# A COMPARATIVE SIMULATION STUDY OF KANBAN, CONWIP, AND MRP MANUFACTURING CONTROL SYSTEMS IN A FLOWSHOP

By

THOMAS ALFONS HOCHREITER

A THESIS PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

1999

To Mum, Dad, and Sis

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Suleyman Tufekci for his assistance and guidance. As my advisor and the chairman of my supervisory committee he has provided the constructive critique to refine the content of this thesis.

Further, I would like to acknowledge the contribution of the other members of my committee, Dr. Diane Schaub and Dr. Sherman Bai. During her course in Applied Probability, Dr. Schaub improved my understanding of statistics while Dr. Bai provided me with the required background on digital simulation.

In addition, I would like to thank Tim Elftman for the professional administration of the computer network and Tobias and Henrik Andersson for their initial support on the material.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

A COMPARATIVE SIMULATION STUDY OF
KANBAN, CONWIP, AND MRP MANUFACTURING
CONTROL SYSTEMS IN A FLOWSHOP

By

Thomas Alfons Hochreiter

May, 1999

Chairman: Dr. Suleyman Tufekci
Major Department: Industrial & Systems Engineering

The globalization of markets due to the improvement of communication and transportation media has had a significant impact on manufacturing technology in recent years. The strong international competition forced companies to establish efficient production facilities ensuring profitability on the long run. The performance of the most prevalent American manufacturing control mechanism, MRP, was questioned after the success of the Japanese Kanban control system during the Just-In-Time era. CONWIP, a generalization of the Kanban control system, was introduced as a result of extensive research done to understand manufacturing systems with the aim of improving their efficiency.

During an extensive simulation study, the performances of Kanban, CONWIP, and MRP were evaluated for a ten identical machine tandem line with respect to batch size, setup time, and machine failure. The utilization (throughput) was kept constant for all control systems. The parameters were introduced to the models one at a time,

thereby increasing the realism and the variability of the manufacturing line. Thus, the performances of the three control mechanisms were explored on three levels of complexity. Initially, only the influence of batch size on the performances of the control systems was investigated. Then, the setup time was taken into consideration in addition to the batch size. Last, machine failure was introduced to augment the models' realism resulting in a higher practical applicability. On each level, the performances were evaluated for steady-state, assuming the manufacturing line would run indefinitely. In addition, the response of the performance to machine failure was observed dynamically while keeping batch size and setup time constant.

Although the performance differences were found to be minute, Kanban and CONWIP were outperformed by the traditional control system, MRP, for experiments with varying batch size and for experiments including both batch size and setup times. On the highest level of variability, with machine failure introduced, Kanban was ranked first, closely followed by CONWIP. The two pull systems easily outranked the push system when evaluated according to average cycle time, maximum cycle time and the standard deviation of cycle time. Kanban performed best for the dynamic response to failure as well, where the system performance was measured by the time taken to recover from failure.

# CHAPTER 1
# INTRODUCTION

## 1.1 Motivation

Primarily due to rapid development of technology in the past thirty years, the market structure throughout the world has changed considerably. Local markets have become accessible to foreign investors, who are not only able to perform well in their newly established territory, but, who are even able to excel because of superior technology. Successful companies embedded globalization in their expansion strategies, consistently seeking for new markets abroad. Consequently, manufacturing companies are facing global competition, forcing them to keep up with new concepts and even to proactively incorporate improvement into their daily production routine.

In 1972 the American Production and Inventory Control Society (APICS) strongly promoted material requirements planning (MRP) in an effort to strengthen the American manufacturing industry and its standing in the international arena. MRP was hoisted to the most prevalent production control system on a national level. After the successes of Just-In-Time (JIT) its dominant appearance in industry was questioned. The Japanese had introduced their superior products manufactured with the support of the Kanban control system enhancing their global competitiveness. An enormous amount of research was directed towards the new system giving rise to a rich body of literature documenting various concepts.

In 1990 another system, striving to maintain a constant work in process (CONWIP), was presented, able to prove its usefulness in theory and in industry. The extensive research produced ample knowledge of system's behavior and good understanding of the factors involved. The newly evolved science, Factory Physics, attempts to describe and formalize the characteristics of the extreme probabilistic systems.

However, the models analyzing and comparing the different control systems analytically are based on too many simplifying and unrealistic assumptions. The results can merely serve as approximations of real systems, a very limiting attribute for their practical applicability. Simulation has established itself as a very powerful alternative to the analytical modeling process. With the reduction in computer hardware prices and the increase of processor speed, simulation has become a popular tool in recent years. It enables modeling with great precision resulting in a very good representation of real systems and trustworthy output data. The simulations software available allows the study of manufacturing systems dynamically, giving the analyst a feeling for the system in addition to generating realistic results.

In this research paper the three control systems Kanban, CONWIP, and MRP are analyzed by means of a comparative simulation study. Ever since the introduction of Kanban to the world of production, MRP has been discredited as an inferior control system. However, despite its significant success, Kanban is not flawless. CONWIP is investigated as a highly praised alternative. An evaluation of their performance with respect to batch size, setup time and failure should unveil the superior control system for the chosen manufacturing line.

## 1.2 Thesis Outline

Chapter 2 highlights the mechanisms and characteristics of the control systems, Kanban, CONWIP, and MRP. A comparison regarding specific attributes reveals basic differences that support the existence of all three control systems. Chapter 3 introduces simulation as the alternative to analytical modeling of manufacturing systems. It denotes the important aspects of a simulation study. Chapter 4 serves as a reference to both, statistical analysis methods unique to simulation, and methods common to general data interpretation. In Chapter 5, the influence of batch size on the performance of the control systems is demonstrated. In Chapter 6, setup time is included in the investigations. Chapter 7 deals with the manufacturing system with the highest degree of realism, including batch size, setup time and failure. The response of the system to failure dependent on time is analyzed as well. Chapter 8, summarizes calculations performed to ensure a high accuracy of the output data on a 95% confidence level while Chapter 9 encompasses the conclusions and suggestions for future work.

# CHAPTER 2
# CONTROL SYSTEMS

A brief theoretical background on the three manufacturing control systems is given in this Chapter. The purpose is to primarily elaborate on the characteristics unique to the individual control systems and their differences and to secondarily explain their most important mechanisms.

