also used, mainly as checks on the experiments. Flow allowance is the time between release date and due date. Every attempt is made to ensure the flow allowances for corresponding static and dynamic lead time experiments are equal. Lateness and percent tardy are used to check for bias in the system. A system with no bias will have zero lateness and 50% tardy, meaning that on average, orders will finish on the due date, an equal amount finishing early and finishing late. Average utilisation is compared and checked against results from the analytical and MPX models. Average utilisations for static and dynamic lead time experiments are set to be equal.

5.4 Operation Strategy

This section describes how the experiments are run. In order to help ensure the lead times (flow allowances) are the same for corresponding static and dynamic lead time experiments, the dynamic lead time experiments are run first. The average flow allowances are then fixed for the static lead time experiments to be equal the average dynamic flow allowance. The static lead time experiments are then run as for the dynamic lead time experiments, but with no updating of lead times. This strategy eliminates the need to correct the results for unequal flow allowances. It is worth noting that there is no prescribed method for setting static lead times. The method used in this research has the advantage of using results from an actual 'production' run (the dynamic run). This is rarely the case in actual manufacturing practice.

5.5 Analysis

This section describes the analysis that is done after the experiments are run. Data from a long run is truncated before tests are run.

5.5.1 Data Truncation

The batch means approach (Pegden et al., 1990) is used to obtain data for analysis. Here, one long run is made and the sequence of data is divided into independent sub-sequences. Each sub-sequence is treated as an independent sample. In cases where the transient (warm up) phase is large, as is the case here, this approach wastes less computer time, as the warm up phase is encountered only once. An accepted rule of thumb for ensuring independent sets of data is that batch size should be at least ten times as large as the largest lag for which correlation between observations remains significant. The data from one long replication is truncated into batches ten times the size computed. Due to the slow execution speed of the system, this approach is impractical. Instead, another approach is used.

An experiment is run and the warm up period truncated. The largest lag for which the correlation remains significant is determined, as before. For most experiments, this lag is expected to be about 26 observations, or 26 weeks, which corresponds to the onset of the low demand season. For the remainder, it is expected to be only a little higher. The batches are then defined as a fixed duration of time (52 weeks, or 1 year). The remaining data from the one long run is then divided as follows:

1st year	→	sample #1
2nd year	→	discard
3rd year	→	sample #2
4th year	→	discard
5th year	→	sample #3
.		.
.		.
.		.
38th year	→	discard
39th year	→	sample #20

Run length is then fixed at 39 years plus a warm up period. A side experiment was run to determine a suitable length for the warm up period. The warm up period lasts about 4 years.

5.5.2 Statistical Packages

The truncation method described above is carried out in the MINITAB statistical analysis program, version 9.2 (Minitab, 1993). The data (20 samples) is then written to a text file. This data is then imported into the SIMAN IV Output Processor, version 1.12 (Systems Modelling, 1989). Both these packages are selected for reasons of availability and familiarity.

5.5.3 Statistical Tests

The first test is done in the SIMAN IV Output Processor (OUTPT). It is a CORRELOGRAM on the flowtime data collected. This test assumes an autocovariance-stationary process and computes sample auto-covariances and auto-correlations over a range of lags (Pegden et al., 1990). It is carried out to check whether the truncation method described in section 5.5.1 will yield samples which violate the independence assumptions. The second test is a paired-t test. This test is carried out on truncated mean tardiness data in OUTPT using the COMPARISONS command. This technique is appealing since the variances from the two groups (A and B) can be unequal and the observations between A and B need not be independent. It is only necessary that observations within each group be independent. The test generates a confidence interval on the difference of means for two sets of data (one set for dynamic lead times experiment, one set for corresponding static lead time experiment). For example, this tells us whether, other factors being at the same levels, the use of dynamically set lead times significantly improved performance of the system. The paired-t test calculates the difference between each pair of observations across the two sets of data. Each data set contains twenty observations representing twenty independent replications. Law and Kelton (1991) provide a complete explanation of the paired-t test.

5.6 Experimental Plan

Table 5.1 outlines the experiments carried out, and the order of execution. While it is recommended for physical experiments that order of execution be randomised, it is not important here since there are no 'noise' factors which could result in unaccounted for variation (i.e. randomness is controlled). A legend of the terminology used is given at the bottom of the table.

The experiments are run on a personal computer (80486DX2 processor, 66MHz CPU, 16Mb RAM). Generating one year's worth of shop floor data takes approximately one hour. This hour is broken down roughly as follows:

Update lead times	30%
MRP explosion	25%
Shop floor emulation	10%
Write out order release data	35%
Total	100%

Exp No.	**Demand Pattern**	**Product Demand (Utilisation)**	**Lead Time Control**	**Product Structure**
1	Synchronised	2100	Dynamic	Modified
2	Synchronised	2100	Dynamic	Original
3	Synchronised	2100	Static	Modified
4	Synchronised	2100	Static	Original
5	Synchronised	2500	Dynamic	Modified
6	Synchronised	2500	Dynamic	Original
7	Synchronised	2500	Static	Modified
8	Synchronised	2500	Static	Original
9	Synchronised	2800	Dynamic	Modified
10	Synchronised	2800	Dynamic	Original
11	Synchronised	2800	Static	Modified
12	Synchronised	2800	Static	Original
13	Staggered	2100	Dynamic	Modified
14	Staggered	2100	Dynamic	Original
15	Staggered	2100	Static	Modified
16	Staggered	2100	Static	Original

17	Staggered	2500	Dynamic	Modified
18	Staggered	2500	Dynamic	Original
19	Staggered	2500	Static	Modified
20	Staggered	2500	Static	Original
21	Staggered	2800	Dynamic	Modified
22	Staggered	2800	Dynamic	Original
23	Staggered	2800	Static	Modified
24	Staggered	2800	Static	Original
25	No Seasonality	2100	Dynamic	Modified
26	No Seasonality	2100	Dynamic	Original
27	No Seasonality	2100	Static	Modified
28	No Seasonality	2100	Static	Original
29	No Seasonality	2500	Dynamic	Modified
30	No Seasonality	2500	Dynamic	Original
31	No Seasonality	2500	Static	Modified
32	No Seasonality	2500	Static	Original
33	No Seasonality	2800	Dynamic	Modified
34	No Seasonality	2800	Dynamic	Original
35	No Seasonality	2800	Static	Modified
36	No Seasonality	2800	Static	Original

Table 5.1 Experimental Plan Outline

Table 5.1 Legend	
Demand Pattern	Type of seasonality shop floor is subjected to
Product Demand	Average demand per product per week (utilisation)
Lead Time Control	Lead time setting mechanism (Dynamic / Static)
Product Structure	Product set used in the experiment

Chapter Six

EXPERIMENTAL RESULTS AND ANALYSIS

The objective of this chapter is to report on the results of the experiments and present analysis of the output. Section one presents results taken directly from the simulation output. Section two provides analysis including results of statistical tests.

6.1 Results

A sample output summary report from the simulation runs is shown in appendix 6. This sample shows the results from experiment 01 (synchronised demand pattern, low level shop loading, dynamic lead times, modified product structure). The other 35 output summary reports are not shown.

All values in the tables that follow are taken from the 36 (3x3x2x2) output summary reports. They are averages from one long simulation run. Table 6.1 summarises results from all experiments for all products on the shop floor. Tables 6.2 and 6.3 separate these results for products P1 and P2 respectively.

Exp	Conditions	ESFT	FLOW	MT	LATE	UTIL	WIP	PT
1	Synch,2100,Dyn,Modi	12.29	12.21	0.76	0.08	46.83	8442	51.6
2	Synch,2100,Dyn,Orig	11.45	11.42	0.75	0.04	54.32	10200	50.2
3	Synch,2100,Sta,Modi	12.46	12.27	1.13	0.19	46.84	8584	49.9
4	Synch,2100,Sta,Orig	11.35	11.41	0.96	-0.05	54.31	10226	46.6
5	Synch,2500,Dyn,Modi	15.35	15.12	1.11	0.24	55.38	12903	51.1
6	Synch,2500,Dyn,Orig	14.26	14.16	0.96	0.12	64.22	15326	50.4
7	Synch,2500,Sta,Modi	15.19	15.29	1.42	-0.08	55.44	12815	44.9
8	Synch,2500,Sta,Orig	14.04	14.30	1.19	0.25	64.19	14987	44.5
9	Synch,2800,Dyn,Modi	19.02	17.94	2.25	1.10	61.83	18732	55.0
10	Synch,2800,Dyn,Orig	18.05	17.45	1.86	0.63	71.62	22099	52.4
11	Synch,2800,Sta,Modi	18.01	18.87	1.81	-0.82	61.84	18095	40.9
12	Synch,2800,Sta,Orig	17.97	18.63	1.89	-0.62	71.71	21785	42.5
13	Stagg,2100,Dyn,Modi	11.84	11.86	0.71	-0.01	46.87	8157	50.0
14	Stagg,2100,Dyn,Orig	11.00	11.06	0.69	-0.05	54.28	9481	49.3
15	Stagg,2100,Sta,Modi	12.13	11.85	1.04	0.28	46.80	8421	55.0
16	Stagg,2100,Sta,Orig	11.41	10.94	1.12	0.47	54.22	10061	61.1
17	Stagg,2500,Dyn,Modi	15.26	15.06	1.04	0.22	55.50	12801	52.2
18	Stagg,2500,Dyn,Orig	14.32	14.19	1.01	0.14	64.23	15294	50.1
19	Stagg,2500,Sta,Modi	15.31	15.23	1.54	0.11	55.47	12922	47.6
20	Stagg,2500,Sta,Orig	14.02	14.34	1.15	-0.30	64.25	14992	44.6
21	Stagg,2800,Dyn,Modi	16.98	16.85	1.12	0.15	61.81	15791	51.7
22	Stagg,2800,Dyn,Orig	15.95	16.02	1.09	-0.05	71.64	18372	47.4
23	Stagg,2800,Sta,Modi	17.04	17.00	1.16	0.07	61.87	15975	49.8
24	Stagg,2800,Sta,Orig	16.12	15.91	1.13	0.23	71.63	18712	55.1
25	No Sea,2100,Dyn,Modi	12.27	12.19	0.76	0.08	47.08	8276	52.0
26	No Sea,2100,Dyn,Orig	11.42	11.38	0.76	0.04	54.55	10011	49.9
27	No Sea,2100,Sta,Modi	12.36	12.25	0.78	0.11	47.04	8358	51.7
28	No Sea,2100,Sta,Orig	11.30	11.36	0.67	-0.06	54.52	10029	48.5
29	No Sea,2500,Dyn,Modi	14.92	14.84	0.91	0.09	55.67	12092	50.8
30	No Sea,2500,Dyn,Orig	13.78	13.76	0.88	0.04	64.47	14370	49.5
31	No Sea,2500,Sta,Modi	14.99	14.90	0.92	0.10	55.73	12208	50.5
32	No Sea,2500,Sta,Orig	13.65	13.73	0.77	-0.07	64.49	14397	48.8
33	No Sea,2800,Dyn,Modi	17.46	17.32	1.11	0.17	62.24	16114	51.4
34	No Sea,2800,Dyn,Orig	16.09	16.06	1.00	0.06	71.97	18728	49.3
35	No Sea,2800,Sta,Modi	17.21	17.44	0.92	-0.21	62.19	15926	45.7
36	No Sea,2800,Sta,Orig	15.78	16.02	0.77	-0.21	72.01	18585	46.7

Legend			
Exp	Experiment Number	LATE	Lateness
ESFT	Exponentially smoothed flowtime	UTIL	Shop utilisation
FLOW	Lead time / flow allowance	WIP	Average WIP level
MT	Mean tardiness	PT	Percent tardy

Table 6.1 Results for all products

Exp	Conditions	ESFT	FLOW	MT
1	Synch,2100,Dyn,Modi	11.38	11.34	0.71
2	Synch,2100,Dyn,Orig	11.53	11.49	0.79
3	Synch,2100,Sta,Modi	11.54	11.39	0.94
4	Synch,2100,Sta,Orig	10.28	11.47	0.39
5	Synch,2500,Dyn,Modi	14.18	14.02	0.99
6	Synch,2500,Dyn,Orig	14.33	14.22	1.00
7	Synch,2500,Sta,Modi	14.15	14.12	1.26
8	Synch,2500,Sta,Orig	12.81	14.34	0.61
9	Synch,2800,Dyn,Modi	17.70	17.05	1.86
10	Synch,2800,Dyn,Orig	18.00	17.45	1.79
11	Synch,2800,Sta,Modi	16.85	17.47	1.61
12	Synch,2800,Sta,Orig	18.84	18.53	1.85
13	Stagg,2100,Dyn,Modi	10.93	10.95	0.66
14	Stagg,2100,Dyn,Orig	11.19	11.25	0.73
15	Stagg,2100,Sta,Modi	11.18	10.94	0.95
16	Stagg,2100,Sta,Orig	10.22	11.08	0.38
17	Stagg,2500,Dyn,Modi	14.06	13.93	0.92
18	Stagg,2500,Dyn,Orig	14.38	14.25	1.02
19	Stagg,2500,Sta,Modi	14.25	14.06	1.34
20	Stagg,2500,Sta,Orig	12.79	14.37	0.57
21	Stagg,2800,Dyn,Modi	15.45	15.46	0.98
22	Stagg,2800,Dyn,Orig	16.19	16.25	1.08
23	Stagg,2800,Sta,Modi	15.95	15.47	1.20
24	Stagg,2800,Sta,Orig	14.86	16.13	0.40
25	No Sea,2100,Dyn,Modi	11.34	11.30	0.71
26	No Sea,2100,Dyn,Orig	11.51	11.47	0.82
27	No Sea,2100,Sta,Modi	11.49	11.35	0.75
28	No Sea,2100,Sta,Orig	10.24	11.44	0.14
29	No Sea,2500,Dyn,Modi	13.75	13.71	0.82
30	No Sea,2500,Dyn,Orig	13.85	13.83	0.91
31	No Sea,2500,Sta,Modi	13.93	13.77	0.87
32	No Sea,2500,Sta,Orig	12.41	13.78	0.17
33	No Sea,2800,Dyn,Modi	15.99	15.93	0.95
34	No Sea,2800,Dyn,Orig	16.18	16.14	1.02
35	No Sea,2800,Sta,Modi	16.10	16.01	0.93
36	No Sea,2800,Sta,Orig	14.50	16.09	0.17

Table 6.2 Results for Product P1 only

Exp	Conditions	ESFT	FLOW	MT
1	Synch,2100,Dyn,Modi	13.19	13.08	0.82
2	Synch,2100,Dyn,Orig	11.38	11.35	0.71
3	Synch,2100,Sta,Modi	13.37	13.15	1.31
4	Synch,2100,Sta,Orig	12.43	11.35	1.53
5	Synch,2500,Dyn,Modi	16.52	16.21	1.23
6	Synch,2500,Dyn,Orig	14.20	14.09	0.93
7	Synch,2500,Sta,Modi	16.22	16.46	1.57
8	Synch,2500,Sta,Orig	15.28	14.26	1.77
9	Synch,2800,Dyn,Modi	20.34	18.83	2.64
10	Synch,2800,Dyn,Orig	18.10	17.45	1.94
11	Synch,2800,Sta,Modi	19.18	20.27	2.02
12	Synch,2800,Sta,Orig	17.11	18.73	1.92
13	Stagg,2100,Dyn,Modi	12.75	12.77	0.76
14	Stagg,2100,Dyn,Orig	10.82	10.87	0.65
15	Stagg,2100,Sta,Modi	13.08	12.76	1.13
16	Stagg,2100,Sta,Orig	12.60	10.80	1.85
17	Stagg,2500,Dyn,Modi	16.47	16.19	1.16
18	Stagg,2500,Dyn,Orig	14.26	14.14	0.99
19	Stagg,2500,Sta,Modi	16.38	16.41	1.74
20	Stagg,2500,Sta,Orig	15.25	14.31	1.72
21	Stagg,2800,Dyn,Modi	18.51	18.23	1.27
22	Stagg,2800,Dyn,Orig	15.70	15.79	1.10
23	Stagg,2800,Sta,Modi	18.13	18.52	1.12
24	Stagg,2800,Sta,Orig	17.37	15.69	1.86
25	No Sea,2100,Dyn,Modi	13.19	13.09	0.81
26	No Sea,2100,Dyn,Orig	11.34	11.30	0.70
27	No Sea,2100,Sta,Modi	13.22	13.14	0.82
28	No Sea,2100,Sta,Orig	12.36	11.29	1.19
29	No Sea,2500,Dyn,Modi	16.09	15.96	1.00
30	No Sea,2500,Dyn,Orig	13.71	13.69	0.86
31	No Sea,2500,Sta,Modi	16.04	16.04	0.97
32	No Sea,2500,Sta,Orig	14.90	13.68	1.37
33	No Sea,2800,Dyn,Modi	18.93	18.71	1.27
34	No Sea,2800,Dyn,Orig	16.00	15.97	0.97
35	No Sea,2800,Sta,Modi	18.33	18.88	0.90
36	No Sea,2800,Sta,Orig	17.07	15.94	1.36

Table 6.3 Results for Product P2 only

Several plots to illustrate typical performance over time are shown in figures 6.1 through 6.5. The data in these plots is for all products on the shop floor so the lines are a little 'fuzzy'. In each case the (a) graph represents dynamic lead time setting and the (b) graph represents static lead time setting. The plots are over a four year period following system stabilisation. Figures 6.1 through 6.4 are meant to show dynamic lead time setting responding to fluctuations in shop loading. Therefore, the data is from an experiment where seasonality is present. The seasonality in the demand is seen in the flowtime plots (figure 6.1). The response of the system when dynamic lead times are in use is evident from figures 6.2 through 6.4. This is in stark contrast to the sharp peaks in lateness and mean tardiness when static lead times are in use. The plots in figures 6.5 (a) and (b) are from an experiment where there is no seasonality in the demand. The flowtimes fluctuate randomly.

The experimental factors identified in section 5.2 all affected the results. Higher shop loading led to an increase in mean tardiness levels when seasonality was present. With no seasonality, the numbers are very close. This suggests an interaction effect between seasonality and shop loading.

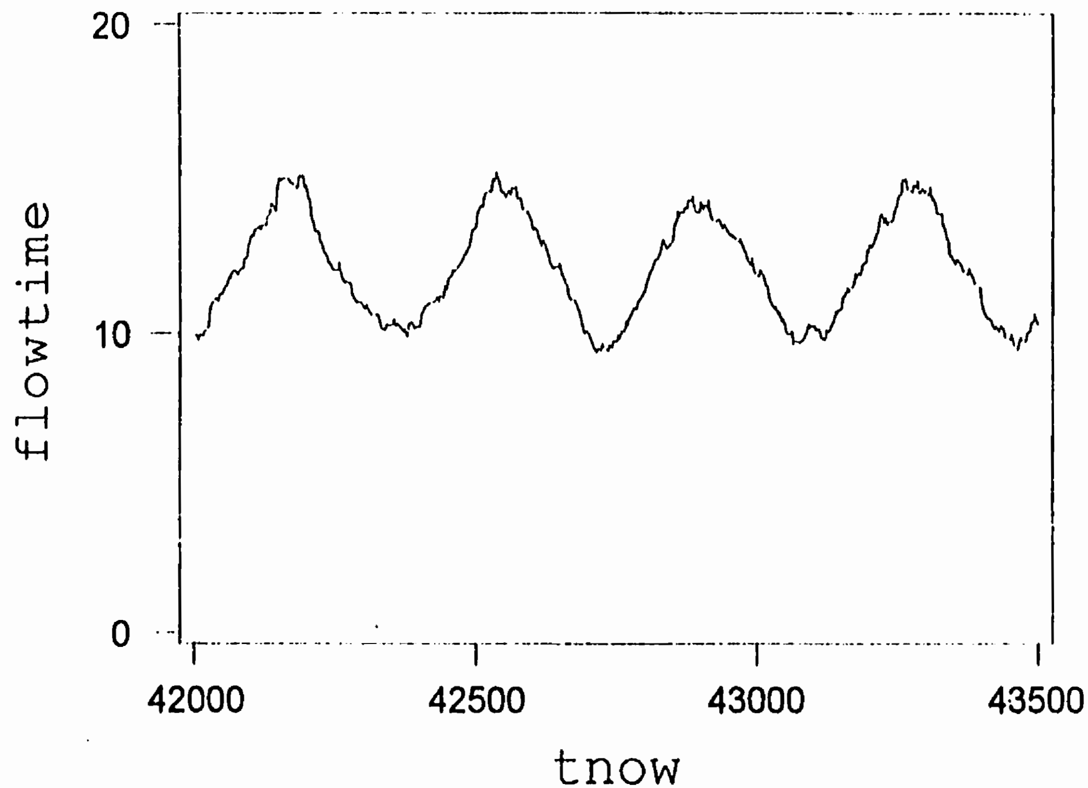


Figure 6.1(a) Flowtime plot (dynamic lead times)

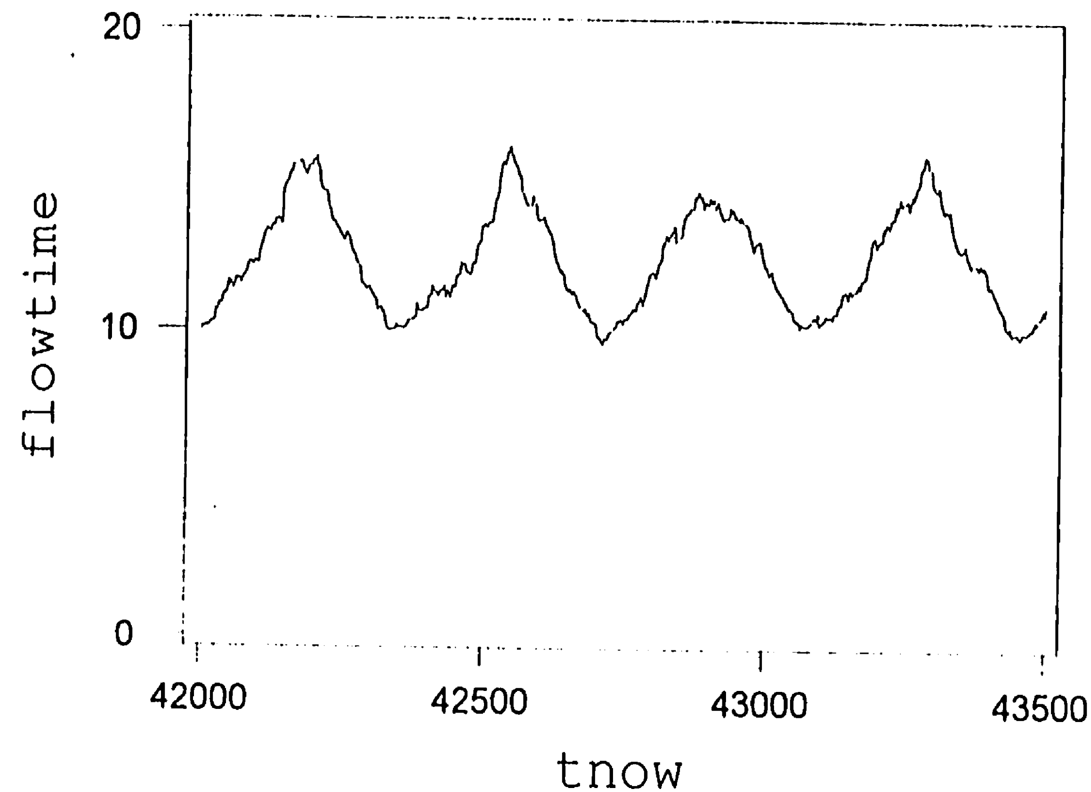


Figure 6.1(b) Flowtime plot (static lead times)

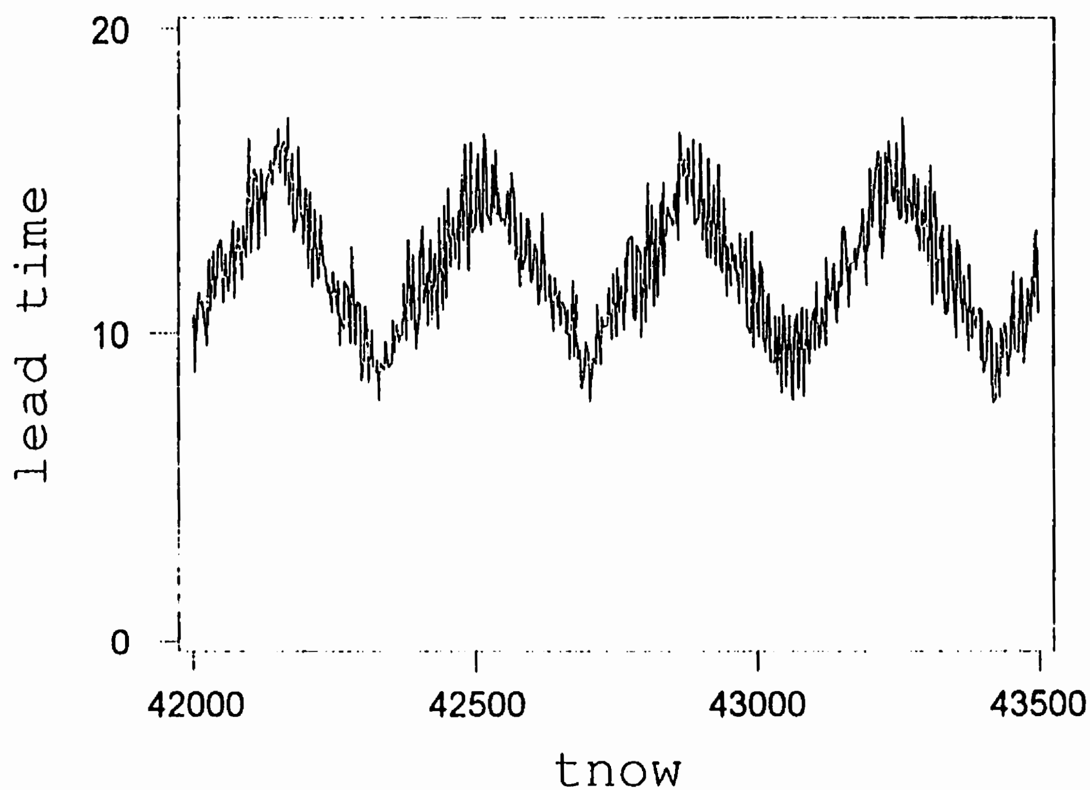


Figure 6.2(a) Product lead times (dynamic lead times)

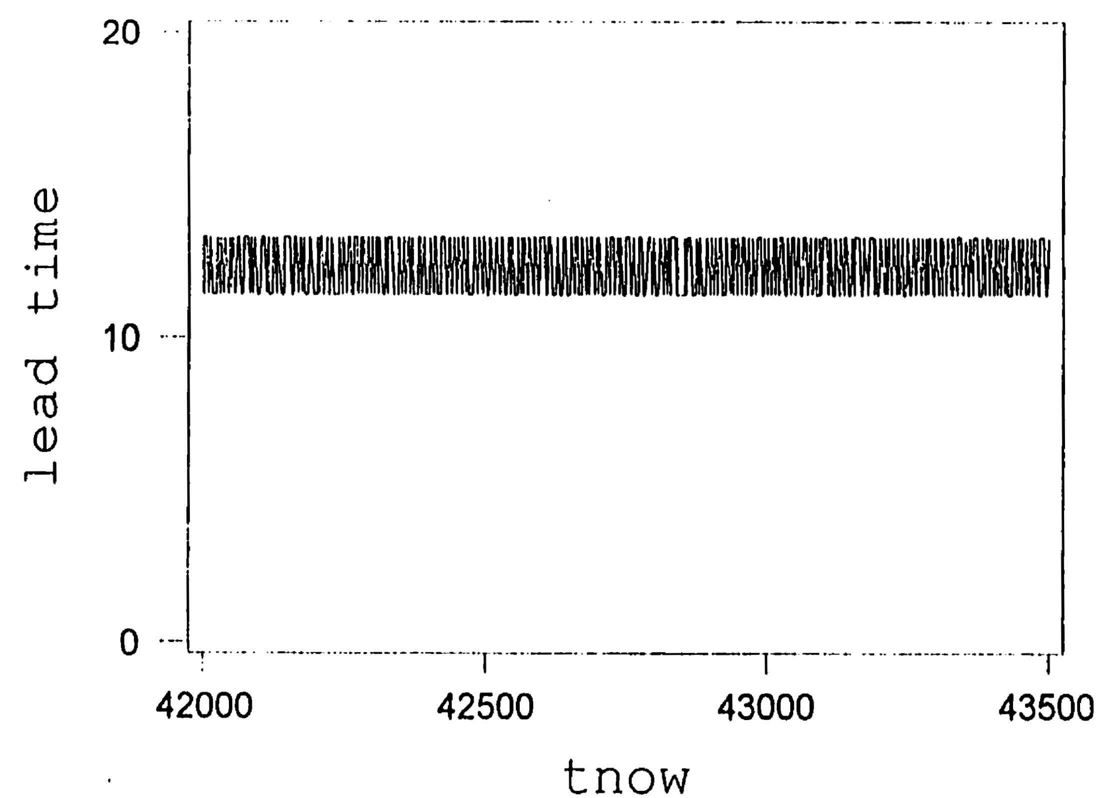


Figure 6.2(b) Product lead times (static lead times)

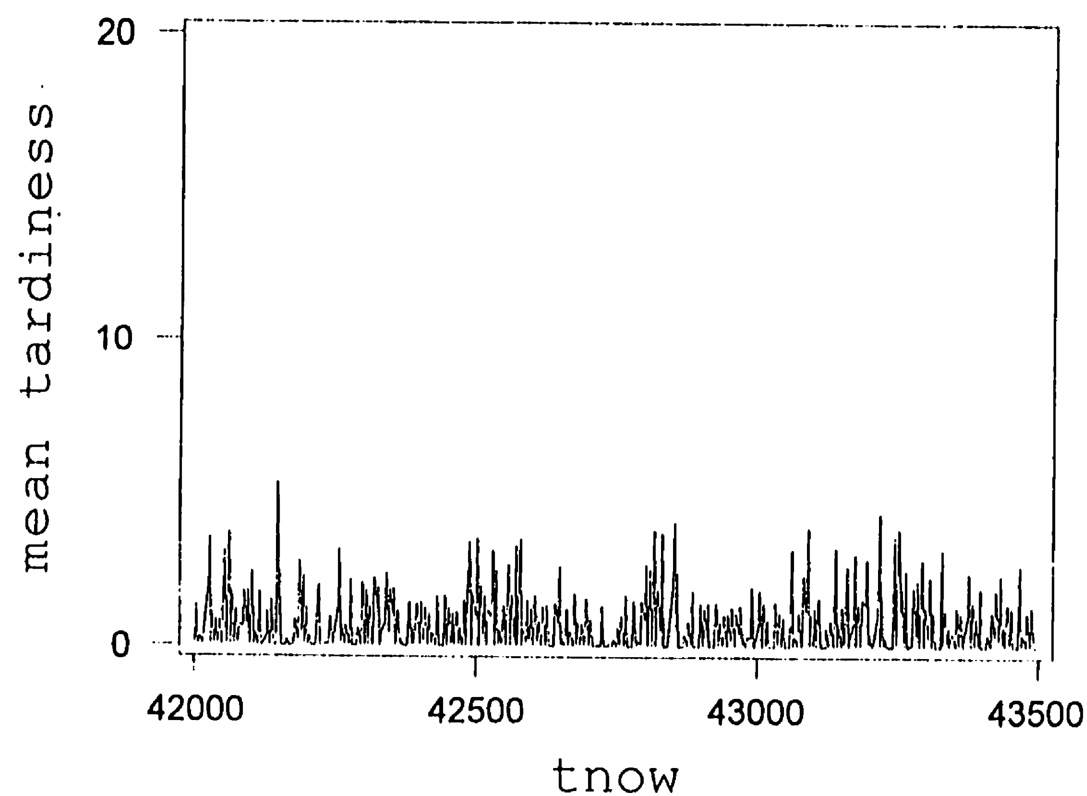


Figure 6.3(a) Mean tardiness (dynamic lead times)

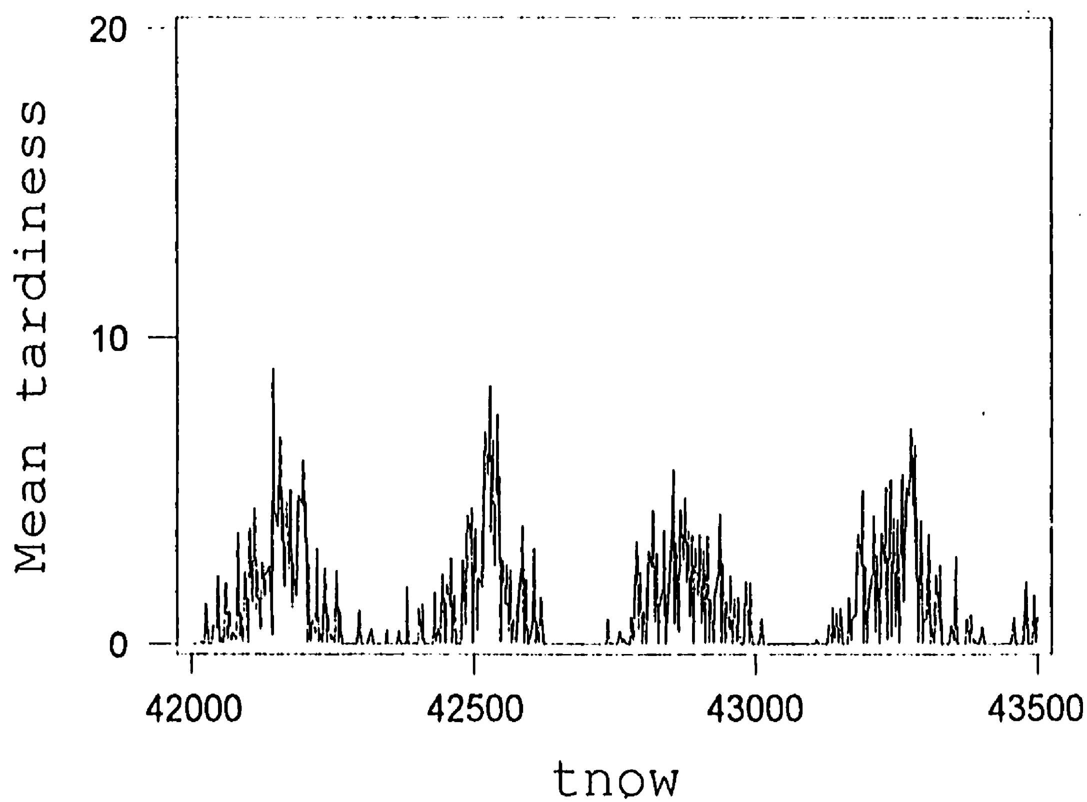


Figure 6.3(b) Mean tardiness (static lead times)

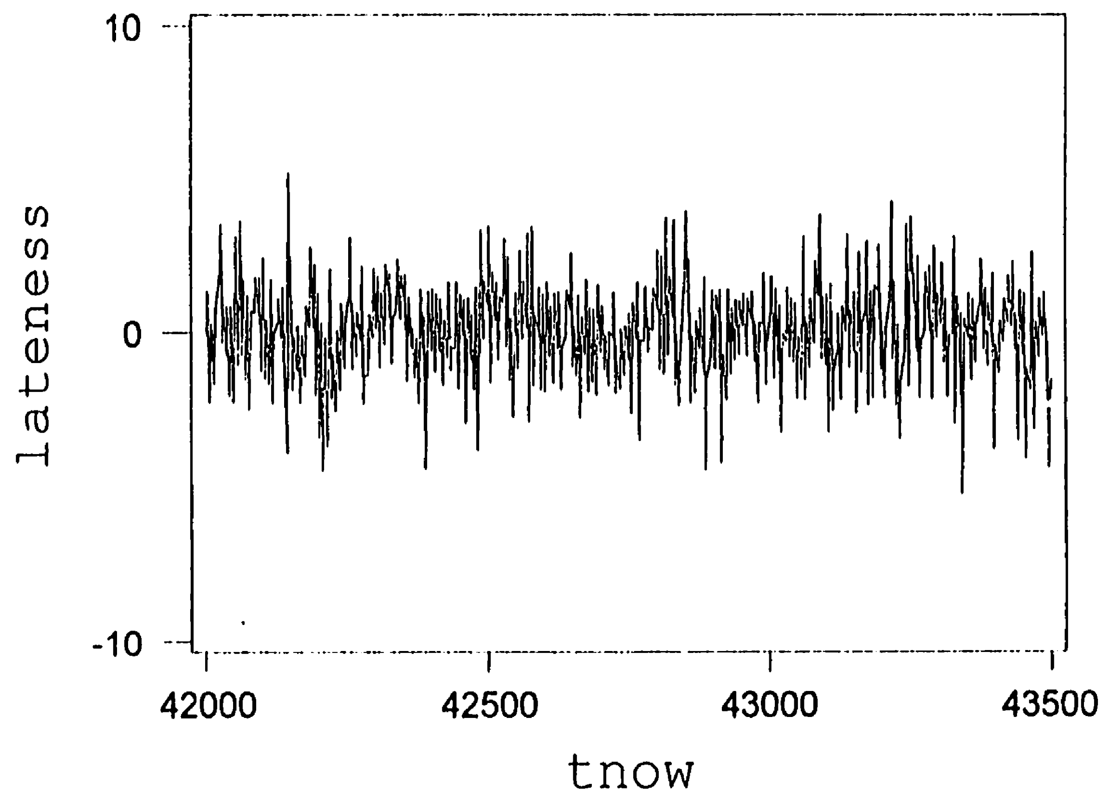


Figure 6.4(a) Product lateness (dynamic lead times)

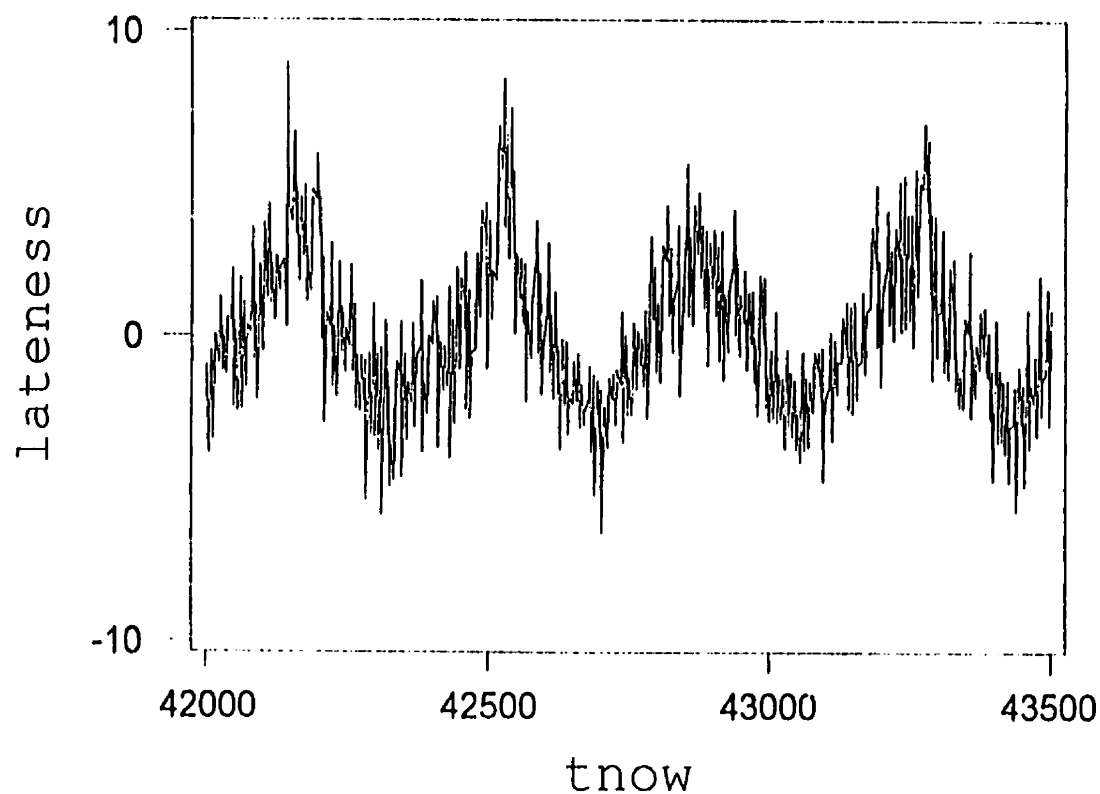


Figure 6.4(b) Product lateness (static lead times)