## 2.1 Push And Pull Systems

Spearman and Hopp [HOP96, p.316] give a very describing quote of Taiichi Ohno, the father of Just-in-Time (JIT), to distinguish the meaning of the two terms, push and pull:

Manufacturers and workplaces can no longer base production on desktop planning alone and then distribute, or *push*, them onto the market. It has become a matter of course for customers, or users, each with a different value system, to stand in the frontline of the marketplace and, so to speak, *pull* the goods they need, in the amount and at the time they need them [OHN88, xiv].

This global perspective can be applied to any individual manufacturing system. The following definition gives a general and thus abstract explanation of the words: A *push* system *schedules* the release of work based on demand, while a pull system *authorizes* the release of work based on system status [HOP96, p.317].

This means that a push system releases an entity to the line according to the exogenous master production schedule (MPS). The release time is not modified for a change in the manufacturing system [see Figure 2-1]. Information flows from the MPS downstream towards the finished goods inventory.

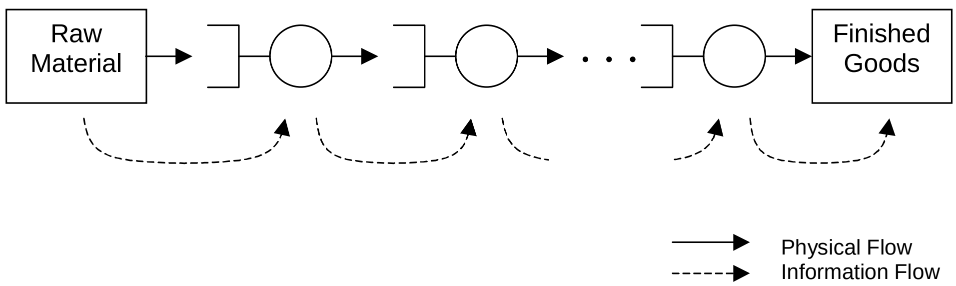


Figure 2-1: A push manufacturing system.

A pull system, however, only allows an entity to enter the system when a signal generated by a change in the line status calls for it. This change results in the most cases from the departure of an entity from the line [see Figure 2-2]. Information flows from the finished goods inventory, the customer, upstream towards the raw material inventory.

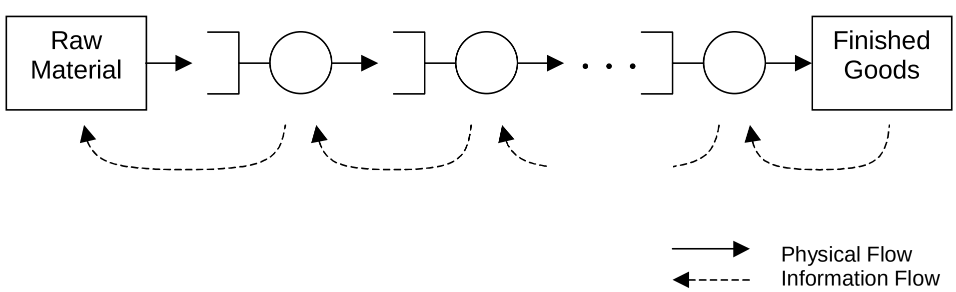


Figure 2-2: A pull manufacturing system.

The performance of the two systems is dependent on scheduling rules as well. Here the most prevalent one, fist come first serve (FCFS), will be assumed throughout. Extensive simulations done by Hum and Lee for JIT systems reveal no dominant rule. However, the results seem to indicate that FCFS is not necessarily justified, its weakness becomes most apparent under tight production conditions. According to them, the user should not arbitrarily adopt a scheduling rule. Instead, the nature of the scheduling rule and the production environment should be understood [HUM98].

As the release of material to the line is initiated by the MPS in MRP, the manufacturing system is controlled by the release rate of material resulting in a specific throughput. The pull systems on the other hand only allow material into the system when a card is liberated, a consequence of a reduction in work in process (WIP). Thus, they control the system by managing the WIP and putting an upper boundary on the material present in a line.

Kanban and CONWIP are the pull systems discussed here. Their performance will be compared with the performance of MRP, the most prevalent push system.

Before a comparison of their characteristics can be made, Kanban, CONWIP, and MRP are discussed as a basis of a practical control system in the following chapters.

## 2.2 Kanban

Mostly the Toyota-style Kanban system is discussed as a pull system and it is hardly surprising that the term *pull* is commonly viewed as synonymous with *Kanban*

[SCH82]. There is an immense Kanban literature often comparing its performance to a push system driven by unreliable demand forecasts [BER92].

In a Kanban system, production is triggered by demand. When a part is removed from the final inventory point, the last workstation in the line is given authorization to replace the part. This workstation in turn sends an authorization signal to the upstream workstation to replace the part it just used. This process continues upstream, replenishing the downstream void by requesting material from the antecedent workstation. To control information transfer, the operator requires both parts and an authorization signal, a card, to work.

### 2.2.1 The Mechanism

The Kanban system simulated here makes use of one inventory storage point and requires only one card per station. The Kanban system developed at Toyota makes use of a two-card system requiring a production card and a move card per station [see HOP96, p.163]. Figure 2-3 illustrates the one-card Kanban system.

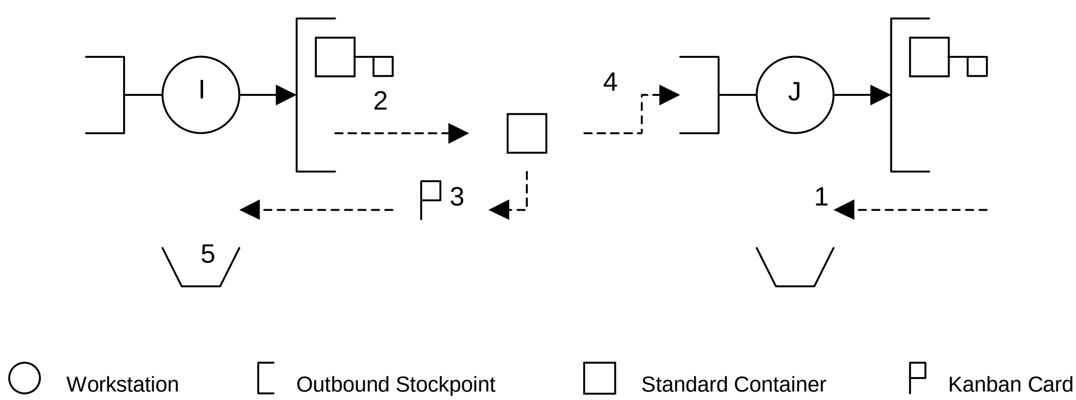


Figure 2-3: The one-card Kanban system.

The operator finds a card in the hold box at workstation J (1). He/she gets material from the outbound stockpoint of the upstream workstation I (2). The card attached to the material is removed and placed into the hold box of the upstream workstation (3). The material enters the manufacturing process and the card in the hold box is attached to the product placed in the outbound stockpoint (4). The operator at the upstream workstation I finds the card in his/her holdbox and starts processing (5). The same cycle is followed for the upstream machines until the raw material inventory is reached [see Figure 2-2]. A Kanban system can be seen as a closed queuing network with blocking. Jobs circulate around the network indefinitely. However, unlike the CONWIP system [see 2.3.1], the Kanban system limits the number of entities per workstation, since the number of production cards at a station establishes a maximum WIP level for that station. Each production cards acts exactly like a space in a finite buffer in front of the workstation. The upstream workstation is blocked when the buffer is full [HOP96, p.325].

Berkley shows that a common model of a Kanban system is equivalent to a traditional tandem production line with finite buffers. His model assumes that kanbans travel instantly to their destinations when they are detached from a part, and that the kanbans and parts travel in quantities of one [BER91]. Gstettner and Kuhn describe and classify different Kanban systems. They analyze the system with respect to production rate and average work in process [GST96].

### 2.2.2 Characteristics

As the amount of material in the system is limited to the number of cards assigned, there is a natural upper bound of material in process.

Due to the presence of the cards the involvement of the operators in controlling the flow of material is enhanced. This involvement and active participation paired with a proactive thinking enables continuous improvement not necessarily given for the push systems.

A Kanban system suits a stable material flow best. The product mix should be fairly stable and not too large as the cards are unique to certain products and expensive in their introduction to a system.

Kanban is not useful in an environment with expensive items that are rarely ordered, since it would require at least one of each kind of item to be in inventory at all times.

The performance is very sensitive to the number of cards assigned to the system and their specific allocation. Gstettner and Kuhn show that the distribution of cards has a significant effect on the performance of Kanban systems. According to them, the different types of Kanban control mechanisms show equivalent performance data, if the distribution pattern is adapted accordingly [GST96].

In most Kanban systems the number of cards assigned to specific workstations is fixed resulting in blockages or starvation. Blocking occurs when all the cards are attached to full containers in the outbound stockpoint, while starvation occurs when at least one production Kanban is in the hold box waiting for a container from the upstream workstation while the machine at that station is idle. Gupta and Al-Turki have developed an algorithm to implement a flexible Kanban system adjusting the number of cards to stochastic processing times and a variable demand environment [GUP97].

Mascolo et al. show that the performance of a multi-stage Kanban system can be derived from evaluating a set of subsystems. The subsystems result from a decomposition of the original line, where each set is being associated with a particular stage. Numerical results show that the method is fairly accurate [MAS96].

## 2.3 CONWIP

The CONWIP (*CON*stant *W*ork *I*n *P*rocess) control system strives to maintain a constant work in process. It was first introduced by Spearman et al. in 1990 and can thus be classified as a very new control concept [SPE90].

### 2.3.1 The Mechanism

CONWIP can be considered a special case of Kanban, where the entire line constitutes one workstation. Departing jobs send production cards back to the beginning of the line to authorize release of new jobs.

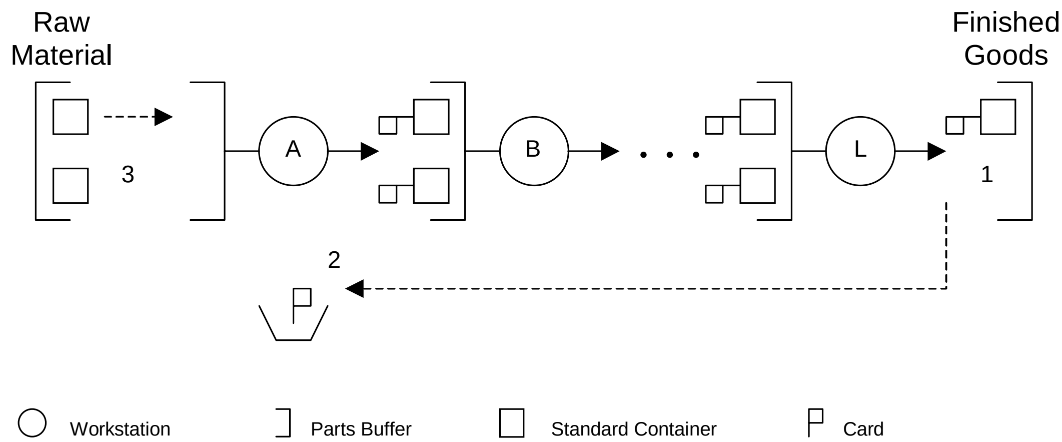


Figure 2-4: A CONWIP production line.

The finished product is taken out of the inventory that is fed by workstation L (1). The production card is sent back to workstation A to authorize the release of a new job (2). The operator at the upstream workstation A finds the card, gets the raw material from the inventory and starts processing the unit (3). In a Kanban system, each card is used to signal production of a specific part. CONWIP production cards are assigned to the production line and are not part number specific. Part numbers are assigned to the cards at the beginning of the production line. The numbers are matched with the cards by referencing a backlog list. When work is needed for the first process center in the production line [see Figure 2-4, (3)], the card is removed from the queue and marked with the first part number in the backlog number for which raw materials are present [SPE90].

Here, the following simplifying assumptions are made for CONWIP:

1. The production line consists of a single routing, along which all parts flow, and
2. WIP can be measured in units (i.e., number of jobs or parts in the line).