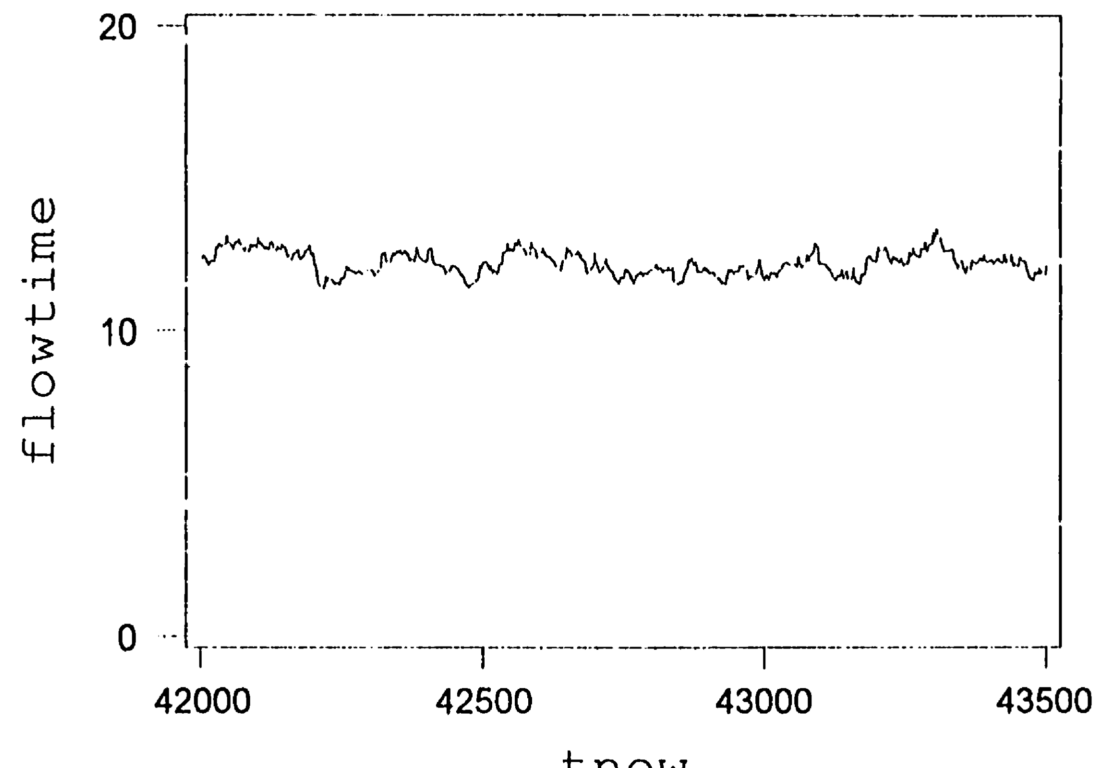


Figure 6.5(a) Flowtime (no seasonality,dynamic lead times)

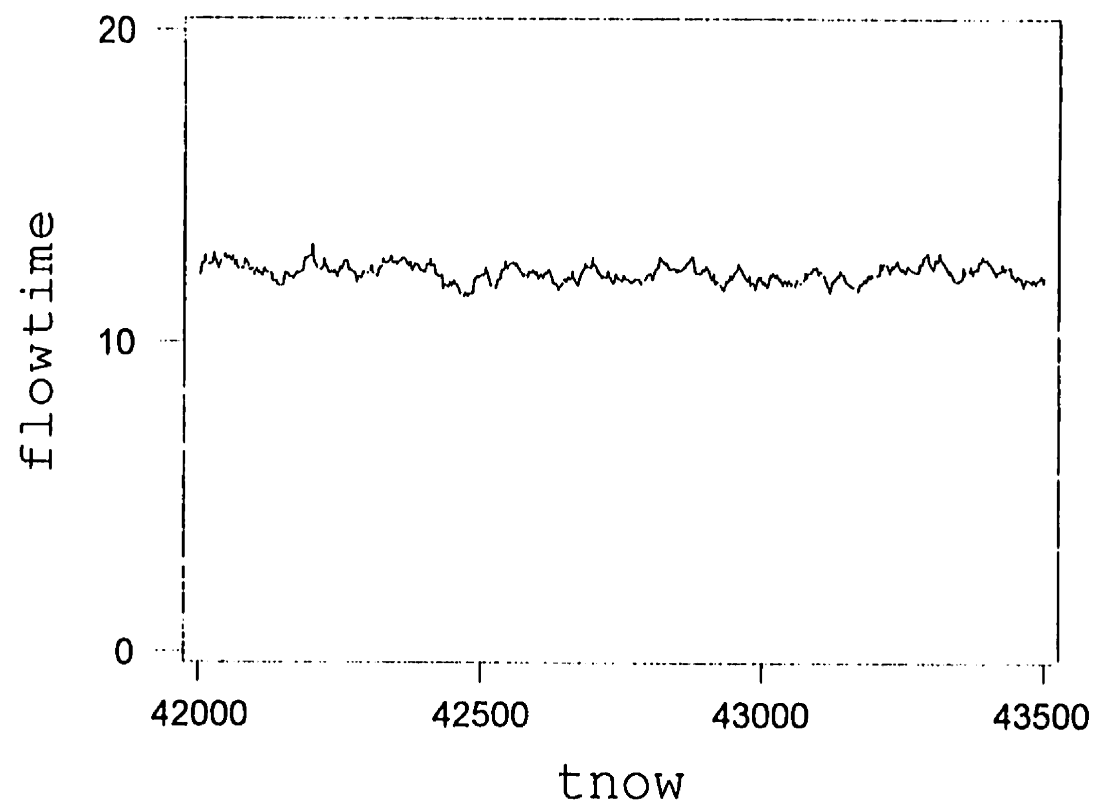


Figure 6.5(b) Flowtime (no seasonality, static lead times)

The benefits of dynamic lead times are best seen when demand fluctuates (seasonality is present). In these instances, dynamic lead times adjust to changing conditions and help improve MRP output validity. When static lead times are used, they are very rarely correct (4 or 5 weeks of the year) hence release dates are set incorrectly most of the time. When there is no seasonality in the demand, adjustments in the lead times are not really necessary (if they were set correctly to start with!).

In this case static lead times are only set after the dynamic experiments have been run. This means **very** good estimates (based on actual production runs) are used to set static lead times. This is not usually the case in actual manufacturing practice. The result is that when there is no seasonality present in the demand, dynamic lead times, at best, offer no advantage over static lead times. In many experiments, static lead times yielded the better delivery performance.

It is rather difficult (and perhaps unfair) to compare the two product structures, since the work and assembly content are different. However, the results described above generally hold true for both product sets.

Comparisons between dynamic and static lead time setting mechanisms on the basis of mean tardiness levels should only be carried out under conditions of equal average lead times (equal flow allowances). Since the static lead times were set equal to the average dynamic lead times, good control was maintained over flow allowances, as seen in Table 6.1

6.2 Analysis of Results

This section presents output from the statistical tests and an analysis of what the results actually show. First the results are displayed. Detailed discussions then follow.

6.2.1 Statistical Test Results

The first test done is a correlation test to check that the method of data truncation does not violate independence assumptions. This test is carried out in the OUTPT processor using the CORRELOGRAM command (Systems Modelling, 1989). It is carried out on the flowtime data collected in each experiment. The largest lag for which correlation remains significant is recorded in table 6.4. Sample CORRELOGRAM output is given in Appendix 7.

Experiment Number	Largest Lag for which Correlation is Significant	Truncation Method
1	26 weeks	OK
2	26 weeks	OK
3	26 weeks	OK
4	26 weeks	OK
5	26 weeks	OK
6	26 weeks	OK
7	26 weeks	OK
8	26 weeks	OK
9	28 weeks	OK
10	27 weeks	OK
11	27 weeks	OK
12	26 weeks	OK
13	30 weeks	OK
14	24 weeks	OK
15	35 weeks	OK
16	25 weeks	OK
17	26 weeks	OK
18	26 weeks	OK
19	26 weeks	OK
20	26 weeks	OK
21	33 weeks	OK
22	26 weeks	OK
23	29 weeks	OK
24	23 weeks	OK
25	31 weeks	OK
26	31 weeks	OK
27	27 weeks	OK
28	21 weeks	OK
29	31 weeks	OK
30	39 weeks	OK
31	33 weeks	OK
32	39 weeks	OK
33	27 weeks	OK
34	48 weeks	OK
35	33 weeks	OK
36	30 weeks	OK

Table 6.4 Summary of results of correlation test

Having ascertained that the truncation method is acceptable, the mean tardiness data is truncated in the MINITAB program. Macros written to perform the truncation are included in Appendix 4. The truncated data for experiments where shop load, demand pattern and products is identical is then subjected to a paired-t test. Hence mean tardiness performance with dynamic lead times are compared against those with static lead times under similar conditions. The null hypothesis is H_0 and the alternative hypothesis is H_A. These are shown below, with μ being the mean tardiness.

$$H_0: \mu_1 = \mu_2$$

$$H_A: \mu_1 \neq \mu_2$$

The result of the paired-t test is a confidence interval on the difference in means of the two sets of data (e.g -0.285, -0.277). If this confidence interval (CI) contains zero, the null hypothesis cannot be rejected. If the CI misses zero, H_0 is rejected. If the CI misses zero and is positive, the static lead times are better than the dynamic lead times for those conditions. If the CI misses zero and is negative, the dynamical lead times are better than the static lead times for those conditions. Appendix 8 contains sample output from the paired-t tests carried out, as well as a normal probability plot indicating that the differences are normally distributed.

6.2.2 Analysis

The results of the correlation tests (table 6.4) indicate that when seasonality is present, the system gets the chance once per year to catch up *if necessary*. This fits in with expectations. During the low demand season there should be very little queueing activity in the shop, hence little correlation with observations when shop loads were higher. When there is no seasonality present the correlation results are a little higher. Again this is expected. There is no low season for the shop floor to catch up but no peak season either to load up the shop in the first place. The maximum value from all the correlation tests is from experiment 34, where the value is still less than one year. This indicates the truncation method outlined in section 5.5.1 is acceptable.

The utilisation values should be the same for experiments whose average shop load is the same. The results in Table 6.1 show that this is the case. Since the shop floor is being fed the same size orders on average, the utilisations should be the same. Another measure used to check the behaviour of the system is lateness. The values in Table 6.1 are average lateness for all products on the shop floor. They are monitored to check for bias. Large positive or negative values indicate that the MRP is biased in its setting of release and due dates. The values in Table 6.1 indicate that there is no bias in the system, since all the lateness values are very close to zero and there are no safety allowances. The percent tardy (PT) results also support this conclusion, as they are tightly scattered about a mean of 49.7% for all experiments. In other words the use of exponentially smoothed flowtime feedback provides unbiased flowtime prediction results on average.

The data on mean tardiness is truncated and paired-t tests carried out as explained in section 6.2.1. Table 6.5 summarises the results of the tests. 'Reject H_o' indicates the means are not equal at the 95% confidence level. 'Fail to reject H_o' indicates the means are equal at the 95% confidence level. Positive confidence interval (CI) values signify that static lead times outperformed dynamic lead times, and negative CI values vice versa.

When there is seasonality in the demand and the bottleneck is not overloaded in the peak season, the use of dynamically set lead times improves due date performance significantly. This is as expected, since the flow allowances (lead times) are being adjusted to fluctuations in demand. When there is no

Experiments Compared	Confidence Interval on Difference in Means	Test Result
01 vs. 03	(-0.430, -0.277)	Reject Ho
02 vs. 04	(-0.285, -0.132)	Reject Ho
05 vs. 07	(-0.322, -0.003)	Reject Ho
06 vs. 08	(-0.310, -0.060)	Reject Ho
09 vs. 11	(-0.243, 0.858)	Fail to reject Ho
10 vs. 12	(-0.563, 0.765)	Fail to reject Ho
13 vs. 15	(-0.405, -0.280)	Reject Ho
14 vs. 16	(-0.485, -0.367)	Reject Ho
17 vs. 19	(-0.556, -0.244)	Reject Ho
18 vs. 20	(-0.214, 0.004)	Fail to reject Ho
21 vs. 23	(-0.060, 0.116)	Fail to reject Ho
22 vs. 24	(-0.130, 0.104)	Fail to reject Ho
25 vs. 27	(-0.059, 0.023)	Fail to reject Ho
26 vs. 28	(0.032, 0.169)	Reject Ho
29 vs. 31	(-0.117, 0.074)	Fail to reject Ho
30 vs. 32	(-0.047, 0.187)	Fail to reject Ho
33 vs. 35	(0.075, 0.269)	Reject Ho
34 vs. 36	(0.116, 0.291)	Reject Ho

Table 6.5 Summary of results of paired-t tests

seasonality, and any fluctuation is random, the use of dynamically set lead times either offers no improvement in due date performance or this performance deteriorates. This can be partially explained as the MRP is continually adjusting flow allowances for minor (& random) fluctuations in demand when these allowances should be left alone. This is somewhat analogous to nervousness in the MRP. The demand in the 'no seasonality' experiments does not vary greatly (coefficient of variation < 0.05). This is in contrast to the processing times where the CV is set at 0.3. It could also be noted that not only is there no seasonality in demand, but there is also no trend. Dynamic lead time setting would also be expected to work better than static lead times if a trend is present.

At heavy shop loading levels (2800 units per week per product), dynamically set lead times do not improve due date performance at all, and when seasonality is removed, due date performance deteriorates. At this loading level, work builds up at the bottleneck in the peak seasons. Batch flowtimes more than double at this time of the year. The input rate of orders into the shop floor does not alter, only the order size gets bigger. Assuming processing times to be approximately the same, this means that for every order that finishes an operation and feeds back part flowtime data to the MRP, at least two more orders will have been released by the MRP with release and due dates calculated using the old lead times. Hence there is a significant lag in adjustments to rapidly changing shop loads. Moreover, the jump in demand (25% in all cases) is higher at the heavy loading levels, since 25% of 2800 units is greater than 25% of 2100 units.

Finally, it is observed from table 6.6 that dynamically setting lead times results in quite a large reduction in lateness variance when seasonality in demand is present. When demand does not fluctuate with seasonality, the values for lateness variance are much closer. These results tend to agree with results published by Bertrand (1983). The reduction in variance can lead to smaller safety allowances being added on to lead times, making them shorter overall.

Exp	Conditions	Mean	Std. Dev.
1	Synch,2100,Dyn,Modi	0.08	1.79
2	Synch,2100,Dyn,Orig	0.04	1.80
3	Synch,2100,Sta,Modi	0.19	2.61
4	Synch,2100,Sta,Orig	-0.05	2.43
5	Synch,2500,Dyn,Modi	0.24	2.58
6	Synch,2500,Dyn,Orig	0.12	2.26
7	Synch,2500,Sta,Modi	-0.08	3.59
8	Synch,2500,Sta,Orig	0.25	3.20
9	Synch,2800,Dyn,Modi	1.10	4.48
10	Synch,2800,Dyn,Orig	0.63	3.98
11	Synch,2800,Sta,Modi	-0.82	5.27
12	Synch,2800,Sta,Orig	-0.62	5.19
13	Stagg,2100,Dyn,Modi	-0.01	1.79
14	Stagg,2100,Dyn,Orig	-0.05	1.78
15	Stagg,2100,Sta,Modi	0.28	2.17
16	Stagg,2100,Sta,Orig	0.47	2.06
17	Stagg,2500,Dyn,Modi	0.22	2.37
18	Stagg,2500,Dyn,Orig	0.14	2.37
19	Stagg,2500,Sta,Modi	0.11	3.70
20	Stagg,2500,Sta,Orig	-0.30	3.14
21	Stagg,2800,Dyn,Modi	0.15	2.62
22	Stagg,2800,Dyn,Orig	-0.05	2.84
23	Stagg,2800,Sta,Modi	0.07	2.76
24	Stagg,2800,Sta,Orig	0.23	2.49
25	No Sea,2100,Dyn,Modi	0.08	1.77
26	No Sea,2100,Dyn,Orig	0.04	1.80
27	No Sea,2100,Sta,Modi	0.11	1.78
28	No Sea,2100,Sta,Orig	-0.06	1.69
29	No Sea,2500,Dyn,Modi	0.09	2.14
30	No Sea,2500,Dyn,Orig	0.04	2.11
31	No Sea,2500,Sta,Modi	0.10	2.16
32	No Sea,2500,Sta,Orig	-0.07	1.96
33	No Sea,2800,Dyn,Modi	0.17	2.56
34	No Sea,2800,Dyn,Orig	0.06	2.39
35	No Sea,2800,Sta,Modi	-0.21	2.51
36	No Sea,2800,Sta,Orig	-0.21	2.13

Table 6.6 Product lateness

Chapter Seven

CONCLUSIONS

This thesis examines the concept of setting planned lead times dynamically in MRP systems. The objective of this chapter is to provide a summary of findings and comment on possible future work in this area.

7.1 Summary

An introduction to the research topic is given in Chapter one. Chapter two provides a survey of issues related to this thesis. MRP and other production planning and control systems are reviewed. Their advantages and limitations are identified. In Chapter three the production environment assumed in this research is defined. Development of the MRP-simulation interface is described in Chapter four together with verification and validation efforts. Chapter five discusses experimental factors, performance measures, operation strategy, and defines an experimental plan.

Results and analysis are presented in Chapter six. The results are only partly as expected. Dynamically set lead times improve due date performance when demand fluctuates seasonally. Due date performance generally deteriorates with dynamically set lead times when demand fluctuation is very small. This could be interpreted as the MRP adjusting lead times when it should not, since shop loads are more or less constant.

As the shop load is increased, the system's ability to respond to changes decreases. This is unexpected, but the deviation from expectations is explained. At the highest loading level, the system is tested under extreme conditions. No satisfactory solution has been found for running an overloaded system. By loading the bottleneck machine beyond capacity, the system is being pushed to instability.

In this research it has been demonstrated that dynamic lead time setting is beneficial under many conditions normally encountered in batch manufacturing. This has been made possible by linking a commercial MRP system with the simulation of a production facility; something that has not knowingly been previously accomplished. Some features of the MRP system are bypassed. Nevertheless, it is now possible to enhance the whole system by reintroducing some of these features and making the simulation more realistic.

The operation strategy used to run the experiments could itself prove useful as a means of determining lead times in MRP systems. A dynamic experiment is run which yields flowtime estimates. These estimates can then be used to set lead times. Such a method of determining planned lead times could be a vast improvement over some methods currently in use.

7.2 Future Work

The work done for this thesis is a good starting point, yet there remains much to be done in this area of research. A natural extension of the work in this thesis is to add more real world features to the MRP and the simulation. Such features include but are not limited to commonality of parts, different batch sizing logic, order review, more fluctuation in demand, more products, scrap and breakdowns.

Some changes should be made to the experimental system. The system's response to changes must be improved by using a different feedback mechanism or more sophisticated flowtime prediction relationships. Including current queue information in the data fed back to the MRP could make the system more responsive. Several existing flowtime prediction relationships may be tried. Operation due dates should also be used throughout the simulation model. This would make the introduction of lot sizing techniques other than lot-for-lot possible.

One of the biggest research limitations of the system developed is its overall execution speed. Each experiment typically took 1-2 days to run on a '486' personal computer. There are several ways to try overcoming this problem, some of which involve major changes. It is possible another MRP system (written in C or FORTRAN) could be obtained with the source code. This system would be launched from SIMAN V through a user-coded function. Another method, which is favoured, is to re-write the simulation model in a package written for the Windows95™ operating system. Arena 3.0 (which includes the SIMAN language), recently introduced by Systems Modelling Corporation, is one such package. Arena 3.0 is capable of linking into spreadsheets through the use of Visual Basic for Applications (VBA) technology. The direct linkage of a spreadsheet-based MRP package to the simulation model would eliminate the need to restart the simulation after every MRP regeneration and would make existing SIMAN output processor tools more useful. Macros currently written in LOTUS™ 1-2-3® would need to be re-written in the VBA language to accommodate this change.

Bibliography

R.B.Altman et al. (1993) *Using 1-2-3 for Windows Release 4* Que Corporation.

K.R.Baker (1984) Sequencing rules and due-date assignments in a job shop *Management Science,* Vol.30, No.9, p1093.

W.Bechte (1988) Theory and Practice of load-oriented manufacturing control *International Journal of Production Research,* 26,3,375-395

D.D.Bedworth, M.R.Henderson, P.M.Wolfe (1991) *Computer-integrated design and manufacturing* McGraw-Hill, New York

J.M.W.Bertrand (1983) The use of workload information to control job lateness in controlled and uncontrolled release production systems. *Journal of Operations Management,*3,79-92.

J.Browne, J.Harhen, J.Shivnan (1996) *Production Management Systems: An Integrated Perspective, 2nd Ed.* Addison-Wesley Publishing Co.

F.R.Chang (1996) Heuristics for dynamic job shop scheduling with real-time updated queuing time estimates *1996 IERC proceedings*, Minneapolis Conference.