Spearman and Hopp [HOP96, p.324] remark that a CONWIP system resembles a closed queuing network, in which entities never leave the system, but instead circulate around the network indefinitely. In reality, the entering jobs are different from the departing jobs. Assuming that all jobs are identical, this difference does not matter for modeling purposes. Gstettner and Kuhn mention that the model developed by Spearman et al. [SPE90] can be refined and adapted to different production environments as done by Duenyas and Hopp [DUE92] and Duenyas et al. [DUE93] [GST96]. Huang and Wang show by means of simulation that the CONWIP production control system is very efficient for the production and inventory control of semi-continuous manufacturing, such as that found in a steel rolling plant [HUA97].

### 2.3.2 Characteristics

As does Kanban, CONWIP controls the total amount of work in process in the system. The WIP is limited to the number of cards assigned to the entire line instead of to the individual machines.

If a machine fails in a CONWIP line, the amount of material downstream of it will eventually be flushed out of the system by the demand process. These demand events will cause the release of new entities to the system. If the machine fails for a long period of time, these entities and the entities already in the system upstream of the failed machine will accumulate in the buffer immediately upstream of the failed machine. The release of the new jobs to the system stops once no more cards are released from entities departing the system [BON97].

There is no blocking in CONWIP lines since buffers are assumed big enough to hold all parts that circulate in the line [GST96].

In CONWIP systems information about demand is sent directly from the last to the first station. The entity goes through all the workstations in the line carrying the information about necessary production.

## 2.4 MRP

The promotion of material requirements planning (MRP) by the American Production and Inventory Control Society (APICS) in 1972 boosted this production control paradigm to the most prevalent system today. Only after the successes of JIT and Kanban its dominant appearance in industry was questioned.

### 2.4.1 The Mechanism

As can be derived from its name, MRP plans material requirements. It deals with the two dimensions of production control: quantities and timing. The system must determine appropriate production quantities of all types of items, from final products that are sold, to components used to build final products, to inputs purchased as raw materials. It must also determine production timing that facilitates meeting order due dates.

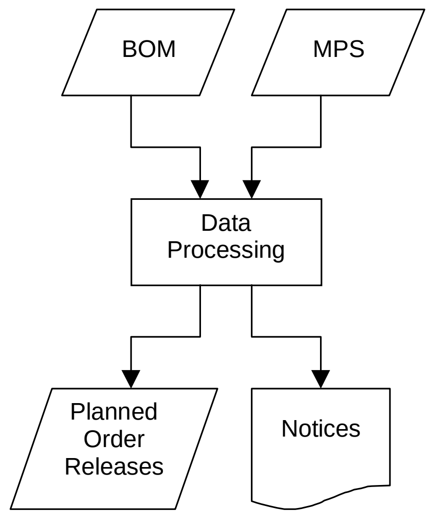


Figure 2-5: Simplified schematic of MRP.

The data from the bill of material (BOM) and the master production schedule (MPS), as the source of demand for MRP, is processed in several steps to produce the planned order releases and notices such as change notices and exception notices [see Figure 2-5]. The BOM describes the relationship between end items and lower level items while the MPS gives the quantity and due dates for all parts to obtain the gross requirements. The schematic is presented to illustrate that all the information needed for the entire manufacturing system originates from the MPS.

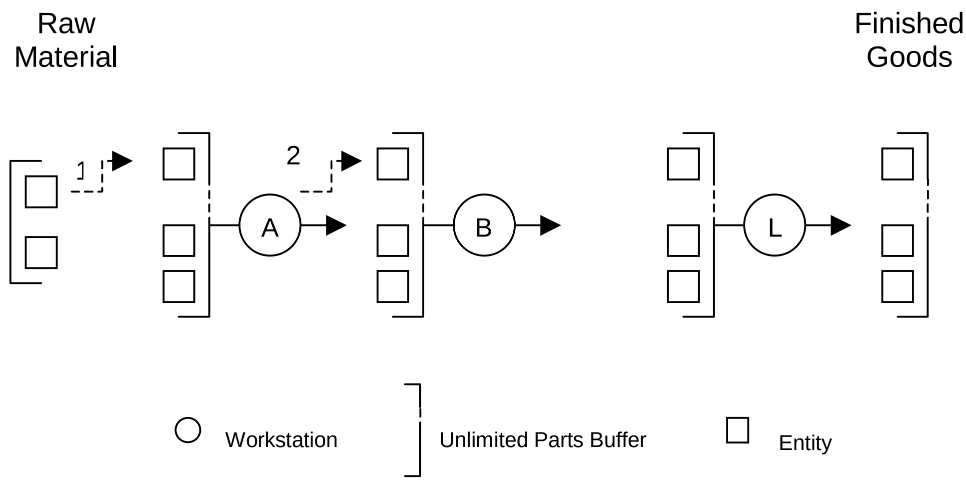


Figure 2-6: A MRP production line.

The order is released at the raw material post (1) as planned with the help of the MPS [see Figure 2-6]. As the entity is released independent of the amount of the material in the buffer preceding Workstation A, the buffer size may not be limited to a specific amount of entities. Mostly constraints are given by physical space on the manufacturing floor. When workstation A is finished with processing the entity, it pushes it on to the next workstation, B (2). This process continues downstream until the entity departs the system at the finished goods post.

To be able to address the huge problem of coordinating thousands of orders with hundreds of tools for thousands of end items made up of additional thousands of components manufacturing resources planning (MRP II) was developed [HOP96, p.143]. It provides a general control structure that breaks the production control problem into a hierarchy based on time scale and product aggregation, thus, primarily taking the capacity of the manufacturing system into account. MRP II brings together many functions to generate a truly integrated manufacturing management system including demand management, forecasting, capacity planning, rough-cut capacity planning, dispatching and input/output control.

### 2.4.2 Characteristics

MRP provides a simple method for ordering materials based on needs, as established by a master production schedule and bills of material. As such, it is well suited for use in controlling the purchasing of components. However, in the control of production MRP shows deficiencies [HOP96, p.143]. This is especially true for manufacturing systems that require proper exploitation of capacity resources by taking bottlenecks into consideration.

According to Spearman and Hopp the real reason for MRP's inability to perform well is the faulty underlying model. The key calculation is performed by using fixed lead times to derive releases from due dates. These lead times are functions of the part number only. They are not affected by the status of the plant. More importantly, the lead times do not consider the loading of the manufacturing system. An MRP system assumes that the time for a part to travel through the plant is the same whether the plant is empty or overflowing with work, which is only true for infinite capacity. Furthermore, to ensure the coordination of parts at assembly, there is a strong incentive to increase the lead times to provide a buffer against unforeseen obstructions. However, as inflating lead times introduces more material into the system, it increases congestion and consequently the cycle times. Instead of delivering on time, the products are delayed even more [HOP96, p.175].

As quoted by the APICS literature, MRP's bad performance in industry was blamed on inaccurate data, including bills of material and inventory records. MRP requires a high standard of data integrity to function properly [LAT81].

## 2.5 Comparison of CONWIP with MRP

As mentioned previously [see 2.1], a push system controls throughput and observes WIP, while a pull system controls WIP and observes throughput. WIP is directly observable, while throughput can only be determined indirectly. The jobs on a shopfloor can be physically counted and maintained according to the WIP cap. In contrast, the release rate for MRP must be set with respect to capacity. If the rate is chosen too high, the system will be congested with material resulting in high cost due to insufficient throughput and high WIP. As estimating capacity is very difficult, optimizing a push system is much more intricate [HOP96, p.325].

Concerning the efficiency, Spearman and Hopp state the following law:
For a given level of throughput, a push system will have more WIP on average than an equivalent CONWIP system [HOP96, p.327].

The law is supported by a calculation for a simple example of a five machine tandem line and exponentially distributed process times with mean one hour.

According to Spearman and Hopp MRP systems have more variable cycle times than equivalent CONWIP systems [HOP96, p.327]. As the total amount of WIP in a line is fixed, the WIP level at the individual stations are negatively correlated. As the WIP level increases at one station, it decreases at all the other stations, which tends to dampen the fluctuations in cycle time. In contrast, WIP levels at the individual stations are independent of one another for MRP. The WIP level at one station reveals no information about the WIP levels at the other stations. The overall WIP level may become extremely high or even low, resulting in great variability of the cycle times that are directly dependent on the WIP level.

Spearman and Hopp state another law to express the robustness of the two systems:

A CONWIP system is more robust to errors in WIP level than MRP is to errors in release rate.

The law is verified with the help of a simple profit function dependent on the throughput and the WIP level expressed in terms of percent error. The coefficients are calculated from empirical data, revealing the functions given in Figure 2-7 [HOP96, p.329].

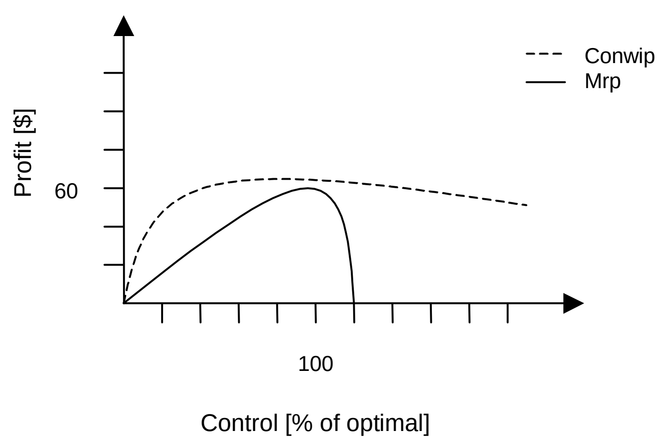


Figure 2-7 Relative robustness of CONWIP and MRP.

The profit function for CONWIP is very flat between WIP levels as low as 40% and as high as 160% of the optimal level. The MRP function declines steadily when the release rate is chosen at a level below the optimum and falls off sharply when the release rate is set even slightly above the optimum level.

## 2.6 Comparison of CONWIP with Kanban

Both CONWIP and Kanban are pull systems since new order releases are triggered by external demand. As both systems control the WIP and limit the level by an upper bound, they show similar performance relative to the push system, MRP.

Gstettner and Kuhn reveal in their comparisons between Kanban and CONWIP that Kanban is more flexible with respect to a certain objective than CONWIP. Not only does the absolute number of cards matter, but, the card distribution is another parameter that influences performance. Selecting a favorable card distribution showed that in a Kanban system a given production rate is reached with less WIP than in a CONWIP system [GST96]. However, Spearman et al. point out that by allowing WIP to collect in front of the bottleneck, CONWIP can function with lower WIP than Kanban [SPE90].

As there is no blocking in CONWIP lines it can easily be understood that a CONWIP system with *n* cards will have a higher production rate than a Kanban system with *n* cards [SPE92].

According to Spearman and Hopp the most obvious difference is that Kanban requires setting more parameters than does CONWIP [HOP96, p.330]. In a one-card system a card count must be established for every workstation, in a two-card system twice as many. In a CONWIP system the amount of cards is set for the entire line, which needs to be established only once. Coming up with the optimal card count requires a combination of analysis and continual adjustment, making it a great deal easier to find the right configuration for the CONWIP system.

Cards are part number specific in a Kanban system and only line specific in a CONWIP system. Instead of being matched to a specific part at the upstream workstation, the cards are matched against a backlog [see 2.3.1], which gives the sequence of parts to be introduced into the line. Thus, in its pure form, a Kanban system must include standard containers of WIP for every active part number in the line to which the cards can be matched. For a high number of parts, although only occasionally produced, this implies a very high overall WIP level swamping the manufacturing system [HOP96, p.330]. Gstettner and Kuhn elaborate on this difference as well, neglecting special release mechanisms in the CONWIP system which are based on a MPS [GST96]. In a paper Spearman et al. mention that although the backlog affords the opportunity for control, it also provides a tremendous challenge. The backlog sequence is the key to assuring adequate capacity when there are significant setups and to optimizing synchronization of production of part components [SPE90].

Hall points out, that Kanban is applicable only in repetitive manufacturing environments [HAL83]. Spearman and Hopp explain repetitive manufacturing by systems where material flows in fixed paths at steady rates [HOP96, p.331]. They mention that large variations in either volume or product mix destroy this flow, at least when parts are viewed individually, and hence seriously undermine Kanban. In another publication Spearman et al. mention that the JIT environment provided by CONWIP can accommodate a changing product mix as it is suitable for short runs of small lots. Furthermore, they find this environment to be more predictable than its pendant provided by Kanban [SPE89]. A CONWIP system is more robust due to the planning capability introduced by the process of generating a work backlog.