T.C.E.Cheng, M.C.Gupta (1989) Survey of scheduling research involving due date determination decisions *European Journal of Operations Research*,38,156-166.

R.W.Conway, W.L.Maxwell, L.W.Miller (1967) *Theory of scheduling* Addison-Wesley Publishing Co.

R.E.Devor, T.Chang, J.W.Sutherland (1992) *Statistical Quality Design and Control* MacMillan Publishing Co.

S.T.Enns (1993) Job shop flowtime prediction and tardiness control using queueing analysis *International Journal of Production Research*,Vol.31, No.9, p2045.

S.T.Enns (1995a) *Scheduling in MRP environments - an analysis of issues and trends* Working Paper WP94-17, University of Calgary.

S.T.Enns (1995b) *An adaptive forecasting model for job shop flowtime prediction* Working Paper WP94-39, University of

Calgary.

S.T.Enns (1995c) *An adaptive forecasting model for flowtime prediction of assembled products* Working Paper WP94-46, University of Calgary.

S.T.Enns (1995e) Finite Scheduling: Performance comparisons when due dates are internally set *Production & Inventory management Journal*, 3rd Qtr, pp5-10.

S.T.Enns (1996a) Analysis of a process improvement path using rapid modelling *Total Quality Management*, Vol 7,3, 1996, p283.

S.T.Enns (1996b) Finite capacity systems: Performance issues and comparisons *Computers in Industrial Engineering*, Vol.30, No.4, pp727-739.

D.W.Fogarty, J.H.Blackstone Jr., T.R.Hoffman (1991) *Production and Inventory Management, 2D Ed.* South-Western / APICS.

T.D.Fry, P.R.Philipoom, R.E.Markland (1989) Due date assignment in a multi-stage job shop *IIE Transactions*, June, p153.

E.M.Goldratt, J.Cox (1992) *The Goal: A process of ongoing improvement 2nd ed.* Croton-on-Hudson, NY: North River Press

J.S.Goodwin, J.C.Goodwin (1982) Operating policies for scheduling assembled products *Decision Sciences*, Vol.13, p585-603.

J.H.Hoyt (1982) Determining dymnamic lead times for manufactured parts in a job shop *Computers in Manufacturing Execution and Control Systems*, Auerbach Publishers Inc, NJ.

J.H.Hoyt (1978) Dynamic lead times that fit today's dynamic planning *Production and Inventory Management*, 1st Qtr, p63.

J.J.Kanet, J.C.Hayya (1982) Priority dispatching with operation due dates in a job shop *Journal of Operations Management*, 2,3, May 1982, 155-163.

P.T.Kidd (1994) *Agile manufacturing: forging new frontiers* Addison-Wesley Publishing Company

A.M.Law, W.D.Kelton (1991) *Simulation modelling and analysis, 2nd Ed* McGraw-Hill International Editions

S.R.Lawrence (1995) Estimating flowt
in complex production systems *IIE Tr*

T.Lunn, S.A.Neff (1992) *MRP: Integra*
planning & modern business Business

S.A.Melnyk, C.J.Piper (1985) Leadti
sizing effect International
Research,Vol.23, No.2, p253.

Minitab Inc. (1993) *Minitab referenc*
College, PA.

Network Dynamics (1991) *MPX user's m*
Ma, Network Dynamics Inc.

H.W.Oden, G.A.Langenwalter, R.A.Lu
material and capacity requirements

L.Onur, W.J.Fabrycky (1987) An input/
the dynamic job-shop *IIE Tranaction*

J.Orlicky (1975) *Material Requireme*
Book Company.

C.D.Pegden, R.E.Shannon, R.P.Sadowski (1990) *Introduction to simulation using SIMAN* McGraw-Hill Inc.

C.J.Piper, R.Kuik (1988) *MRP with explicit production rates and continuous delivery* Working Paper 88-32, October 1988, University of Western Ontario

G.L.Ragatz, V.A.Mabert (1984) A simulation analysis of due date assignment rules *Journal of Operations Management*, Vol.5, No.1, p27-39.

R.Ramasesh (1990) Dynamic job shop scheduling: A survey of simulation research *International Journal of Management Science*, Vol.18, No.1, p43-57

P.I.Roder Finite scheduling systems *APICS - The Performance Advantage*, August, pp40-42.

M.L.Spearman, W.J.Hopp, D.L.Woodruff (1990a) CONWIP: A pull alternative to Kanban *International Journal of Production Research*, Vol.28, No.5, p879.

M.L.Spearman, W.J.Hopp, D.L.Woodruff (1990b) Practical strategies for lead time reduction *Manufacturing Review*, Vol.3, No.2, June.

R.Suri, S.de Treville (1991) Full speed ahead: A look at rapid modelling technology in operations management *OR/MS Today*, June, pp34-42.

Systems Modelling Corporation (1989) *The SIMAN IV reference guide*, Sewickley, PA

User Solutions Inc.(1996) *Spreadsheet Resource Manager ver.3*, South Lyon, MI.

M.M.Vig, K.J.Dooley (1993) Mixing static and dynamic flowtime estimates for due-date assignment *Journal of Operations Management*, 11 (1993) 67-79

T.E.Vollmann, W.L.Berry, D.C.Whybark (1992) *Manufacturing Planning and Control Systems, 3rd Ed.* Richard D. Irwin, Inc.

E.F.Watson, D.J.Medeiros, R.P.Sadowski (1993) Generating Component Release Plans with Backward Simulation *Proceedings, 1993 Winter Simulation Conference*, pp930-938 G.W.Evans, M.Mollaghasemi, E.C.Russel, W.E.Biles (eds.)

H.J.Watson, J.H.Blackstone,Jr. (1989) Computer Simulation 2nd Ed., John Wiley, New York.

W.Whitt (1983) The queuing network analyser *The Bell Systems Technical Journal*, Vol 62, 2779-2815.

D.C.Whybark, J.G.Williams (1976) Material requirements planning with uncertainty *Decision Sciences*, Vol.7, p595.

H.P.Wiendahl (1995) *Load-oriented manufacturing control* Springer-Verlag, New York

O.Wight (1970) Input/Output control - a real handle on lead time *Production and Inventory Management*, 3rd Qtr.

R.L.Winkler, W.L.Hayes (1975) *Statistics: Probability, inference, and decision, 2nd Ed.* Holt Rinehart & Winston, NY.

F.P.Wyman (1993) Thinking about simulation-based scheduling? Here's what you ought to know *APICS - The Performance Advantage*, November, pp57-61.

Table of Appendices

Appendix 1


```
BEGIN,yes,Modified;
;
;       ATTRIBUTES (also used in ExitSystem stn as temp vars)
;       A(1) - Job Number
;       A(2) - Part Type
;       A(3) - Batch Size
;       A(4) - Component A Release Date
;       A(5) - Component Due Date (ESFT all types)
;       A(6) - Time In at Shop
;       A(7) - Time In at Station (ESFT by type)
;       A(8) - End Item Due Date
;       A(9) - Tardiness Attribute
;       A(10) - Component B Release Date
;
;
;       VARIABLES
;       X(1) - Total Number of Batches Completed
;       X(2) - Number of Batches Completed THIS Week
;       X(3) - Number of Batches Completed by LAST Week
;       X(4) - Job Number Index
;       X(5) - Parts WIP P1 C1
;       X(6) - Number of Tardy Jobs
;       X(7) - Temp Variable
;       X(8) -
;       X(9) -
;       X(10) - Which file to read P1 jobs from
;       X(11) - Which file to read P2 jobs from
;       X(12) - LT temp variable (used at all stns)
;       X(13) -
;       X(14) - Smoothed Adj FT per part at Saw Stn (P2)
;       X(15) - Smoothed Adj FT per part at Shear Stn (P1)
;       X(16) -
;       X(17) - Smoothed Adj FT per part at Brake Stn (P1)
;       X(18) - Smoothed Adj FT per part at Brake Stn (P2)
;       X(19) - Smoothed Adj FT per part at Punch Stn (P1)
;       X(20) - Smoothed Adj FT per part at Punch Stn (P2)
;       X(21) - Smoothed Adj FT per part at Weld Stn (P1)
;       X(22) - Smoothed Adj FT per part at Weld Stn (P2)
;       X(23) - Smoothed Adj FT per part at Paint Stn (P1)
;       X(24) - Smoothed Adj FT per part at Paint Stn (P2)
;       X(25) - Smoothed Adj FT per part at Pack Stn (P1)
;       X(26) - Smoothed Adj FT per part at Pack Stn (P2)
;       X(27) - Smoothed Adj Overall Batch FT for P1
;       X(28) - Smoothed Adj Overall Batch FT for P2
;       X(29) - Smoothed FT for all jobs (P1 & P2)
;       X(30) - Parts WIP P1 C2
;       X(31) - Parts WIP P2
;       X(32) -
;       Stup - Setup Time (defined in an array)
;       Mean - Process Mean (defined in an array)
;       Stdv - Process Standard Deviation (defined in an array)

;   Initialise Global Variables

        CREATE;
        ASSIGN:        X(4)=10:
                       X(5)=6000:
                       X(10)=1:
                       X(11)=2:
                       X(14)=0.0010662:
                       X(15)=0.0011303:
                       X(17)=0.00117965:
                       X(18)=0.00132345:
                       X(19)=0.0010228:
                       X(20)=0.00110015:
                       X(21)=0.00120295:
                       X(22)=0.0012858:
```

```
                        X(23)=0.00081345:
                        X(24)=0.00081025:
                        X(25)=0.00116445:
                        X(26)=0.00086335:
                        X(27)=10.6:
                        X(28)=10.4:
                        X(29)=10.5:
                        X(30)=2000:
                        X(31)=6000;
          WRITE,        SawOut2:X(14);
          CLOSE,        SawOut2;
          WRITE,        ShearOut1:X(15);
          CLOSE,        ShearOut1;
          WRITE,        BrakeOut1:X(17);
          CLOSE,        BrakeOut1;
          WRITE,        BrakeOut2:X(18);
          CLOSE,        BrakeOut2;
          WRITE,        PunchOut1:X(19);
          CLOSE,        PunchOut1;
          WRITE,        PunchOut2:X(20);
          CLOSE,        PunchOut2;
          WRITE,        WeldOut1:X(21);
          CLOSE,        WeldOut1;
          WRITE,        WeldOut2:X(22);
          CLOSE,        WeldOut2;
          WRITE,        PaintOut1:X(23);
          CLOSE,        PaintOut1;
          WRITE,        PaintOut2:X(24);
          CLOSE,        PaintOut2;
          WRITE,        PackOut1:X(25);
          CLOSE,        PackOut1;
          WRITE,        PackOut2:X(26);
          CLOSE,        PackOut2:
                        DISPOSE;


;
;    Create Type I Jobs
;
          CREATE,,      36570;
repeat1   DELAY:        7;
          READ,         X(10):
                        A(1),A(3),A(4),A(10),A(8),A(5);
          ASSIGN:       A(2)=1:
                        M=Enter:
                        X(4)=X(4)+1:
                        A(1)=X(4);
          BRANCH,       3:
                        ALWAYS,Close1,Yes:
                        ALWAYS,Comp1,No:
                        ALWAYS,Comp2,No;

Close1    CLOSE,        X(10):
                        NEXT(repeat1);

Comp1     ASSIGN:       NS=1;
          DELAY:        MAX(A(4)-TNOW,0),1;
          ASSIGN:       X(5)=X(5)+A(3);
          ROUTE:        0,SEQ:MARK(A(6));

Comp2     ASSIGN:       NS=2;
          DELAY:        MAX(A(10)-TNOW,0),2;
          ASSIGN:       X(30)=X(30)+A(3);
          ROUTE:        0,SEQ:MARK(A(6));

;
;    Create Type II Jobs
;
          CREATE,,      36570;
repeat2   DELAY:        7;
          READ,         X(11):
```

```
                A(1),A(3),A(4),A(10),A(8),A(5);
        ASSIGN: A(2)=2:
                M=Enter:
                X(4)=X(4)+1:
                A(1)=X(4):
                A(5)=A(8);
        BRANCH, 3:
                ALWAYS,Close2,Yes:
                ALWAYS,Comp3,No;

Close2  CLOSE,  X(11):
                NEXT(repeat2);

Comp3   ASSIGN: NS=3;
        DELAY:  MAX(A(4)-TNOW,0),3;
        ASSIGN: X(31)=X(31)+A(3);
        ROUTE:  0,SEQ:MARK(A(6));

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                  ;
;                 Shop Floor Portion of Model                      ;
;                                                                  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

        STATION,  SawStn;
        ASSIGN:   A(7)=TNOW;
        QUEUE,    Saw_Q;
        SEIZE:    Saw:Machinist;
        DELAY:    Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                  Stdv(NS,M),9);
        RELEASE:  Saw:Machinist;
        BRANCH,   1:
                  IF,A(2)==1,label1A:
                  IF,A(2)==2,label1B;
label1A TALLY:    M,INT(A(7));
        TALLY:    21+M,INT(A(6));
        ASSIGN:   X(12)=(TNOW-A(7))/A(3):
                  X(13)=0.1*X(12)+0.9*X(13);
        WRITE,    SawOut1:X(13);
        CLOSE,    SawOut1:NEXT(out1);
label1B TALLY:    14+M,INT(A(7));
        TALLY:    21+M,INT(A(6));
        ASSIGN:   X(12)=(TNOW-A(7))/A(3):
                  X(14)=0.1*X(12)+0.9*X(14);
        WRITE,    SawOut2:X(14);
        CLOSE,    SawOut2;
out1    ROUTE:    0,SEQ;


        STATION,  ShearStn;
        ASSIGN:   A(7)=TNOW;
        QUEUE,    Shear_Q;
        SEIZE:    Shear:Machinist;
        DELAY:    Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                  Stdv(NS,M),9);
        RELEASE:  Shear:Machinist;
        BRANCH,   1:
                  IF,A(2)==1,label2A:
                  IF,A(2)==2,label2B;
label2A TALLY:    M,INT(A(7));
        TALLY:    21+M,INT(A(6));
        ASSIGN:   X(12)=(TNOW-A(7))/A(3):
                  X(15)=0.1*X(12)+0.9*X(15);
        WRITE,    ShearOut1:X(15);
        CLOSE,    ShearOut1:NEXT(out2);
label2B TALLY:    14+M,INT(A(7));
        TALLY:    21+M,INT(A(6));
        ASSIGN:   X(12)=(TNOW-A(7))/A(3):
                  X(16)=0.1*X(12)+0.9*X(16);
        WRITE,    ShearOut2:X(16);
        CLOSE,    ShearOut2;
```

```
out2      ROUTE:       0,SEQ;

          STATION,     BrakeStn;
          ASSIGN:      A(7)=TNOW;
          QUEUE,       Brake_Q;
          SEIZE:       Brake:Machinist;
          DELAY:       Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                       Stdv(NS,M),9);
          RELEASE:     Brake:Machinist;
          BRANCH,      1:
                       IF,A(2)==1,label3A:
                       IF,A(2)==2,label3B;
label3A   TALLY:       M,INT(A(7));
          TALLY:       21+M,INT(A(6));
          ASSIGN:      X(12)=(TNOW-A(7))/A(3):
                       X(17)=0.1*X(12)+0.9*X(17);
          WRITE,       BrakeOut1:X(17);
          CLOSE,       BrakeOut1:NEXT(out3);
label3B   TALLY:       14+M,INT(A(7));
          TALLY:       21+M,INT(A(6));
          ASSIGN:      X(12)=(TNOW-A(7))/A(3):
                       X(18)=0.1*X(12)+0.9*X(18);
          WRITE,       BrakeOut2:X(18);
          CLOSE,       BrakeOut2;
out3      ROUTE:       0,SEQ;

          STATION,     PunchStn;
          ASSIGN:      A(7)=TNOW;
          QUEUE,       Punch_Q;
          SEIZE:       Punch:Machinist;
          DELAY:       Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                       Stdv(NS,M),9);
          RELEASE:     Punch:Machinist;
          BRANCH,      1:
                       IF,A(2)==1,label4A:
                       IF,A(2)==2,label4B;
label4A   TALLY:       M,INT(A(7));
          TALLY:       21+M,INT(A(6));
          ASSIGN:      X(12)=(TNOW-A(7))/A(3):
                       X(19)=0.1*X(12)+0.9*X(19);
          WRITE,       PunchOut1:X(19);
          CLOSE,       PunchOut1:NEXT(out4);
label4B   TALLY:       14+M,INT(A(7));
          TALLY:       21+M,INT(A(6));
          ASSIGN:      X(12)=(TNOW-A(7))/A(3):
                       X(20)=0.1*X(12)+0.9*X(20);
          WRITE,       PunchOut2:X(20);
          CLOSE,       PunchOut2;
out4      ROUTE:       0,SEQ;

          STATION,     WeldStn;
          ASSIGN:      A(7)=TNOW;
          BRANCH,      1:
                       IF,NS==1,Q13:
                       IF,NS==2,Q24;
Q13       QUEUE,       Comp13Q:DETACH;
Q24       QUEUE,       Comp24Q:DETACH;
          MATCH,       A(1):
                       Q24,sort:
                       Q13,sort;
sort      TALLY:       7+NS,INT(A(7));
          BRANCH,      1:
                       IF,NS==1,out:
                       IF,NS==2,disp2;

disp2     ASSIGN:      X(30)=X(30)-A(3):
                       X(7)=A(6):
                       DISPOSE;
```

```
out        ASSIGN:     A(6)=MN(A(6),X(7));
           QUEUE,      Welder_Q;
           SEIZE:      Weld:Welder;
           DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                       Stdv(NS,M),9);
           RELEASE:    Weld:Welder;
           BRANCH,     1:
                       IF,A(2)==1,label5A:
                       IF,A(2)==2,label5B;
label5A    TALLY:      M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(21)=0.1*X(12)+0.9*X(21);
           WRITE,      WeldOut1:X(21);
           CLOSE,      WeldOut1:NEXT(out5);
label5B    TALLY:      14+M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(22)=0.1*X(12)+0.9*X(22);
           WRITE,      WeldOut2:X(22);
           CLOSE,      WeldOut2;
out5       ROUTE:      0,SEQ;

           STATION,    PaintStn;
           ASSIGN:     A(7)=TNOW;
           QUEUE,      Paint_Q;
           SEIZE:      Paint:Painter;
           DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                       Stdv(NS,M),9);
           RELEASE:    Paint:Painter;
           BRANCH,     1:
                       IF,A(2)==1,label6A:
                       IF,A(2)==2,label6B;
label6A    TALLY:      M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(23)=0.1*X(12)+0.9*X(23);
           WRITE,      PaintOut1:X(23);
           CLOSE,      PaintOut1:NEXT(out6);
label6B    TALLY:      14+M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(24)=0.1*X(12)+0.9*X(24);
           WRITE,      PaintOut2:X(24);
           CLOSE,      PaintOut2;
out6       ROUTE:      0,SEQ;

           STATION,    PackageStn;
           ASSIGN:     A(7)=TNOW;
           QUEUE,      Package_Q;
           SEIZE:      Package:Packager;
           DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                       Stdv(NS,M),9);
           RELEASE:    Package:Packager;
           BRANCH,     1:
                       IF,A(2)==1,label7A:
                       IF,A(2)==2,label7B;
label7A    TALLY:      M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(25)=0.1*X(12)+0.9*X(25);
           WRITE,      PackOut1:X(25);
           CLOSE,      PackOut1:NEXT(out7);
label7B    TALLY:      14+M,INT(A(7));
           TALLY:      21+M,INT(A(6));
           ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                       X(26)=0.1*X(12)+0.9*X(26);
           WRITE,      PackOut2:X(26);
           CLOSE,      PackOut2;
out7       ROUTE:      0,SEQ;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                      ;
;    Collect Statistics on Finished Orders/Jobs        ;
;                                                      ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

          STATION,       ExitSystem;

          ASSIGN:        X(1)=X(1)+1;
          TALLY:         11+NS,INT(A(6));
          TALLY:         13,INT(A(6));
          ASSIGN:        X(12)=TNOW-A(6):
                         X(29)=0.1*X(12)+0.9*X(29):
                         A(5)=X(29);
          BRANCH,        1:
                         IF,A(2)==1,ESFT1:
                         IF,A(2)==2,ESFT2;

ESFT1     ASSIGN:        X(12)=TNOW-A(6):
                         X(27)=0.1*X(12)+0.9*X(27):
                         A(7)=X(27);
          TALLY:         31,A(7);
          TALLY:         37,A(8)-A(6):NEXT(Tardiness);

ESFT2     ASSIGN:        X(12)=TNOW-A(6):
                         X(28)=0.1*X(12)+0.9*X(28):
                         A(7)=X(28);
          TALLY:         32,A(7);
          TALLY:         38,A(8)-A(6):NEXT(Tardiness);

Tardiness TALLY:         33,A(7);
          TALLY:         34,TNOW-A(8);
          TALLY:         36,(TNOW-A(6))-1.1*(A(8)-A(6));
          TALLY:         39,(TNOW-A(6))-1.2*(A(8)-A(6));
          BRANCH,        1:
                         IF,(TNOW-A(6))-(A(8)-A(6)).GT.0,Tardy:
                         ELSE,NoTardy;

Tardy     ASSIGN:        A(9)=(TNOW-A(6))-(A(8)-A(6)):
                         X(6)=X(6)+1:
                         NEXT(MeanTardy);

NoTardy   ASSIGN:        A(9)=0;

MeanTardy TALLY:         28+A(2),A(9);
          TALLY:         35,A(9);
          COUNT:         A(2),A(3);
          COUNT:         A(2)+2,1;
          BRANCH,        1:
                         IF,A(2)==1,disp1:
                         IF,A(2)==2,disp3;

disp1     ASSIGN:        X(5)=X(5)-A(3);
          WRITE,         P1info:
                         TNOW,A(7),TNOW-A(6),A(8)-A(6),A(3):
                         NEXT(mixINFO);

disp3     ASSIGN:        X(31)=X(31)-A(3);
          WRITE,         P2info:
                         TNOW,A(7),TNOW-A(6),A(8)-A(6),A(3):
                         NEXT(mixINFO);

mixINFO   WRITE,         P1P2info:
                         TNOW,A(5),TNOW-A(6),A(8),A(6),X(5)+X(30)+X(31),
                         DAVG(12),DAVG(22),DAVG(26),A(3):
                         DISPOSE;

          CREATE;
delblk    DELAY:         7;
          WRITE,         timefile:TNOW;
          CLOSE,         timefile:
```

```
        NEXT(delblk);
END;
```

```
BEGIN,yes,Original;
;
;        ATTRIBUTES (also used in ExitSystem stn as temp vars)
;        A(1) - Job Number
;        A(2) - Part Type
;        A(3) - Batch Size
;        A(4) - Component A Release Date
;        A(5) - Component Due Date (ESFT all types)
;        A(6) - Time In at Shop
;        A(7) - Time In at Station (ESFT by type)
;        A(8) - End Item Due Date
;        A(9) - Tardiness Attribute
;        A(10) - Component B Release Date
;
;
;        VARIABLES
;        X(1) - Total Number of Batches Completed
;        X(2) - Number of Batches Completed THIS Week
;        X(3) - Number of Batches Completed by LAST Week
;        X(4) - Job Number Index
;        X(5) - Parts WIP P1 C1
;        X(6) - Number of Tardy Jobs
;        X(7) - Temp Variable
;        X(8) -
;        X(9) -
;        X(10) - Which file to read P1 jobs from
;        X(11) - Which file to read P2 jobs from
;        X(12) - LT temp variable (used at all stns)
;        X(13) - Smoothed Adj FT per part at Saw Stn (P1)
;        X(14) - Smoothed Adj FT per part at Saw Stn (P2)
;        X(15) - Smoothed Adj FT per part at Shear Stn (P1)
;        X(16) - Smoothed Adj FT per part at Shear Stn (P2)
;        X(17) - Smoothed Adj FT per part at Brake Stn (P1)
;        X(18) - Smoothed Adj FT per part at Brake Stn (P2)
;        X(19) - Smoothed Adj FT per part at Punch Stn (P1)
;        X(20) - Smoothed Adj FT per part at Punch Stn (P2)
;        X(21) - Smoothed Adj FT per part at Weld Stn (P1)
;        X(22) - Smoothed Adj FT per part at Weld Stn (P2)
;        X(23) - Smoothed Adj FT per part at Paint Stn (P1)
;        X(24) - Smoothed Adj FT per part at Paint Stn (P2)
;        X(25) - Smoothed Adj FT per part at Pack Stn (P1)
;        X(26) - Smoothed Adj FT per part at Pack Stn (P2)
;        X(27) - Smoothed Adj Overall Batch FT for P1
;        X(28) - Smoothed Adj Overall Batch FT for P2
;        X(29) - Smoothed FT for all jobs (P1 & P2)
;        X(30) - Parts WIP P1 C2
;        X(31) - Parts WIP P2 C3
;        X(32) - Parts WIP P2 C4
;        Stup - Setup Time (defined in an array)
;        Mean - Process Mean (defined in an array)
;        Stdv - Process Standard Deviation (defined in an array)

;    Initialise Global Variables

         CREATE;
         ASSIGN:        X(4)=10:
                        X(5)=6000:
                        X(10)=1:
                        X(11)=2:
                        X(13)=0.00100635:
                        X(14)=0.0010662:
                        X(15)=0.0011303:
                        X(16)=0.00114:
                        X(17)=0.00117965:
                        X(18)=0.00132345:
                        X(19)=0.0010228:
                        X(20)=0.00110015:
                        X(21)=0.00120295:
                        X(22)=0.0012858:
                        X(23)=0.00081345:
                        X(24)=0.00081025:
```

```
                        X(25)=0.00116445:
                        X(26)=0.00086335:
                        X(27)=10.6:
                        X(28)=10.4:
                        X(29)=10.5:
                        X(30)=2000:
                        X(31)=6000:
                        X(32)=2000;
          WRITE,        SawOut1:X(13);
          CLOSE,        SawOut1;
          WRITE,        SawOut2:X(14);
          CLOSE,        SawOut2;
          WRITE,        ShearOut1:X(15);
          CLOSE,        ShearOut1;
          WRITE,        ShearOut2:X(16);
          CLOSE,        ShearOut2;
          WRITE,        BrakeOut1:X(17);
          CLOSE,        BrakeOut1;
          WRITE,        BrakeOut2:X(18);
          CLOSE,        BrakeOut2;
          WRITE,        PunchOut1:X(19);
          CLOSE,        PunchOut1;
          WRITE,        PunchOut2:X(20);
          CLOSE,        PunchOut2;
          WRITE,        WeldOut1:X(21);
          CLOSE,        WeldOut1;
          WRITE,        WeldOut2:X(22);
          CLOSE,        WeldOut2;
          WRITE,        PaintOut1:X(23);
          CLOSE,        PaintOut1;
          WRITE,        PaintOut2:X(24);
          CLOSE,        PaintOut2;
          WRITE,        PackOut1:X(25);
          CLOSE,        PackOut1;
          WRITE,        PackOut2:X(26);
          CLOSE,        PackOut2:
                        DISPOSE;


;
;    Create Type I Jobs
;
          CREATE,,      36570;
repeat1   DELAY:        7;
          READ,         X(10):
                        A(1),A(3),A(4),A(10),A(8),A(5);
          ASSIGN:       A(2)=1:
                        M=Enter:
                        X(4)=X(4)+1:
                        A(1)=X(4);
          BRANCH,       3:
                        ALWAYS,Close1,Yes:
                        ALWAYS,Comp1,No:
                        ALWAYS,Comp2,No;

Close1    CLOSE,        X(10):
                        NEXT(repeat1);

Comp1     ASSIGN:       NS=1;
          DELAY:        MAX(A(4)-TNOW,0),1;
          ASSIGN:       X(5)=X(5)+A(3);
          ROUTE:        0,SEQ:MARK(A(6));

Comp2     ASSIGN:       NS=2;
          DELAY:        MAX(A(10)-TNOW,0),2;
          ASSIGN:       X(30)=X(30)+A(3);
          ROUTE:        0,SEQ:MARK(A(6));

;
;    Create Type II Jobs
;
```