Spearman and Hopp mention prevalent employee issues differentiating CONWIP and Kanban. The pull mechanism at every workstation results in great operator stress as described by Klein [KLN89]. When the operator receives a card having to wait for the material to start processing, the void has to be replenished as quickly as possible upon arrival of this material. This is only true for the first workstation in a CONWIP system. The other station function according to a push system where the operators are subjected to less pacing stress [HOP96, pp.332-333].

The previous comparisons illustrate the advantages of CONWIP over MRP and Kanban. Most fundamentally, the differences between the pull and the push systems can be utilized as an advantage to building a manufacturing system that encompasses the positive attributes of the different mechanisms. The result is an integration of the systems to compensate for the weaknesses on both sides. According to Titone integration of various functions into a total comprehensive manufacturing strategy leads to world-class manufacturing and profits. Using MRP II for planning and JIT for the execution combines two powerful tools into an efficient manufacturing system [TIT94]. Wang et al. introduce an experimental push/pull production planning and control software system which is designed as an alternative to a MRP II system for mass manufacturing enterprises in China [WAN96].

Bonvik et al. compare a two-boundary hybrid system to conventional systems. The system is a hybrid of basestock and Kanban control. Basestock control limits the amount of inventory between each production stage and the demand process. Each machine tries to maintain a certain amount of material in its output buffer, subtracting backlogged finished goods demand, if any [KIB88]. For the hybrid system demand information is propagated directly as in basestock control and inventory at the individual workstations is limited as in Kanban control. The hybrid control policy demonstrated superior performance in achieving a high service level target with minimal inventories [BON97].

The three control mechanisms were evaluated by means of simulation as the analytical methods available serve as approximations limited to special cases not applicable to more complex systems.

# CHAPTER 3
# SIMULATION

Simulation refers to a broad collection of methods and applications to mimic the behavior of real systems, usually on a computer with appropriate software. Since computers and software have evolved tremendously in recent time, simulation has become very powerful and popular [KEL98, p.3]. Simulation, like most analysis methods, involves systems and their models. A system is a facility or process, either actual or planned. It is a collection of elements that cooperate to accomplish some stated objectives. A model is a collection of symbols and ideas that approximately represent the functional relationship of the elements in a system [BAI98, p.2]. The system is studied to measure its performance, improve its operation or to determine an optimal design. As sometimes the primary goal is to focus attention on understanding how a system works, the results after the modeling process may become irrelevant. Often, simulation analysts find that the process of defining how a system works, which must be done before developing a model, provides great insight into the mechanisms of the system.

From a practical viewpoint, simulation is the process of designing and creating a computerized model of a real or proposed system for the purpose of conducting numerical experiments to improve the understanding of the behavior of that system for a given set of conditions [KEL98, p.7].

Here, the purpose of the simulation was to evaluate the behavior of the system under different sets of conditions by using the models to carry out groups of experiments. The simulations primarily provided estimates of the statistics of system performance. The systems, Kanban, CONWIP, and MRP, were modeled by a ten identical machine tandem line and exponential distributed process time with mean 20 seconds. Indeed, the modeling process gave great insight into the mechanisms of the systems creating a feeling for their behavior.

Yavuz and Satir reviewed selected published research on Kanban-based operational planning and control in assembly and flow lines. Their article focuses on simulation models and distinguishes between explorative and comparative type research. Operational and experimental design features are summarized in tabular format giving a good overview of work done in this area [YAV95].

## 3.1 The Software

Two simulation tools were used to conduct the experiments: EFML and Arena.

### 3.1.1 EFML

The Emulated Flexible Manufacturing Laboratory (EFML) was developed in the Department of Industrial & Systems Engineering at the University of Florida. The originating concept was to develop a hands-on environment where students and companies could test and study manufacturing operations in a factory setting, giving students and managers the ability to test the performance of a manufacturing facility, which could be distributed over several computers.

The EFML is composed of a network of personal computers linked together through the Virtual Manufacturing Software, which enables the communication of the computers via the TCP/IP protocol and the internet. The software is written with Borland's Delphi Developers Toolkit based on an object oriented architecture. The objects machine, dispatch/raw material inventory storage, repair and maintenance facility, transportation, assembly line, and finished goods inventory storage can be assigned to different computers to construct a complete factory. The object architecture is illustrated in

Figure **3-1**.

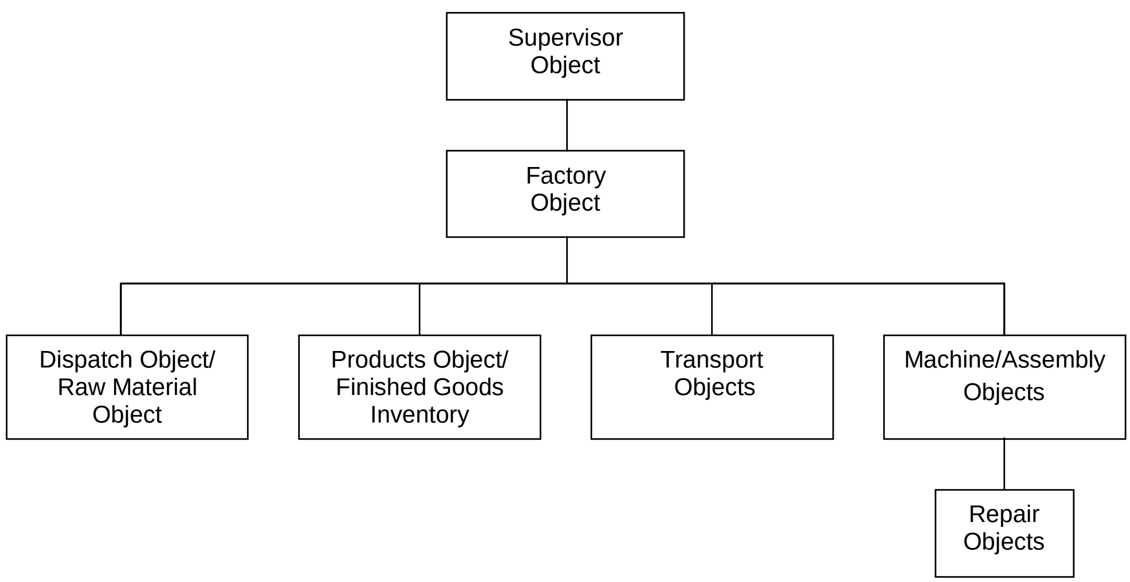


Figure 3-1: The object architecture for the EFML.