x

```
         CREATE,,        36570;
repeat2  DELAY:          7;
         READ,           X(11):
                         A(1),A(3),A(4),A(10),A(8),A(5);
         ASSIGN:         A(2)=2:
                         M=Enter:
                         X(4)=X(4)+1:
                         A(1)=X(4);
         BRANCH,         3:
                         ALWAYS,Close2,Yes:
                         ALWAYS,Comp3,No:
                         ALWAYS,Comp4,No;

Close2   CLOSE,          X(11):
                         NEXT(repeat2);

Comp3    ASSIGN:         NS=3;
         DELAY:          MAX(A(4)-TNOW,0),3;
         ASSIGN:         X(31)=X(31)+A(3);
         ROUTE:          0,SEQ:MARK(A(6));

Comp4    ASSIGN:         NS=4;
         DELAY:          MAX(A(10)-TNOW,0),4;
         ASSIGN:         X(32)=X(32)+A(3);
         ROUTE:          0,SEQ:MARK(A(6));

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                       ;
;                  Shop Floor Portion of Model                          ;
;                                                                       ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

         STATION,        SawStn;
         ASSIGN:         A(7)=TNOW;
         QUEUE,          Saw_Q;
         SEIZE:          Saw:Machinist;
         DELAY:          Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                         Stdv(NS,M),9);
         RELEASE:        Saw:Machinist;
         BRANCH,         1:
                         IF,A(2)==1,label1A:
                         IF,A(2)==2,label1B;
label1A  TALLY:          M,INT(A(7));
         TALLY:          21+M,INT(A(6));
         ASSIGN:         X(12)=(TNOW-A(7))/A(3):
                         X(13)=0.1*X(12)+0.9*X(13);
         WRITE,          SawOut1:X(13);
         CLOSE,          SawOut1:NEXT(out1);
label1B  TALLY:          14+M,INT(A(7));
         TALLY:          21+M,INT(A(6));
         ASSIGN:         X(12)=(TNOW-A(7))/A(3):
                         X(14)=0.1*X(12)+0.9*X(14);
         WRITE,          SawOut2:X(14);
         CLOSE,          SawOut2;
out1     ROUTE:          0,SEQ;


         STATION,        ShearStn;
         ASSIGN:         A(7)=TNOW;
         QUEUE,          Shear_Q;
         SEIZE:          Shear:Machinist;
         DELAY:          Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                         Stdv(NS,M),9);
         RELEASE:        Shear:Machinist;
         BRANCH,         1:
                         IF,A(2)==1,label2A:
                         IF,A(2)==2,label2B;
label2A  TALLY:          M,INT(A(7));
         TALLY:          21+M,INT(A(6));
         ASSIGN:         X(12)=(TNOW-A(7))/A(3):
                         X(15)=0.1*X(12)+0.9*X(15);
```

```
          WRITE,        ShearOut1:X(15);
          CLOSE,        ShearOut1:NEXT(out2);
label2B   TALLY:        14+M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(16)=0.1*X(12)+0.9*X(16);
          WRITE,        ShearOut2:X(16);
          CLOSE,        ShearOut2;
out2      ROUTE:        0,SEQ;


          STATION,      BrakeStn;
          ASSIGN:       A(7)=TNOW;
          QUEUE,        Brake_Q;
          SEIZE:        Brake:Machinist;
          DELAY:        Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                        Stdv(NS,M),9);
          RELEASE:      Brake:Machinist;
          BRANCH,       1:
                        IF,A(2)==1,label3A:
                        IF,A(2)==2,label3B;
label3A   TALLY:        M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(17)=0.1*X(12)+0.9*X(17);
          WRITE,        BrakeOut1:X(17);
          CLOSE,        BrakeOut1:NEXT(out3);
label3B   TALLY:        14+M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(18)=0.1*X(12)+0.9*X(18);
          WRITE,        BrakeOut2:X(18);
          CLOSE,        BrakeOut2;
out3      ROUTE:        0,SEQ;


          STATION,      PunchStn;
          ASSIGN:       A(7)=TNOW;
          QUEUE,        Punch_Q;
          SEIZE:        Punch:Machinist;
          DELAY:        Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                        Stdv(NS,M),9);
          RELEASE:      Punch:Machinist;
          BRANCH,       1:
                        IF,A(2)==1,label4A:
                        IF,A(2)==2,label4B;
label4A   TALLY:        M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(19)=0.1*X(12)+0.9*X(19);
          WRITE,        PunchOut1:X(19);
          CLOSE,        PunchOut1:NEXT(out4);
label4B   TALLY:        14+M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(20)=0.1*X(12)+0.9*X(20);
          WRITE,        PunchOut2:X(20);
          CLOSE,        PunchOut2;
out4      ROUTE:        0,SEQ;

          STATION,      WeldStn;
          ASSIGN:       A(7)=TNOW;
          BRANCH,       1:
                        IF,NS==1.OR.NS==3,Q13:
                        IF,NS==2.OR.NS==4,Q24;
Q13       QUEUE,        Comp13Q:DETACH;
Q24       QUEUE,        Comp24Q:DETACH;
          MATCH,        A(1):
                        Q24,sort:
                        Q13,sort;
sort      TALLY:        7+NS,INT(A(7));
```

```
        BRANCH,     1:
                    IF,NS==1.OR.NS==3,out:
                    IF,NS==2,disp2:
                    IF,NS==4,disp4;

disp2   ASSIGN:     X(30)=X(30)-A(3):
                    X(7)=A(6):
                    DISPOSE;

disp4   ASSIGN:     X(32)=X(32)-A(3):
                    X(7)=A(6);
                    DISPOSE;

out     ASSIGN:     A(6)=MN(A(6),X(7));
        QUEUE,      Welder_Q;
        SEIZE:      Weld:Welder;
        DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                    Stdv(NS,M),9);
        RELEASE:    Weld:Welder;
        BRANCH,     1:
                    IF,A(2)==1,label5A:
                    IF,A(2)==2,label5B;
label5A TALLY:      M,INT(A(7));
        TALLY:      21+M,INT(A(6));
        ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                    X(21)=0.1*X(12)+0.9*X(21);
        WRITE,      WeldOut1:X(21);
        CLOSE,      WeldOut1:NEXT(out5);
label5B TALLY:      14+M,INT(A(7));
        TALLY:      21+M,INT(A(6));
        ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                    X(22)=0.1*X(12)+0.9*X(22);
        WRITE,      WeldOut2:X(22);
        CLOSE,      WeldOut2;
out5    ROUTE:      0,SEQ;

        STATION,    PaintStn;
        ASSIGN:     A(7)=TNOW;
        QUEUE,      Paint_Q;
        SEIZE:      Paint:Painter;
        DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                    Stdv(NS,M),9);
        RELEASE:    Paint:Painter;
        BRANCH,     1:
                    IF,A(2)==1,label6A:
                    IF,A(2)==2,label6B;
label6A TALLY:      M,INT(A(7));
        TALLY:      21+M,INT(A(6));
        ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                    X(23)=0.1*X(12)+0.9*X(23);
        WRITE,      PaintOut1:X(23);
        CLOSE,      PaintOut1:NEXT(out6);
label6B TALLY:      14+M,INT(A(7));
        TALLY:      21+M,INT(A(6));
        ASSIGN:     X(12)=(TNOW-A(7))/A(3):
                    X(24)=0.1*X(12)+0.9*X(24);
        WRITE,      PaintOut2:X(24);
        CLOSE,      PaintOut2;
out6    ROUTE:      0,SEQ;

        STATION,    PackageStn;
        ASSIGN:     A(7)=TNOW;
        QUEUE,      Package_Q;
        SEIZE:      Package:Packager;
        DELAY:      Stup(NS,M)+A(3)*NORM(Mean(NS,M),
                    Stdv(NS,M),9);
        RELEASE:    Package:Packager;
        BRANCH,     1:
                    IF,A(2)==1,label7A:
                    IF,A(2)==2,label7B;
label7A TALLY:      M,INT(A(7));
```

```
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(25)=0.1*X(12)+0.9*X(25);
          WRITE,        PackOut1:X(25);
          CLOSE,        PackOut1:NEXT(out7);
label7B   TALLY:        14+M,INT(A(7));
          TALLY:        21+M,INT(A(6));
          ASSIGN:       X(12)=(TNOW-A(7))/A(3):
                        X(26)=0.1*X(12)+0.9*X(26);
          WRITE,        PackOut2:X(26);
          CLOSE,        PackOut2;
out7      ROUTE:        0,SEQ;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;    Collect Statistics on Finished Orders/Jobs ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

          STATION,      ExitSystem;

          ASSIGN:       X(1)=X(1)+1;
          TALLY:        11+NS,INT(A(6));
          TALLY:        13,INT(A(6));
          ASSIGN:       X(12)=TNOW-A(6):
                        X(29)=0.1*X(12)+0.9*X(29):
                        A(5)=X(29);
          BRANCH,       1:
                        IF,A(2)==1,ESFT1:
                        IF,A(2)==2,ESFT2;

ESFT1     ASSIGN:       X(12)=TNOW-A(6):
                        X(27)=0.1*X(12)+0.9*X(27):
                        A(7)=X(27);
          TALLY:        31,A(7);
          TALLY:        37,A(8)-A(6):NEXT(Tardiness);

ESFT2     ASSIGN:       X(12)=TNOW-A(6):
                        X(28)=0.1*X(12)+0.9*X(28):
                        A(7)=X(28);
          TALLY:        32,A(7);
          TALLY:        38,A(8)-A(6):NEXT(Tardiness);

Tardiness TALLY:        33,A(7);
          TALLY:        34,TNOW-A(8);
          TALLY:        36,(TNOW-A(6))-1.1*(A(8)-A(6));
          TALLY:        39,(TNOW-A(6))-1.2*(A(8)-A(6));
          BRANCH,       1:
                        IF,(TNOW-A(6))-(A(8)-A(6)).GT.0,Tardy:
                        ELSE,NoTardy;

Tardy     ASSIGN:       A(9)=(TNOW-A(6))-(A(8)-A(6)):
                        X(6)=X(6)+1:
                        NEXT(MeanTardy);

NoTardy   ASSIGN:       A(9)=0;

MeanTardy TALLY:        28+A(2),A(9);
          TALLY:        35,A(9);
          COUNT:        A(2),A(3);
          COUNT:        A(2)+2,1;
          BRANCH,       1:
                        IF,A(2)==1,disp1:
                        IF,A(2)==2,disp3;

disp1     ASSIGN:       X(5)=X(5)-A(3);
          WRITE,        P1info:
                        TNOW,A(7),TNOW-A(6),A(8)-A(6),A(3):
                        NEXT(mixINFO);

disp3     ASSIGN:       X(31)=X(31)-A(3);
          WRITE,        P2info:
                        TNOW,A(7),TNOW-A(6),A(8)-A(6),A(3):
```

```
                        NEXT(mixINFO);

mixINFO  WRITE,         P1P2info:
                        TNOW,A(5),TNOW-A(6),A(8),A(6),X(5)+X(30)+X(31)+X(32),
                        DAVG(12),DAVG(22),DAVG(27),A(3):
                        DISPOSE;

         CREATE;
delblk   DELAY:         7;
         WRITE,         timefile:TNOW;
         CLOSE,         timefile:
                        NEXT(delblk);

END;
```

Appendix 2


```
BEGIN,NO,YES;

PROJECT,        Modified, RM;

DISCRETE,       2000,10,9,9;

VARIABLES:      1:
                2:
                3:
                4:
                5:
                6:
                7:
                8:
                9:
                10:
                11:
                12:
                13:
                14:
                15:
                16:
                17:
                18:
                19:
                20:
                21:
                22:
                23:
                24:
                25:
                26:
                27:
                28:
                29:
                30:
                31:
                Stup(4,7),      0.1250,  0.1250,  0.1250,  0.1250,
                                0.0833,  0.0833,  0.1042,  0.1042,
                                0.0729,  0.0729,  0.0833,  0.0833,
                                0.0313,  0.0313,  0.0104,  0.0104,
                                0.0625,  0,       0.0625,  0,
                                0.1250,  0,       0.0521,  0,
                                0.0208,  0,       0.0833,  0:
                Mean(4,7),      0.00076, 0.00076, 0.00078, 0.00078,
                                0.00086, 0.00086, 0.00081, 0.00081,
                                0.00093, 0.00093, 0.00114, 0.00114,
                                0.00093, 0.00093, 0.00100, 0.00100,
                                0.00083, 0,       0.00089, 0,
                                0.00071, 0,       0.00078, 0,
                                0.00109, 0,       0.00071, 0:
                Stdv(4,7),      0.000227,0.000227,0.000234,0.000234,
                                0.000259,0.000259,0.000242,0.000242,
                                0.000278,0.000278,0.000341,0.000341,
                                0.000278,0.000278,0.000300,0.000300,
                                0.000250,0,       0.000268,0,
                                0.000214,0,       0.000234,0,
                                0.000326,0,       0.000214,0;

QUEUES:         1,Saw_Q,LVF(A(8)):
                2,Shear_Q,LVF(A(8)):
                3,Brake_Q,LVF(A(8)):
                4,Punch_Q,LVF(A(8)):
                5,Welder_Q,LVF(A(8)):
                6,Paint_Q,LVF(A(8)):
                7,Package_Q,LVF(A(8)):
                8,Comp13Q:
```

```
                9,Comp24Q;

STATIONS:       1,SawStn:
                2,ShearStn:
                3,BrakeStn:
                4,PunchStn:
                5,WeldStn:
                6,PaintStn:
                7,PackageStn:
                8,ExitSystem:
                9,Enter;

SEQUENCES:      1,,ShearStn&PunchStn&
                     WeldStn&PaintStn&
                     PackageStn&ExitSystem:
                2,,BrakeStn&WeldStn:
                3,,SawStn&ShearStn&
                     PunchStn&BrakeStn&
                     PaintStn&PackageStn&ExitSystem;

FILES:          1,Data1,"..\SRM\DATA1.TXT",SEQ,FREE,IGNORE:
                2,Data2,"..\SRM\DATA2.TXT",SEQ,FREE,IGNORE:
                3,P1info,"P1info.TXT",SEQ,FREE,IGNORE:
                4,P2info,"P2info.TXT",SEQ,FREE,IGNORE:
                5,SawOut1,"..\SRM\SawP1.TXT",SEQ,FREE,IGNORE:
                6,SawOut2,"..\SRM\SawP2.TXT",SEQ,FREE,IGNORE:
                7,ShearOut1,"..\SRM\ShearP1.TXT",SEQ,FREE,IGNORE:
                8,ShearOut2,"..\SRM\ShearP2.TXT",SEQ,FREE,IGNORE:
                9,BrakeOut1,"..\SRM\BrakeP1.TXT",SEQ,FREE,IGNORE:
                10,BrakeOut2,"..\SRM\BrakeP2.TXT",SEQ,FREE,IGNORE:
                11,PunchOut1,"..\SRM\PunchP1.TXT",SEQ,FREE,IGNORE:
                12,PunchOut2,"..\SRM\PunchP2.TXT",SEQ,FREE,IGNORE:
                13,WeldOut1,"..\SRM\WeldP1.TXT",SEQ,FREE,IGNORE:
                14,WeldOut2,"..\SRM\WeldP2.TXT",SEQ,FREE,IGNORE:
                15,PaintOut1,"..\SRM\PaintP1.TXT",SEQ,FREE,IGNORE:
                16,PaintOut2,"..\SRM\PaintP2.TXT",SEQ,FREE,IGNORE:
                17,PackOut1,"..\SRM\PackP1.TXT",SEQ,FREE,IGNORE:
                18,PackOut2,"..\SRM\PackP2.TXT",SEQ,FREE,IGNORE:
                19,P1P2info,"P1P2info.txt",SEQ,FREE,IGNORE:
                20,timefile,"..\SRM\timefile.txt",,FREE,;

STORAGES:       1,RelDate1:
                2,RelDate2:
                3,RelDate3;

RESOURCES:      Machinist,4:
                Welder:
                Painter:
                Packager:
                Saw:
                Shear:
                Brake:
                Punch:
                Weld:
                Paint:
                Package;

ARRIVALS:       1,STATION(BrakeStn),,1,
                 A(1)=1,A(2)=1,A(3)=2000,A(4)=36571,A(5)=36576,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
                 M=1,NS=2,IS=1:
                2,STATION(ShearStn),,1,
                 A(1)=1,A(2)=1,A(3)=2000,A(4)=36571,A(5)=36576,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
                 M=2,NS=1,IS=1:
                3,STATION(PaintStn),,1,
                 A(1)=2,A(2)=1,A(3)=2000,A(4)=36570,A(5)=36574,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36581,A(10)=36570,
                 M=6,NS=1,IS=4:
                4,STATION(PackageStn),,1,
                 A(1)=3,A(2)=1,A(3)=2000,A(4)=36569,A(5)=36573,
```

```
                 A(6)=TNOW,A(7)=TNOW,A(8)=36580,A(10)=36569,
                 M=7,NS=1,IS=5:
                5,STATION(ShearStn),,1,
                 A(1)=4,A(2)=2,A(3)=2000,A(4)=36571,A(5)=36576,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
                 M=2,NS=3,IS=1:
                6,STATION(PaintStn),,1,
                 A(1)=5,A(2)=2,A(3)=2000,A(4)=36570,A(5)=36574,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36581,A(10)=36570,
                 M=5,NS=3,IS=4:
                7,STATION(PackageStn),,1,
                 A(1)=6,A(2)=2,A(3)=2000,A(4)=36569,A(5)=36573,
                 A(6)=TNOW,A(7)=TNOW,A(8)=36579,A(10)=36569,
                 M=7,NS=3,IS=5;

DSTATS:         100*NR(Saw),Saw Util:
                100*NR(Shear),Shear Util:
                100*NR(Brake),Brake Util:
                100*NR(Punch),Punch Util:
                100*NR(Weld),Weld Util:
                100*NR(Paint),Paint Util:
                100*NR(Package),Package Util:
                100*NR(Machinist)/MR(Machinist),Machinist Util:
                100*NR(Welder),Welder Util:
                100*NR(Painter),Painter Util:
                100*NR(Packager),Packager Util:
                100*(NR(Saw)+NR(Shear)+NR(Brake)
                  +NR(Punch)+NR(Weld)+NR(Paint)+NR(Package))/7,Shop Avg Util:
                NQ(Saw_Q),Saw Q Length:
                NQ(Shear_Q),Shear Q Length:
                NQ(Brake_Q),Brake Q Length:
                NQ(Punch_Q),Punch Q Length:
                NQ(Welder_Q),Welder Q Length:
                NQ(Paint_Q),Paint Q Length:
                NQ(Package_Q),Package Q Length:
                NQ(Comp13Q),Match Q1 Length:
                NQ(Comp24Q),Match Q2 Length:
                100*X(6)/MX(X(1),1),Percent Tardy:
                X(5),C1P1 in system:
                X(30),C2 in system:
                X(31),P2 in system:
                X(5)+X(30)+X(31),Total WIP;

TALLIES:        1,Saw FT Type I:
                2,Shear FT Type I:
                3,Brake FT Type I:
                4,Punch FT Type I:
                5,Weld FT Type I:
                6,Paint FT Type I:
                7,Package FT Type I:
                8,C1 Wait for Match:
                9,C2 Wait for Match:
                10,C3 Wait for Match:
                11,C4 Wait for Match:
                12,Type I FT:
                13,Overall FT:
                14,Type II FT:
                15,Saw FT Type II:
                16,Shear FT Type II:
                17,Brake FT Type II:
                18,Punch FT Type II:
                19,Weld FT Type II:
                20,Paint FT Type II:
                21,Package FT Type II:
                22,Stn1 Cum. FT:
                23,Stn2 Cum. FT:
                24,Stn3 Cum. FT:
                25,Stn4 Cum. FT:
                26,Stn5 Cum. FT:
                27,Stn6 Cum. FT:
                28,Stn7 Cum. FT:
```

```
               29,Mean Tardiness I:
               30,Mean Tardiness II:
               31,Type I ESFT:
               32,Type II ESFT:
               33,Overall ESFT:
               34,No SF Lateness:
               35,Mean Tardiness Avg:
               36,Adj 10% Lateness:
               37,Flow Allowance I:
               38,Flow Allowance II:
               39,Adj 20% Lateness;

COUNTERS:      1,Type I Parts Finished:
               2,Type II Parts Finished:
               3,Type I Batches Finished:
               4,Type II Batches Finished;

OUTPUTS:       TAVG(31),,Type I ESFT:
               TAVG(32),,Type II ESFT:
               TAVG(33),,Overall ESFT:
               TAVG(29),,Mean Tardiness I:
               TAVG(30),,Mean Tardiness II:
               TAVG(35),,Mean Tardiness Avg:
               TAVG(34),,No SF Lateness:
               TAVG(36),,Adj 10% Lateness:
               TAVG(39),,Adj 20% Lateness:
               TAVG(37),,Flow Allowance I:
               TAVG(38),,Flow Allowance II:
               DAVG(22),,Percent Tardy:
               DAVG(26),,WIP in system:
               DAVG(12),,Shop Avg Util:
               TNOW,,End of Rep;

SEEDS:         1,12345,no:
               2,23456,no:
               3,34567,no:
               4,45678,no:
               5,56789,no:
               6,67890,no:
               7,78901,no:
               8,89012,no:
               9,90123,no:
               10,01234,no;

REPLICATE,     1,36570;

END;
```

```
BEGIN,NO,YES;

PROJECT,        Original, RM;

DISCRETE,       2000,10,9,9;

VARIABLES:      1:
                2:
                3:
                4:
                5:
                6:
                7:
                8:
                9:
                10:
                11:
                12:
                13:
                14:
                15:
                16:
                17:
                18:
                19:
                20:
                21:
                22:
                23:
                24:
                25:
                26:
                27:
                28:
                29:
                30:
                31:
                32:
                Stup(4,7),      0,        0.1250,  0,        0.1250,
                                0.0833,   0,       0.1042,   0,
                                0,        0.0729,  0,        0.0833,
                                0.0313,   0,       0.0104,   0,
                                0.0625,   0,       0.0625,   0,
                                0.1250,   0,       0.0521,   0,
                                0.0208,   0,       0.0833,   0:
                Mean(4,7),      0,        0.00076, 0,        0.00078,
                                0.00086,  0,       0.00081,  0,
                                0,        0.00093, 0,        0.00114,
                                0.00093,  0,       0.00100,  0,
                                0.00083,  0,       0.00089,  0,
                                0.00071,  0,       0.00078,  0,
                                0.00109,  0,       0.00071,  0:
                Stdv(4,7),      0,        0.000227,0,        0.000234,
                                0.000259,0,        0.000242,0,
                                0,        0.000278,0,        0.000341,
                                0.000278,0,        0.000300,0,
                                0.000250,0,        0.000268,0,
                                0.000214,0,        0.000234,0,
                                0.000326,0,        0.000214,0;

QUEUES:         1,Saw_Q,LVF(A(5)):
                2,Shear_Q,LVF(A(5)):
                3,Brake_Q,LVF(A(5)):
                4,Punch_Q,LVF(A(5)):
                5,Welder_Q,LVF(A(8)):
                6,Paint_Q,LVF(A(8)):
                7,Package_Q,LVF(A(8)):
                8,Comp13Q:
                9,Comp24Q;

STATIONS:       1,SawStn:
```

```
               2,ShearStn:
               3,BrakeStn:
               4,PunchStn:
               5,WeldStn:
               6,PaintStn:
               7,PackageStn:
               8,ExitSystem:
               9,Enter;

SEQUENCES:     1,,ShearStn&PunchStn&
                    WeldStn&PaintStn&
                    PackageStn&ExitSystem:
               2,,SawStn&BrakeStn&WeldStn:
               3,,ShearStn&PunchStn&
                    WeldStn&PaintStn&
                    PackageStn&ExitSystem:
               4,,SawStn&BrakeStn&WeldStn;

FILES:         1,Data1,"..\SRM\DATA1.TXT",SEQ,FREE,IGNORE:
               2,Data2,"..\SRM\DATA2.TXT",SEQ,FREE,IGNORE:
               3,P1info,"P1info.TXT",SEQ,FREE,IGNORE:
               4,P2info,"P2info.TXT",SEQ,FREE,IGNORE:
               5,SawOut1,"..\SRM\SawP1.TXT",SEQ,FREE,IGNORE:
               6,SawOut2,"..\SRM\SawP2.TXT",SEQ,FREE,IGNORE:
               7,ShearOut1,"..\SRM\ShearP1.TXT",SEQ,FREE,IGNORE:
               8,ShearOut2,"..\SRM\ShearP2.TXT",SEQ,FREE,IGNORE:
               9,BrakeOut1,"..\SRM\BrakeP1.TXT",SEQ,FREE,IGNORE:
               10,BrakeOut2,"..\SRM\BrakeP2.TXT",SEQ,FREE,IGNORE:
               11,PunchOut1,"..\SRM\PunchP1.TXT",SEQ,FREE,IGNORE:
               12,PunchOut2,"..\SRM\PunchP2.TXT",SEQ,FREE,IGNORE:
               13,WeldOut1,"..\SRM\WeldP1.TXT",SEQ,FREE,IGNORE:
               14,WeldOut2,"..\SRM\WeldP2.TXT",SEQ,FREE,IGNORE:
               15,PaintOut1,"..\SRM\PaintP1.TXT",SEQ,FREE,IGNORE:
               16,PaintOut2,"..\SRM\PaintP2.TXT",SEQ,FREE,IGNORE:
               17,PackOut1,"..\SRM\PackP1.TXT",SEQ,FREE,IGNORE:
               18,PackOut2,"..\SRM\PackP2.TXT",SEQ,FREE,IGNORE:
               19,P1P2info,"P1P2info.txt",SEQ,FREE,IGNORE:
               20,timefile,"..\SRM\timefile.txt",,FREE,;

STORAGES:      1,RelDate1:
               2,RelDate2:
               3,RelDate3:
               4,RelDate4;

RESOURCES:     Machinist,4:
               Welder:
               Painter:
               Packager:
               Saw:
               Shear:
               Brake:
               Punch:
               Weld:
               Paint:
               Package;

ARRIVALS:      1,STATION(SawStn),,1,
                A(1)=1,A(2)=1,A(3)=2000,A(4)=36571,A(5)=36576,
                A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
                M=1,NS=2,IS=1:
               2,STATION(ShearStn),,1,
                A(1)=1,A(2)=1,A(3)=2000,A(4)=36571,A(5)=36576,
                A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
                M=2,NS=1,IS=1:
               3,STATION(PaintStn),,1,
                A(1)=2,A(2)=1,A(3)=2000,A(4)=36570,A(5)=36574,
                A(6)=TNOW,A(7)=TNOW,A(8)=36581,A(10)=36570,
                M=6,NS=1,IS=4:
               4,STATION(PackageStn),,1,
                A(1)=3,A(2)=1,A(3)=2000,A(4)=36569,A(5)=36573,
                A(6)=TNOW,A(7)=TNOW,A(8)=36580,A(10)=36569,
```

```
         M=7,NS=1,IS=5:
         5,STATION(SawStn),,1,
          A(1)=4,A(2)=2,A(3)=2000,A(4)=36571,A(5)=36576,
          A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
          M=1,NS=4,IS=1:
         6,STATION(ShearStn),,1,
          A(1)=4,A(2)=2,A(3)=2000,A(4)=36571,A(5)=36576,
          A(6)=TNOW,A(7)=TNOW,A(8)=36585,A(10)=36571,
          M=2,NS=3,IS=1:
         7,STATION(PaintStn),,1,
          A(1)=5,A(2)=2,A(3)=2000,A(4)=36570,A(5)=36574,
          A(6)=TNOW,A(7)=TNOW,A(8)=36581,A(10)=36570,
          M=5,NS=3,IS=4:
         8,STATION(PackageStn),,1,
          A(1)=6,A(2)=2,A(3)=2000,A(4)=36569,A(5)=36573,
          A(6)=TNOW,A(7)=TNOW,A(8)=36579,A(10)=36569,
          M=7,NS=3,IS=5;

DSTATS:  100*NR(Saw),Saw Util:
         100*NR(Shear),Shear Util:
         100*NR(Brake),Brake Util:
         100*NR(Punch),Punch Util:
         100*NR(Weld),Weld Util:
         100*NR(Paint),Paint Util:
         100*NR(Package),Package Util:
         100*NR(Machinist)/MR(Machinist),Machinist Util:
         100*NR(Welder),Welder Util:
         100*NR(Painter),Painter Util:
         100*NR(Packager),Packager Util:
         100*(NR(Saw)+NR(Shear)+NR(Brake)
           +NR(Punch)+NR(Weld)+NR(Paint)+NR(Package))/7,Shop Avg Util:
         NQ(Saw_Q),Saw Q Length:
         NQ(Shear_Q),Shear Q Length:
         NQ(Brake_Q),Brake Q Length:
         NQ(Punch_Q),Punch Q Length:
         NQ(Welder_Q),Welder Q Length:
         NQ(Paint_Q),Paint Q Length:
         NQ(Package_Q),Package Q Length:
         NQ(Comp13Q),Match Q1 Length:
         NQ(Comp24Q),Match Q2 Length:
         100*X(6)/MX(X(1),1),Percent Tardy:
         X(5),C1P1 in System:
         X(30),C2 in system:
         X(31),C3P2 in system:
         X(32),C4 in system:
         X(5)+X(30)+X(31)+X(32),Total WIP;

TALLIES: 1,Saw FT Type I:
         2,Shear FT Type I:
         3,Brake FT Type I:
         4,Punch FT Type I:
         5,Weld FT Type I:
         6,Paint FT Type I:
         7,Package FT Type I:
         8,C1 Wait for Match:
         9,C2 Wait for Match:
         10,C3 Wait for Match:
         11,C4 Wait for Match:
         12,Type I FT:
         13,Overall FT:
         14,Type II FT:
         15,Saw FT Type II:
         16,Shear FT Type II:
         17,Brake FT Type II:
         18,Punch FT Type II:
         19,Weld FT Type II:
         20,Paint FT Type II:
         21,Package FT Type II:
         22,Stn1 Cum. FT:
         23,Stn2 Cum. FT:
         24,Stn3 Cum. FT:
```

```
                25,Stn4 Cum. FT:
                26,Stn5 Cum. FT:
                27,Stn6 Cum. FT:
                28,Stn7 Cum. FT:
                29,Mean Tardiness I:
                30,Mean Tardiness II:
                31,Type I ESFT:
                32,Type II ESFT:
                33,Overall ESFT:
                34,Lateness:
                35,Mean Tardiness Avg:
                36,Adj 10% Lateness:
                37,Flow Allowance I:
                38,Flow Allowance II:
                39,Adj 20% Lateness;

COUNTERS:       1,Type I Parts Finished:
                2,Type II Parts Finished:
                3,Type I Batches Finished:
                4,Type II Batches Finished;

OUTPUTS:        TAVG(31),,Type I ESFT:
                TAVG(32),,Type II ESFT:
                TAVG(33),,Overall ESFT:
                TAVG(29),,Mean Tardiness I:
                TAVG(30),,Mean Tardiness II:
                TAVG(35),,Mean Tardiness Avg:
                TAVG(34),,No SF Lateness:
                TAVG(36),,Adj 10% Lateness:
                TAVG(39),,Adj 20% Lateness:
                TAVG(37),,Flow Allowance I:
                TAVG(38),,Flow Allowance II:
                DAVG(22),,Percent Tardy:
                DAVG(27),,WIP in system:
                DAVG(12),,Shop Avg Util:
                TNOW,,End of Rep;

SEEDS:          1,12345,no:
                2,23456,no:
                3,34567,no:
                4,45678,no:
                5,56789,no:
                6,67890,no:
                7,78901,no:
                8,89012,no:
                9,90123,no:
                10,01234,no;

REPLICATE,      1,36570;

END;
```

Original Product Set - Calculation of Average Utilisation Level

total available time	3360
demand per week	2100

	P1				P2				TOTALS		
	setup time	parts/hr rate	part proc time	total time	setup time	parts/hr rate	part proc time	total time	Total Prod Time	Mach Util %	B/n m/c
saw	60	165	0.36	823.64	60	160	0.38	847.50	1671.14	49.74	
shear	40	145	0.41	908.97	50	155	0.39	862.90	1771.87	52.73	
brake	35	135	0.44	968.33	40	110	0.55	1185.45	2153.79	64.10	brake
punch	15	135	0.44	948.33	5	125	0.48	1013.00	1961.33	58.37	
weld	30	150	0.40	870.00	30	140	0.43	930.00	1800.00	53.57	
paint	60	175	0.34	780.00	25	160	0.38	812.50	1592.50	47.40	
pack	10	115	0.52	1105.65	40	175	0.34	760.00	1865.65	55.53	

Avg Util 54.49098

Modified Product Set - Calculation of Average Utilisation Level

total available time 3360
demand per week 2100

	P1				P2				TOTALS		
	setup time	parts/hr rate	part proc time	total time	setup time	parts/hr rate	part proc time	total time	Total Proc Time	Mach Util %	B/n m/c
saw	0	0	0.00	0.00	60	160	0.38	847.50	847.50	25.22	
shear	40	145	0.41	908.97	50	155	0.39	862.90	1771.87	52.73	
brake	35	135	0.44	968.33	40	110	0.55	1185.45	2153.79	64.10	brake
punch	15	135	0.44	948.33	5	125	0.48	1013.00	1961.33	58.37	
weld	30	150	0.40	870.00	0	0	0.00	0.00	870.00	25.89	
paint	60	175	0.34	780.00	25	160	0.38	812.50	1592.50	47.40	
pack	10	115	0.52	1105.65	40	175	0.34	760.00	1865.65	55.53	