As the dispatch object releases material to the shop floor, based on predetermined release times, the behavior of each factory component can be observed in real time. According to Mijon the advantage of the EFML over traditional simulation software is the visual interface providing meaningful output. This output lets the viewer see where the problem is arising and potentially the reason for its occurrence [MIJ97, p. 3].

The EFML is an evolving system which is continuously improved, adding more features to increase the realism of the system and to enhance user friendliness even at the time of writing this thesis.

### 3.1.2 Arena

Arena combines the ease of use found in high-level simulators with the flexibility of simulation languages down to general-purpose procedural languages like the Microsoft Visual Basic programming system, FORTRAN, or C. It does this by providing alternative and interchangeable templates of graphical simulation modeling-and-analysis models that one can combine to build a fairly wide variety of simulation models. For ease of display and organization, modules are typically grouped into panels to compose a template. By switching templates one can gain access to a whole different set of simulation modeling constructs and capabilities. In many cases, modules from different panels and templates can be mixed together in the same model. The modules in Arena templates are composed of SIMAN components. Arena maintains its modeling flexibility by being fully hierarchical, as depicted in Figure 3-2.

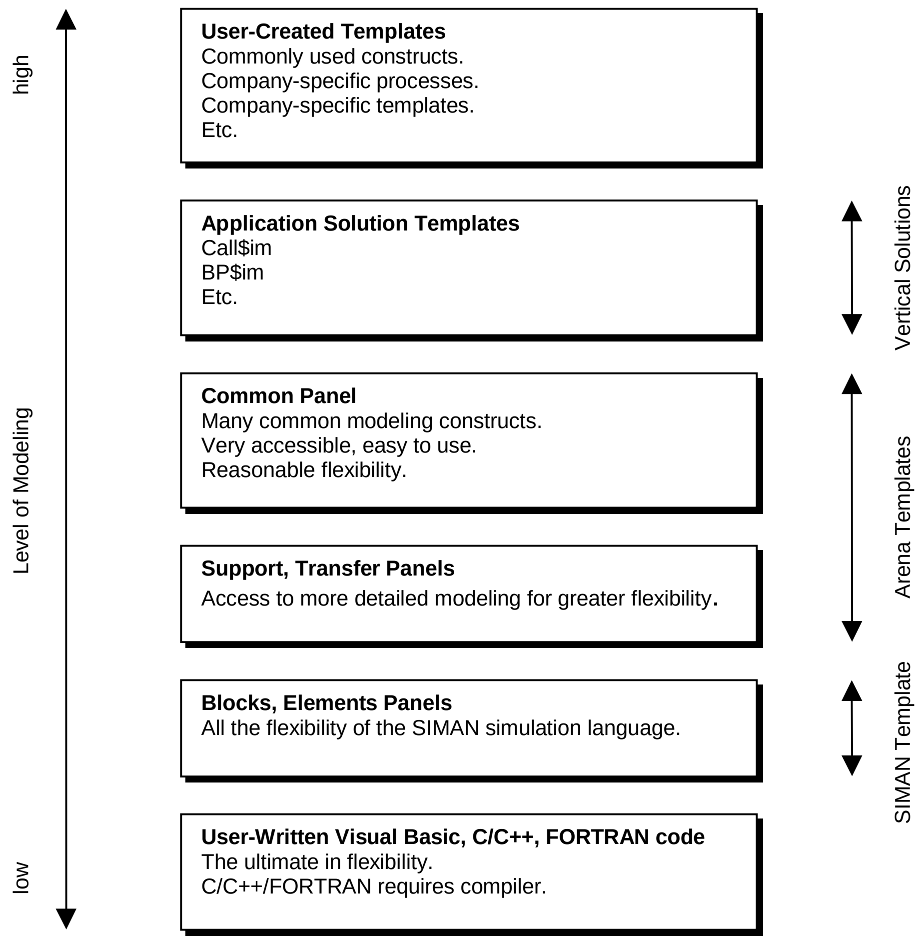


Figure 3-2: Arena's hierarchical Structure.

Arena includes dynamic animation in the same work environment. It also provides integrated support, including graphics, for some of the statistical design and analysis issues that are part of a good simulation study [KEL98, p.13].

The models for Kanban, CONWIP, and MRP were created with the Blocks and Elements Panels to utilize all the flexibility of the SIMAN simulation language.

EFML and Arena served as the framework for the simulation study, which is introduced next.

### 3.2 The Simulation Study

Issues related to design and analysis and representing the model in the software certainly are essential to a successful simulation study. However, there are more aspects that should be taken into consideration. Following the flowchart in Figure 3-3 should improve the chances of conducting a successful study.