Avg Util 47.035

@AVG(K6..K12)

@IF(G6=0,0,+60/G6)

(F6)+(H6*C2)

+E6+I6

100*J6/C1

@IF(@MAX(J$6..J$12)=J6,A6,"")

```
MACRO
Truncate w1 w2

MCOLUMN w1 w2 x1 x2 temp temp2
MCONSTANT yst yend n i tm

copy w1 w2 x1-x2;
omit w1=0:38029.9999.

LET yst=38030
LET yend=38395
LET n=20

DO i=1:n
  COPY x2 temp;
  USE x1=yst:yend.
  LET tm=mean(temp)
  PRINT tm
  LET temp2(i)=tm
  LET yst=yst+730.5
  LET yend=yend+730.5
ENDDO

WRITE 'samples.txt' temp2

NOTE
NOTE         ** CHANGE FILE NAME **
NOTE
NOTE ** DO NOT forget to change file names as necessary **
NOTE

ENDMACRO
```

```
GMACRO
P1P2Read
erase c1-c20
Read 'C:\THESIS\DYNAMIC\EXP27\P1P2INFO.TXT' c1-c10.
END
let c11=c1-c4
let c12=c1-c1
rmaximum c11 c12 c13
name c1='tnow'
name c2='esft'
name c3='ft'
name c4='duedate'
name c5='time in'
name c6='act wip'
name c7='avg util'
name c8='%tardy'
name c9='avg wip'
name c10='batch sz'
name c11='lateness'
name c12='zero'
name c13='mtardy'
describe c11
Save  'C:\THESIS\MINITAB\P1P2.MTW';
Replace.
NOTE
NOTE
NOTE ** Do NOT forget to change file names as necessary **
NOTE
NOTE
ENDMACRO
```

```
MACRO
Filetext w1 w2

MCOLUMN w1 w2 x1 x2

copy w1 w2 x1-x2;
omit w1=0:38029.9999.

WRITE 'samples.txt' x1 x2

NOTE
NOTE
NOTE ** DO NOT forget to change file names as necessary **
NOTE
NOTE

ENDMACRO
```

Appendix 5

SIMAN V - License #9510000
Roger Mattar

Summary for Replication 1 of 1

Project: Modified Set, Component Due Date   Run execution date : 2/27/1997
Analyst: RM   Model revision date: 2/26/1997

Replication ended at time : 42708.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
Saw FT Type I	--	--	--	--	0
Shear FT Type I	2.8262	.40842	.00000	7.7723	876
Brake FT Type I	4.9277	.60484	.47251	18.751	876
Punch FT Type I	3.3516	.42604	.67655	10.068	876
Weld FT Type I	4.6639	.53487	.46750	15.721	876
Paint FT Type I	2.5805	.50582	.40438	11.018	876
Package FT Type I	3.6433	.48436	.30963	12.971	877
C1 Wait for Match	2.2827	1.0149	.00000	11.830	876
C2 Wait for Match	.33502	2.4649	.00000	5.9870	876
C3 Wait for Match	--	--	--	--	0
C4 Wait for Match	--	--	--	--	0
Type I FT	17.042	.25980	3.1432	33.042	877
Overall FT	18.499	.30471	3.1432	38.133	1753
Type II FT	19.957	.31587	7.4288	38.133	876
Saw FT Type II	2.2895	.32343	.22499	4.9867	875
Shear FT Type II	2.5973	.42397	.37298	11.216	877
Brake FT Type II	5.7286	.63262	.32899	17.818	875
Punch FT Type II	3.6762	.57817	.00000	14.204	876
Weld FT Type II	--	--	--	--	0
Paint FT Type II	2.7919	.41125	.52680	7.5362	876
Package FT Type II	2.8961	.46584	.19637	8.8315	877
Stn1 Cum.FT	2.2895	.32343	.22499	4.9867	875
Stn2 Cum.FT	3.8567	.42862	.00000	13.290	1753
Stn3 Cum.FT	9.6079	.65577	.47251	30.056	1751
Stn4 Cum.FT	7.3706	.36519	1.2065	19.611	1752
Stn5 Cum.FT	10.842	.28125	4.5735	23.045	876
Stn6 Cum.FT	15.237	.33631	1.4568	34.449	1752
Stn7 Cum.FT	18.491	.30530	3.1432	38.133	1754
Mean Tardiness I	.63294	2.3385	.00000	12.042	877
Mean Tardiness II	3.6796	1.0648	.00000	17.533	876
Type I ESFT	17.003	.15406	9.5111	24.711	877
Type II ESFT	19.894	.19996	9.9450	31.040	876
Overall ESFT	18.448	.19861	9.5111	31.040	1753
No SF Lateness	.91227	4.8535	-8.784E+00	17.533	1753
Mean Tardiness Avg	2.1554	1.5447	.00000	17.533	1753
Adj 10% Lateness	-8.464E-01	-5.176E+00	-1.069E+01	15.881	1753
Flow Allowance I	18.459	.13875	10.000	21.000	877
Flow Allowance II	16.713	.15295	9.0000	21.000	876
Adj 20% Lateness	-2.605E+00	-1.670E+00	-1.279E+01	14.229	1753

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Saw Util	32.638	1.4366	.00000	100.00	.00000
Shear Util	68.265	.68181	.00000	100.00	100.00
Brake Util	84.492	.42842	.00000	100.00	100.00
Punch Util	78.043	.53041	.00000	100.00	.00000
Weld Util	33.900	1.3963	.00000	100.00	.00000
Paint Util	62.009	.78272	.00000	100.00	100.00
Package Util	73.099	.60662	.00000	100.00	.00000
Machinist Util	65.859	.28351	.00000	100.00	50.000
Welder Util	33.900	1.3963	.00000	100.00	.00000
Painter Util	62.009	.78272	.00000	100.00	100.00
Packager Util	73.099	.60662	.00000	100.00	.00000
Shop Avg Util	61.778	.23094	.00000	100.00	42.857
Saw Q Length	.00000	--	.00000	.00000	.00000
Shear Q Length	.09191	3.1856	.00000	2.0000	.00000
Brake Q Length	.67507	1.3752	.00000	5.0000	.00000
Punch Q Length	.22257	2.0794	.00000	3.0000	.00000
Welder Q Length	8.2633E-04	34.773	.00000	1.0000	.00000
Paint Q Length	.14713	2.4949	.00000	3.0000	1.0000
Package Q Length	.20337	2.1533	.00000	3.0000	.00000
Match Q1 Length	.32579	1.5147	.00000	2.0000	.00000
Match Q2 Length	.04781	4.4625	.00000	1.0000	.00000
Percent Tardy	50.830	.09992	.00000	72.727	50.998
C1P1 in system	7001.7	.41761	.00000	15930.	4850.0
C2 in system	2187.4	.86604	.00000	10420.	.00000
P2 in system	8255.8	.46357	.00000	20460.	4530.0
Total WIP	17445.	.43677	.00000	43370.	9380.0

COUNTERS

Identifier	Count	Limit
Type I Parts Finished	2444080	Infinite
Type II Parts Finished	2438570	Infinite
Type I Batches Finishe	877	Infinite
Type II Batches Finish	876	Infinite

OUTPUTS

Identifier	Value
Type I ESFT	17.003
Type II ESFT	19.894
Overall ESFT	18.448
Mean Tardiness I	.63294
Mean Tardiness II	3.6796
Mean Tardiness Avg	2.1554
No SF Lateness	.91227
Adj 10% Lateness	-8.464E-01
Adj 20% Lateness	-2.605E+00
Flow Allowance I	18.459
Flow Allowance II	16.713
Percent Tardy	50.830
WIP in system	17445.
Shop Avg Util	61.778
End of Rep	42708.

Execution time: 0.00 minutes.
Simulation run complete.

SIMAN V - License #9510000
Roger Mattar

Summary for Replication 1 of 1

Project: Modified Set, End Item Due Date   Run execution date : 2/27/1997
Analyst: RM   Model revision date: 2/27/1997

Replication ended at time : 42708.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
Saw FT Type I	--	--	--	--	0
Shear FT Type I	2.7841	.39042	.00000	7.7823	877
Brake FT Type I	6.4544	.77189	.46243	24.730	876
Punch FT Type I	3.4643	.55244	.37528	13.347	876
Weld FT Type I	5.4998	.73841	.60300	22.931	876
Paint FT Type I	2.5122	.43488	.34347	8.0154	877
Package FT Type I	3.5786	.43089	.30963	11.980	877
C1 Wait for Match	3.1162	1.2557	.00000	20.050	876
C2 Wait for Match	.83695	1.8209	.00000	10.523	876
C3 Wait for Match	--	--	--	--	0
C4 Wait for Match	--	--	--	--	0
Type I FT	17.935	.31565	3.1432	38.134	877
Overall FT	18.662	.30774	3.1432	38.134	1754
Type II FT	19.389	.29582	7.4288	37.418	877
Saw FT Type II	2.2862	.32230	.22499	4.9262	875
Shear FT Type II	2.6684	.36415	.51736	8.7512	877
Brake FT Type II	5.2381	.58630	.34930	20.114	875
Punch FT Type II	3.5697	.53307	.42159	13.948	876
Weld FT Type II	--	--	--	--	0
Paint FT Type II	2.7815	.43829	.21836	7.5142	877
Package FT Type II	2.8723	.50620	.19637	9.5642	878
Stn1 Cum.FT	2.2862	.32230	.22499	4.9262	875
Stn2 Cum.FT	3.8691	.41389	.00000	10.825	1754
Stn3 Cum.FT	10.106	.59194	.46243	28.103	1751
Stn4 Cum.FT	7.3877	.36531	2.3749	19.326	1752
Stn5 Cum.FT	11.867	.38397	4.9390	27.568	876
Stn6 Cum.FT	15.442	.33417	1.4568	33.413	1754
Stn7 Cum.FT	18.654	.30833	3.1432	38.134	1755
Mean Tardiness I	1.3707	1.9094	.00000	17.134	877
Mean Tardiness II	2.8247	1.1266	.00000	17.098	877
Type I ESFT	17.878	.20151	9.5111	28.443	877
Type II ESFT	19.315	.18915	9.9450	29.912	877
Overall ESFT	18.597	.19883	9.5111	29.912	1754
No SF Lateness	.84893	4.9890	-9.873E+00	17.134	1754
Mean Tardiness Avg	2.0977	1.4311	.00000	17.134	1754
Adj 10% Lateness	-9.325E-01	-4.457E+00	-1.197E+01	15.066	1754
Flow Allowance I	18.509	.13746	10.000	21.000	877
Flow Allowance II	17.117	.16401	9.0000	21.000	877
Adj 20% Lateness	-2.714E+00	-1.509E+00	-1.407E+01	13.034	1754

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Saw Util	32.591	1.4381	.00000	100.00	.00000
Shear Util	69.739	.65872	.00000	100.00	.00000
Brake Util	84.783	.42364	.00000	100.00	100.00
Punch Util	76.936	.54751	.00000	100.00	100.00
Weld Util	33.989	1.3936	.00000	100.00	.00000
Paint Util	61.961	.78352	.00000	100.00	.00000
Package Util	72.604	.61426	.00000	100.00	100.00
Machinist Util	66.012	.30094	.00000	100.00	50.000
Welder Util	33.989	1.3936	.00000	100.00	.00000
Painter Util	61.961	.78352	.00000	100.00	.00000
Packager Util	72.604	.61426	.00000	100.00	100.00
Shop Avg Util	61.801	.23876	.00000	100.00	42.857
Saw Q Length	.00000	--	.00000	.00000	.00000
Shear Q Length	.08168	3.3734	.00000	2.0000	.00000
Brake Q Length	.82024	1.3752	.00000	6.0000	.00000
Punch Q Length	.23472	2.1841	.00000	4.0000	.00000
Welder Q Length	2.8555E-04	59.169	.00000	1.0000	.00000
Paint Q Length	.13675	2.5582	.00000	2.0000	.00000
Package Q Length	.19641	2.2413	.00000	3.0000	.00000
Match Q1 Length	.44475	1.4837	.00000	3.0000	.00000
Match Q2 Length	.11945	2.7557	.00000	2.0000	.00000
Percent Tardy	54.661	.10526	.00000	82.812	54.332
C1P1 in system	7364.3	.46129	.00000	17490.	4850.0
C2 in system	3044.6	.83337	.00000	12600.	.00000
P2 in system	7988.4	.44273	.00000	19790.	2370.0
Total WIP	18397.	.47094	.00000	46810.	7220.0

COUNTERS

Identifier	Count	Limit
Type I Parts Finished	2444080	Infinite
Type II Parts Finished	2440730	Infinite
Type I Batches Finishe	877	Infinite
Type II Batches Finish	877	Infinite

OUTPUTS

Identifier	Value
Type I ESFT	17.878
Type II ESFT	19.315
Overall ESFT	18.597
Mean Tardiness I	1.3707
Mean Tardiness II	2.8247
Mean Tardiness Avg	2.0977
No SF Lateness	.84893
Adj 10% Lateness	-9.325E-01
Adj 20% Lateness	-2.714E+00
Flow Allowance I	18.509
Flow Allowance II	17.117
Percent Tardy	54.661
WIP in system	18397.
Shop Avg Util	61.801
End of Rep	42708.

Execution time: 0.00 minutes.
Simulation run complete.

SIMAN V - License #9510000
Roger Mattar

Summary for Replication 1 of 1

Project: Original Set, Component Due Date   Run execution date : 2/28/1997
Analyst: RM   Model revision date: 2/28/1997

Replication ended at time : 41343.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
Saw FT Type I	2.3841	.39827	.34212	6.5498	681
Shear FT Type I	2.5998	.40023	.25327	7.2619	681
Brake FT Type I	4.5176	.65201	.00000	16.717	681
Punch FT Type I	3.2313	.55138	.28702	13.988	681
Weld FT Type I	3.7975	.61801	.44286	14.242	680
Paint FT Type I	2.3226	.37581	.32167	6.1074	681
Package FT Type I	3.5433	.40891	.16552	8.9903	682
C1 Wait for Match	1.0723	1.7271	.00000	11.723	681
C2 Wait for Match	.89600	1.3868	.00000	7.7950	681
C3 Wait for Match	1.3484	1.5244	.00000	12.004	680
C4 Wait for Match	.72385	1.6194	.00000	6.3532	680
Type I FT	16.393	.29326	3.1432	31.205	682
Overall FT	17.031	.29154	3.1432	33.699	1363
Type II FT	17.670	.28532	4.4557	33.699	681
Saw FT Type II	3.2859	.39135	.60978	8.3852	681
Shear FT Type II	3.6710	.39988	.41328	9.8097	681
Brake FT Type II	5.5982	.59497	.92456	19.249	680
Punch FT Type II	3.7879	.50216	.40515	13.254	681
Weld FT Type II	4.2119	.58636	.38292	15.188	680
Paint FT Type II	2.4078	.34590	.36330	5.5979	680
Package FT Type II	2.7709	.48941	.45739	8.4679	681
Stn1 Cum.FT	2.8350	.42913	.34212	8.3852	1362
Stn2 Cum.FT	3.1354	.44016	.25327	9.8097	1362
Stn3 Cum.FT	7.8917	.46473	1.9185	22.377	1361
Stn4 Cum.FT	6.6451	.38472	2.0135	16.546	1362
Stn5 Cum.FT	11.546	.33826	4.3269	26.554	1360
Stn6 Cum.FT	13.894	.31001	1.4568	28.895	1361
Stn7 Cum.FT	17.031	.29154	3.1432	33.699	1363
Mean Tardiness I	.78856	2.0577	.00000	10.205	682
Mean Tardiness II	2.2057	1.1418	.00000	13.159	681
Type I ESFT	16.356	.18821	9.5769	24.514	682
Type II ESFT	17.621	.18501	9.7169	26.472	681
Overall ESFT	16.988	.19024	9.5769	26.472	1363
Lateness	.36375	9.1646	-8.946E+00	13.159	1363
Mean Tardiness Avg	1.4966	1.4919	.00000	13.159	1363
Adj 10% Lateness	-1.303E+00	-2.514E+00	-1.105E+01	11.105	1363
Flow Allowance I	17.379	.15989	10.000	21.000	682
Flow Allowance II	15.955	.18800	9.0000	21.000	681
Adj 20% Lateness	-2.970E+00	-1.093E+00	-1.315E+01	9.0511	1363

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Saw Util	65.372	.72780	.00000	100.00	100.00
Shear Util	68.680	.67529	.00000	100.00	100.00
Brake Util	84.834	.42281	.00000	100.00	100.00
Punch Util	76.206	.55876	.00000	100.00	.00000
Weld Util	70.547	.64614	.00000	100.00	100.00
Paint Util	62.582	.77324	.00000	100.00	100.00
Package Util	73.285	.60376	.00000	100.00	.00000
Machinist Util	73.773	.31124	.00000	100.00	75.000
Welder Util	70.547	.64614	.00000	100.00	100.00
Painter Util	62.582	.77324	.00000	100.00	100.00
Packager Util	73.285	.60376	.00000	100.00	.00000
Shop Avg Util	71.644	.24531	.00000	100.00	71.428
Saw Q Length	.15551	2.3580	.00000	2.0000	1.0000
Shear Q Length	.20818	1.9905	.00000	2.0000	1.0000
Brake Q Length	.59429	1.4640	.00000	4.0000	.00000
Punch Q Length	.23942	2.0984	.00000	3.0000	.00000
Welder Q Length	.09099	3.2393	.00000	2.0000	.00000
Paint Q Length	.04892	4.4211	.00000	2.0000	.00000
Package Q Length	.16880	2.3552	.00000	2.0000	.00000
Match Q1 Length	.34529	1.8372	.00000	4.0000	1.0000
Match Q2 Length	.23096	1.9496	.00000	2.0000	.00000
Percent Tardy	48.884	.12580	.00000	64.814	50.476
C1P1 in System	6408.6	.48346	.00000	17110.	6030.0
C2 in system	3224.5	.61216	.00000	10310.	3120.0
C3P2 in system	6942.0	.46254	.00000	17360.	8720.0
C4 in system	3960.1	.56873	.00000	13660.	5890.0
Total WIP	20535.	.46432	.00000	57020.	23760.

COUNTERS

Identifier	Count	Limit
Type I Parts Finished	1901020	Infinite
Type II Parts Finished	1896330	Infinite
Type I Batches Finishe	682	Infinite
Type II Batches Finish	681	Infinite

OUTPUTS

Identifier	Value
Type I ESFT	16.356
Type II ESFT	17.621
Overall ESFT	16.988
Mean Tardiness I	.78856
Mean Tardiness II	2.2057
Mean Tardiness Avg	1.4966
Lateness	.36375
Adjusted Lateness	-1.303E+00
Flow Allowance I	17.379
Flow Allowance II	15.955
Percent Tardy	48.884
WIP in system	20535.
Shop Avg Util	71.644
End of Rep	41343.

Execution time: 0.00 minutes.
Simulation run complete.

SIMAN V - License #9510000
Roger Mattar

Summary for Replication 1 of 1

Project: Original Set, End Item Due Date   Run execution date : 2/28/1997
Analyst: RM   Model revision date: 2/26/1997

Replication ended at time : 42708.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
Saw FT Type I	2.5107	.43304	.26094	7.8415	876
Shear FT Type I	2.6074	.37238	.00000	6.6015	876
Brake FT Type I	5.6478	.81354	.00000	23.444	876
Punch FT Type I	3.2877	.60915	.37542	16.485	875
Weld FT Type I	5.0046	.87658	.40886	23.760	875
Paint FT Type I	2.2814	.37379	.27178	6.0359	876
Package FT Type I	3.5544	.40136	.16552	9.1053	876
C1 Wait for Match	2.2575	1.7581	.00000	21.071	875
C2 Wait for Match	.95304	1.4451	.00000	6.9519	875
C3 Wait for Match	2.1471	1.6450	.00000	19.911	875
C4 Wait for Match	.91974	1.5683	.00000	7.0267	875
Type I FT	17.733	.33941	3.1432	35.593	876
Overall FT	18.272	.32099	3.1432	36.264	1752
Type II FT	18.810	.30087	4.4557	36.264	876
Saw FT Type II	3.3837	.36750	.50381	8.3966	876
Shear FT Type II	3.7208	.35469	.68150	8.6808	875
Brake FT Type II	6.4499	.70815	.68050	22.700	875
Punch FT Type II	3.8659	.55857	.48979	19.994	875
Weld FT Type II	5.0166	.77355	.93094	21.791	875
Paint FT Type II	2.3908	.35018	.07827	6.0571	875
Package FT Type II	2.7442	.47616	.32808	8.1312	876
Stn1 Cum.FT	2.9472	.42299	.26094	8.3966	1752
Stn2 Cum.FT	3.1638	.40616	.00000	8.6808	1751
Stn3 Cum.FT	8.9957	.56453	2.0426	28.064	1751
Stn4 Cum.FT	6.7410	.38847	1.5789	21.675	1750
Stn5 Cum.FT	12.813	.39729	4.6875	31.700	1750
Stn6 Cum.FT	15.137	.35687	1.4568	33.612	1751
Stn7 Cum.FT	18.272	.32099	3.1432	36.264	1752
Mean Tardiness I	1.5277	1.9703	.00000	14.593	876
Mean Tardiness II	2.5057	1.2896	.00000	15.264	876
Type I ESFT	17.706	.22635	9.5769	29.766	876
Type II ESFT	18.771	.20185	9.7169	29.832	876
Overall ESFT	18.238	.21575	9.5769	29.832	1752
Lateness	.65464	6.7643	-1.196E+01	15.264	1752
Mean Tardiness Avg	2.0167	1.5668	.00000	15.264	1752
Adj 10% Lateness	-1.107E+00	-3.947E+00	-1.406E+01	13.164	1752
Flow Allowance I	18.176	.14921	10.000	21.000	876
Flow Allowance II	17.058	.18068	9.0000	21.000	876
Adj 20% Lateness	-2.869E+00	-1.509E+00	-1.616E+01	11.064	1752

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Saw Util	65.179	.73091	.00000	100.00	.00000
Shear Util	68.852	.67260	.00000	100.00	100.00
Brake Util	85.057	.41914	.00000	100.00	100.00
Punch Util	76.030	.56149	.00000	100.00	100.00
Weld Util	70.495	.64694	.00000	100.00	.00000
Paint Util	62.190	.77972	.00000	100.00	100.00
Package Util	73.347	.60280	.00000	100.00	100.00
Machinist Util	73.779	.30879	.00000	100.00	75.000
Welder Util	70.495	.64694	.00000	100.00	.00000
Painter Util	62.190	.77972	.00000	100.00	100.00
Packager Util	73.347	.60280	.00000	100.00	100.00
Shop Avg Util	71.593	.24044	.00000	100.00	71.428
Saw Q Length	.18947	2.0893	.00000	2.0000	.00000
Shear Q Length	.21464	1.9250	.00000	2.0000	.00000
Brake Q Length	.87516	1.4559	.00000	7.0000	.00000
Punch Q Length	.25976	2.2026	.00000	5.0000	.00000
Welder Q Length	.09571	3.2239	.00000	2.0000	.00000
Paint Q Length	.04471	4.6346	.00000	2.0000	.00000
Package Q Length	.16569	2.4144	.00000	3.0000	.00000
Match Q1 Length	.62791	1.7819	.00000	6.0000	.00000
Match Q2 Length	.26717	1.8424	.00000	3.0000	1.0000
Percent Tardy	51.152	.10747	.00000	64.062	51.655
C1P1 in System	6919.9	.51115	.00000	19770.	4720.0
C2 in system	3762.7	.66593	.00000	13590.	2460.0
C3P2 in system	7309.3	.47598	.00000	18970.	4430.0
C4 in system	4424.0	.58613	.00000	14220.	2310.0
Total WIP	22416.	.49668	.00000	61160.	13920.

COUNTERS

Identifier	Count	Limit
Type I Parts Finished	2441830	Infinite
Type II Parts Finished	2443290	Infinite
Type I Batches Finishe	876	Infinite
Type II Batches Finish	876	Infinite

OUTPUTS

Identifier	Value
Type I ESFT	17.706
Type II ESFT	18.771
Overall ESFT	18.238
Mean Tardiness I	1.5277
Mean Tardiness II	2.5057
Mean Tardiness Avg	2.0167
Lateness	.65464
Adjusted Lateness	-1.107E+00
Flow Allowance I	18.176
Flow Allowance II	17.058
Percent Tardy	51.152
WIP in system	22416.
Shop Avg Util	71.593
End of Rep	42708.

Execution time: 0.00 minutes.
Simulation run complete.

SIMAN V - License #9210467
Roger Mattar

Summary for Replication 1 of 1

Project: Experiment 01
Analyst: RM

Run execution date : 4/ 3/1997
Model revision date: 4/ 1/1997

Replication ended at time : 52823.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
Saw FT Type I	--	--	--	--	0
Shear FT Type I	2.4088	.39168	.32648	6.0284	2321
Brake FT Type I	2.3576	.45640	.08409	9.1620	2320
Punch FT Type I	2.2518	.37130	.22661	6.1783	2320
Weld FT Type I	2.4593	.50208	.00000	9.6095	2320
Paint FT Type I	1.8547	.40221	.00882	6.2070	2321
Package FT Type I	2.4194	.36044	.17932	6.8786	2321
C1 Wait for Match	.65649	1.5371	.00000	6.6712	2320
C2 Wait for Match	.65283	1.4102	.00000	6.5981	2320
C3 Wait for Match	--	--	--	--	0
C4 Wait for Match	--	--	--	--	0
Type I FT	11.388	.21098	3.1432	21.773	2321
Overall FT	12.296	.21955	3.1432	23.664	4642
Type II FT	13.205	.20258	6.7872	23.664	2321
Saw FT Type II	1.7435	.31453	.26193	3.8782	2320
Shear FT Type II	2.3361	.41002	.00000	6.6336	2322
Brake FT Type II	2.9144	.39365	.14320	8.3236	2320
Punch FT Type II	2.3321	.36279	.19936	6.4169	2320
Weld FT Type II	--	--	--	--	0
Paint FT Type II	1.9635	.40579	.07861	5.4722	2322
Package FT Type II	1.9222	.42759	.24745	6.4386	2322
Stn1 Cum.FT	1.7435	.31453	.26193	3.8782	2320
Stn2 Cum.FT	3.2445	.42241	.32648	8.7973	4643
Stn3 Cum.FT	5.8420	.65532	.08409	17.438	4640
Stn4 Cum.FT	5.5364	.31405	1.3792	13.215	4640
Stn5 Cum.FT	7.1202	.22620	3.2835	15.200	2320
Stn6 Cum.FT	10.127	.24211	1.4568	20.253	4643
Stn7 Cum.FT	12.295	.21975	3.1432	23.664	4643
Mean Tardiness I	.70841	1.4925	.00000	8.3334	2321
Mean Tardiness II	.81560	1.3246	.00000	7.2809	2321
Type I ESFT	11.382	.10590	8.9534	14.374	2321
Type II ESFT	13.189	.10286	9.9450	16.563	2321
Overall ESFT	12.286	.12774	8.9534	16.563	4642
No SF Lateness	.08451	21.202	-6.857E+00	8.3334	4642
Mean Tardiness Avg	.76200	1.4043	.00000	8.3334	4642
Adj 10% Lateness	-1.137E+00	-1.598E+00	-7.857E+00	6.9894	4642
Flow Allowance I	11.341	.15902	7.4100	16.510	2321
Flow Allowance II	13.083	.15747	8.0800	18.910	2321
Adj 20% Lateness	-2.358E+00	-7.907E-01	-9.147E+00	5.6454	4642

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Saw Util	24.873	1.7379	.00000	100.00	.00000
Shear Util	52.385	.95338	.00000	100.00	.00000
Brake Util	63.938	.75100	.00000	100.00	100.00
Punch Util	58.247	.84664	.00000	100.00	100.00
Weld Util	25.717	1.6995	.00000	100.00	.00000
Paint Util	47.206	1.0575	.00000	100.00	.00000
Package Util	55.456	.89623	.00000	100.00	100.00
Machinist Util	49.861	.37246	.00000	100.00	50.000
Welder Util	25.717	1.6995	.00000	100.00	.00000
Painter Util	47.206	1.0575	.00000	100.00	.00000
Packager Util	55.456	.89623	.00000	100.00	100.00
Shop Avg Util	46.832	.26862	.00000	85.714	42.857
Saw Q Length	1.4407E-04	83.306	.00000	1.0000	.00000
Shear Q Length	.15389	2.3455	.00000	2.0000	.00000
Brake Q Length	.11319	2.8613	.00000	2.0000	.00000
Punch Q Length	.07194	3.5918	.00000	1.0000	1.0000
Welder Q Length	1.7167E-04	76.316	.00000	1.0000	.00000
Paint Q Length	.07332	3.5600	.00000	2.0000	.00000
Package Q Length	.06578	3.8079	.00000	3.0000	1.0000
Match Q1 Length	.09371	3.1112	.00000	2.0000	.00000
Match Q2 Length	.09319	3.1194	.00000	1.0000	.00000
Percent Tardy	51.636	.06096	.00000	52.688	52.261
C1P1 in system	3485.0	.42606	.00000	10910.	3800.0
C2 in system	921.38	1.1802	.00000	5440.0	2010.0
P2 in system	4035.3	.39713	.00000	10610.	3920.0
Total WIP	8441.6	.35897	.00000	21380.	9730.0

COUNTERS

Identifier	Count	Limit
Type I Parts Finished	4857130	Infinite
Type II Parts Finished	4851190	Infinite
Type I Batches Finishe	2321	Infinite
Type II Batches Finish	2321	Infinite

OUTPUTS

Identifier	Value
Type I ESFT	11.382
Type II ESFT	13.189
Overall ESFT	12.286
Mean Tardiness I	.70841
Mean Tardiness II	.81560
Mean Tardiness Avg	.76200
No SF Lateness	.08451
Adj 10% Lateness	-1.137E+00
Adj 20% Lateness	-2.358E+00
Flow Allowance I	11.341
Flow Allowance II	13.083
Percent Tardy	51.636
WIP in system	8441.6
Shop Avg Util	46.832
End of Rep	52823.

Execution time: 0.00 minutes.
Simulation run complete.

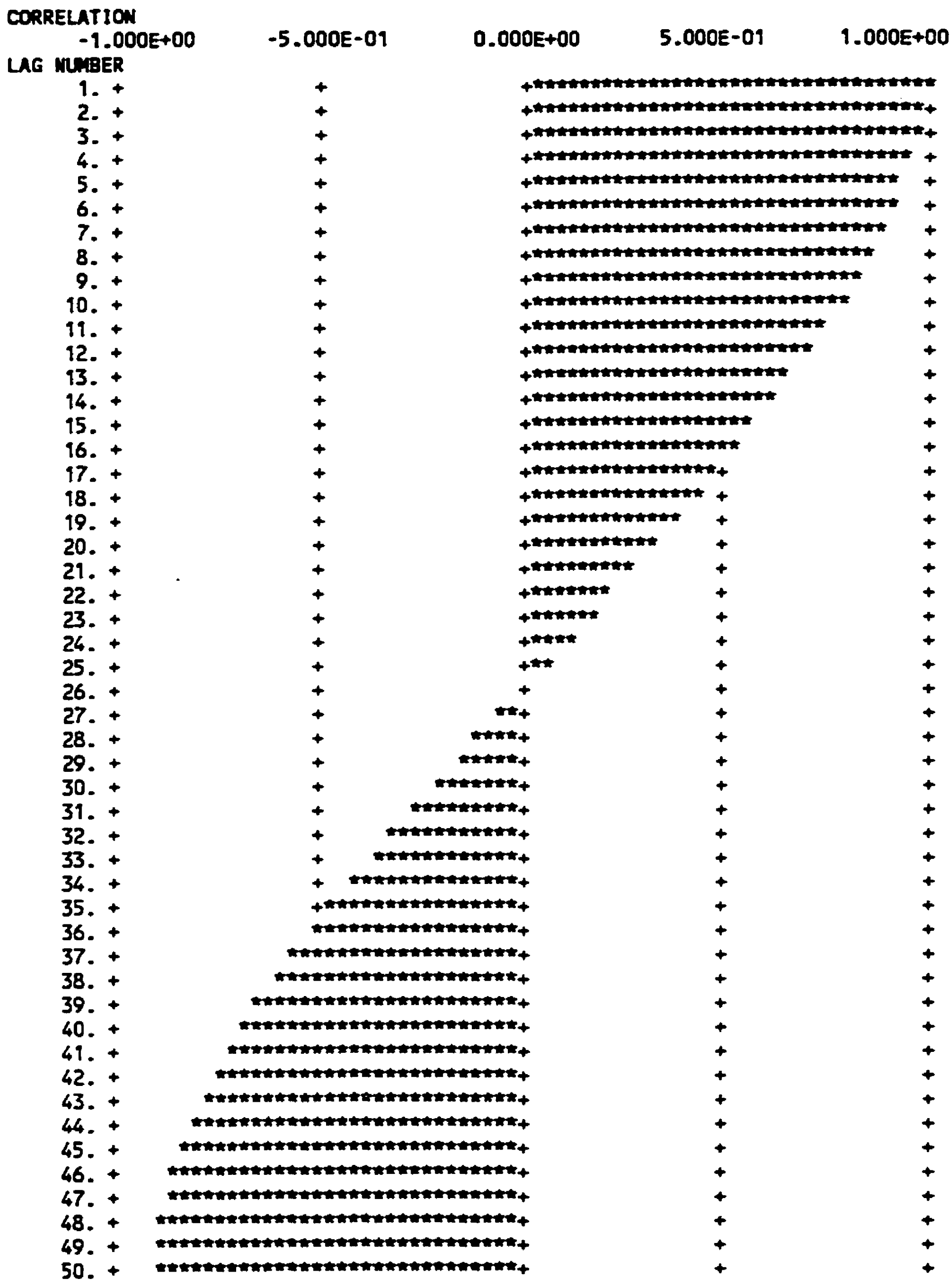
CORRELOGRAM : CRRLGRAM.01

LAG	COVARIANCE	CORRELATION
1	2.7970	.9910
2	2.7737	.9828
3	2.7243	.9653
4	2.6811	.9500
5	2.6139	.9262
6	2.5532	.9047
7	2.4695	.8750
8	2.3925	.8477
9	2.2934	.8126
10	2.2021	.7803
11	2.0898	.7405
12	1.9862	.7038
13	1.8618	.6597
14	1.7462	.6187
15	1.6106	.5707
16	1.4848	.5261
17	1.3397	.4747
18	1.2062	.4274
19	1.0550	.3738
20	.91603	.3246
21	.75994	.2693
22	.61693	.2186
23	.45849	.1625
24	.31385	.1112
25	.15438	.0547
26	9.53379E-03	.0034
27	-.15018	-.0532
28	-.29475	-.1044
29	-.45213	-.1602
30	-.59340	-.2103
31	-.74644	-.2645
32	-.88210	-.3125
33	-1.0298	-.3649
34	-1.1598	-.4109
35	-1.3006	-.4608
36	-1.4227	-.5041
37	-1.5548	-.5509
38	-1.6669	-.5906
39	-1.7878	-.6335
40	-1.8877	-.6689
41	-1.9947	-.7068
42	-2.0793	-.7367
43	-2.1710	-.7692
44	-2.2401	-.7937
45	-2.3150	-.8203
46	-2.3664	-.8385
47	-2.4235	-.8587
48	-2.4572	-.8706
49	-2.4962	-.8845
50	-2.5112	-.8898

SUMMARY STATISTICS

SAMPLE MEAN	:	12.30
SAMPLE VARIANCE	:	2.822
SAMPLE SIZE	:	4226
WEIGHTED SUM OF COV. (CSUM)	:	10.81

CORRELOGRAM : CRRLGRAM.01



PAIRED-T MEANS COMPARISON : EXP01_VS_EXP03

IDENTIFIER	ESTD. MEAN DIFFERENCE	STANDARD DEVIATION	.950 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS
0.03-0.04	-.353	.163	.761E-01	.599	.899	20
				.843	1.41	20

REJECT H0 => MEANS ARE NOT EQUAL AT .050 LEVEL

MEAN-DIFFERENCE INTERVALS :
EXP01_VS_EXP03

```
                   -.353                                  0
0.03-0.04        (---------X---------)---------------------------------|
               -.430          -.277
```

```
---------------------------------------------------------------------
| | = TEST ( 0 OR 1 )  ( = LOWER 95% CL  X = AVERAGE  ) = UPPER 95% CL |
---------------------------------------------------------------------
```

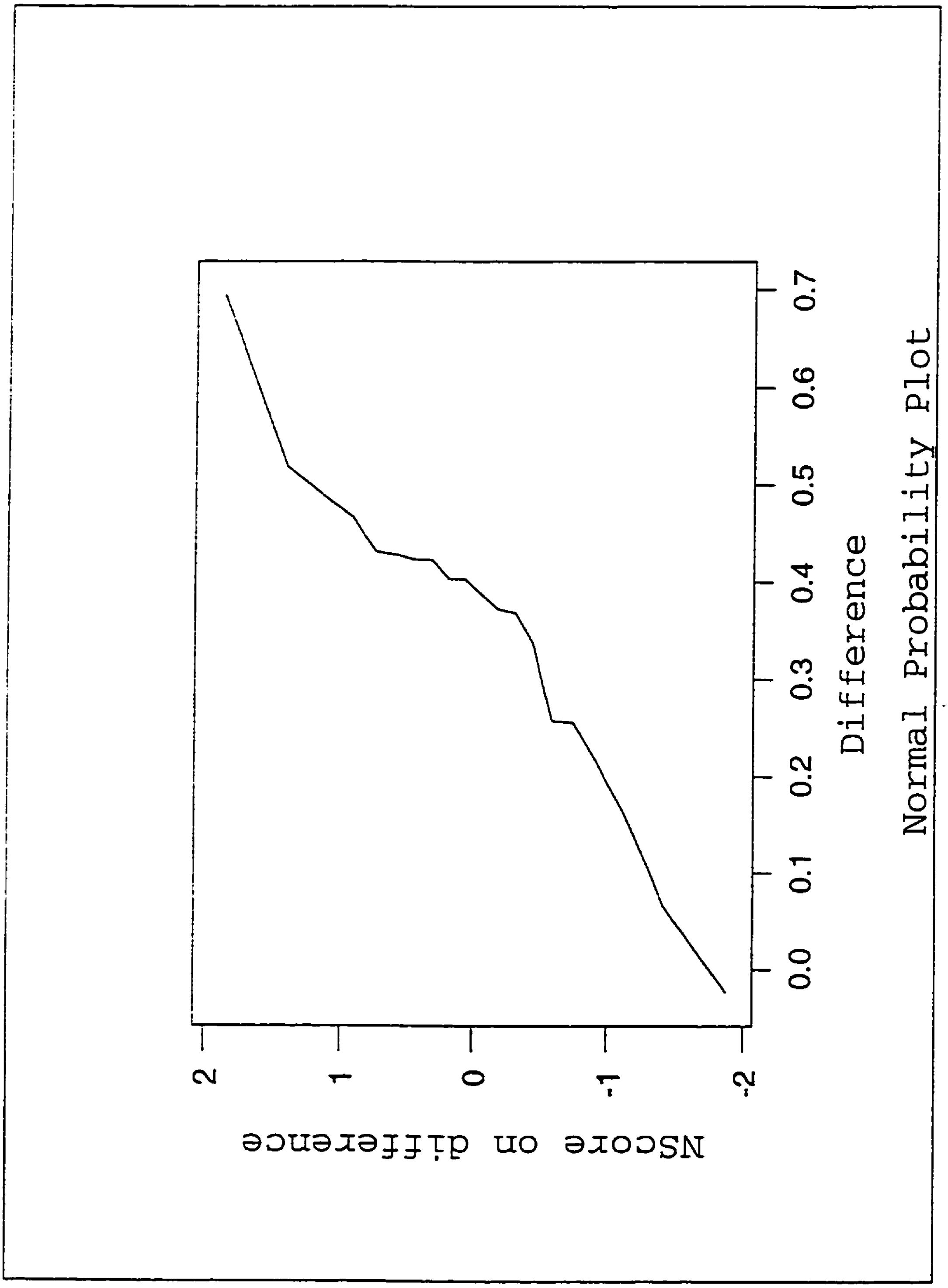
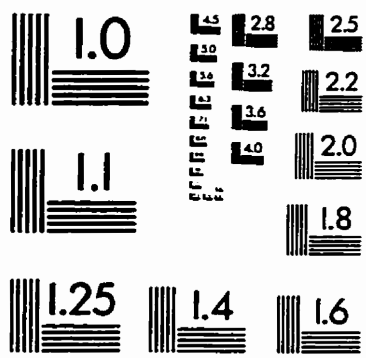

NScore on difference
2
1
0
-1
-2
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Difference
Normal Probability Plot

IMAGE EVALUATION TEST TARGET (QA-3)



150mm

6"

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989