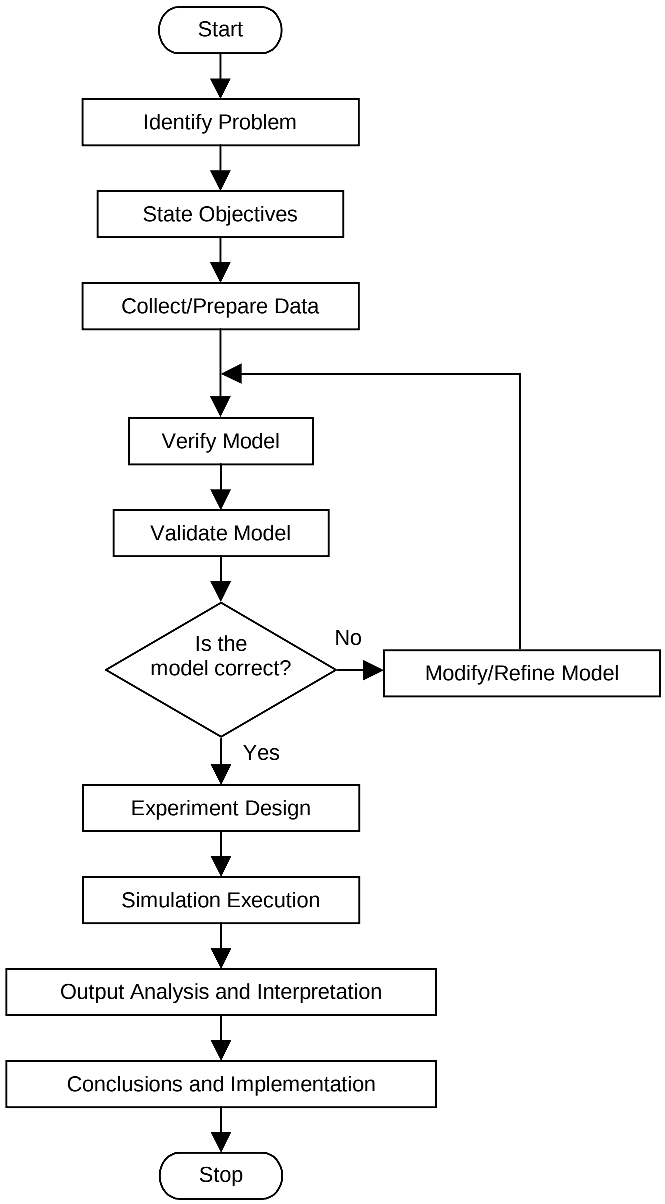


Figure 3-3: Flowchart of a simulation study.

The simulation study does not necessarily have to exactly follow the given flowchart, there is no general formula to guarantee success. It rather gives a rough path to follow. Here, the identification of a problem can be omitted directly proceeding to the second step, stating the objective.

### 3.2.1 State Objective

The objective is to compare the performance of the three manufacturing control systems: Kanban, CONWIP, and MRP. The comparison should involve three main parameters influencing the performance of a manufacturing system:

- Batch size,
- Setup time, and
- Machine failure.

To observe the influence of the individual parameters without any blurring interaction between one another, the central parameter of this study, batch size, is introduced first. The complexity of the models is increased steadily by adding setup time and failure in two further steps. This process allows to build new investigations on the knowledge gained during prior steps improving the realism with the increasing number of parameters.

After determining the objective of this study, the focus had to be directed on the input data.

### 3.2.2 Collect/Prepare Data

The data is produced by the random number generator provided by the software packages. The Arena random number generator was tested by applying the chi-square test of uniformity to the numbers generated. The null hypothesis of uniformity was not rejected at level $\alpha = 0.10$ revealing that the numbers generated didn't behave in a way significantly different from the expectations for truly independent and identically distributed random variables [BAI98, p.60]. Similar behavior was expected from EFML. As previously mentioned, the exponential distribution function was chosen as the input distribution function. This distribution function is commonly used for simulations on manufacturing systems as it has the remarkable memoryless property, where the past history of a random variable, that is distributed exponentially, plays no role in predicting its future [KLE75, p. 66]. Unlike most other probability distributions, the shape of the exponential distribution is governed by a single quantity. Further, it is a distribution with the property that its mean equals its standard deviation [MCC94, p.250].

### 3.2.3 Formulate Models

The models of the systems were built according to the descriptions previously given. Figures 1-3, 1-4, and 1-6 depict the graphical models of Kanban, CONWIP, and MRP respectively. For each control system 4 models were created to enable simulations on the 4 levels including the following parameters:

- Batch size,
- Batch size and setup time,
- Batch size, setup time, and failure (dynamic response), and
- Batch size, setup time, and failure (in steady state).

A few assumptions were made to simplify the simulation process, unfortunately resulting in a less realistic system. The most important assumptions were the following:

- The 10 stages are in series, i.e., each stage has only one supplier and one consumer,
- There is an infinite supply of raw parts at the input of the production system,
- The systems are saturated, there are always demands for finished parts,
- Information is transmitted instantly,
- Transportation within and between workstations is instantaneous,
- The system produces a single part type,
- Kanbans are associated with batches and not with individual entities, and
- Any kanban detached at the output of a stage is immediately available for the upstream stage, there is no return delay.

More assumptions may result implicitly from those given above.

### 3.2.4 Verification of the Models

The three basic models were verified with the EFML output. EFML was verified formally. However, the output data has not been verified before with another simulation software, therefore making this verification process an especially interesting task.

For both Kanban and CONWIP 25 replications were run on Arena and EFML. For MRP 30 replications were carried out. The configurations are given in Table 3-1. The interarrival time corresponds to batch interarrival times.

Table 3-1: Configuration for Kanban, CONWIP, and MRP to verify correctness of the models.

| Control System | Process Time | Batch Size | Number of Cards | Interarrival Time |
|---|---|---|---|---|
| Kanban | 20 | 4 | 20 | - |
| CONWIP | 20 | 4 | 20 | - |
| MRP | 20 | 5 | - | 105 |

A paired t-test [see 4.2.3] was performed on the output data to test the following hypothesis:

$H_0$: True mean of average cycle time differences is equal to 0, and

$H_a$: True mean of average cycle time differences is not equal to 0,

to calculate the 95% confidence interval. The statistics are given in Table 3-2.

Table 3-2: Statistics on t-test to verify concurrence of output between EFML and Arena for Kanban, CONWIP, and MRP.

| System | t-value | df | p-value | Interval | Estimate of mean of diff. | Average Cycle Time |
|---|---|---|---|---|---|---|
| Kanban | 0.4048 | 24 | 0.6892 | (-3.6486; 5.4292) | 0.8903 | 1608.517 |
| CONWIP | 0.2335 | 24 | 0.8174 | (-2.2046; 2.7671) | 0.2812 | 1848.7355 |
| MRP | -0.164 | 29 | 0.8709 | (-115.2965; 98.1806) | -8.5580 | 4166.046 |

All of the intervals include the value 0 resulting in the failure of rejecting the null hypothesis. The 95% confidence intervals indicate a small deviation of the average cycle times for Kanban and CONWIP.

For MRP the interval calculated is considerably bigger, even evaluated relative to the average cycle time. Here CONWIP presents a very small deviation. The reason for the strong deviation of MRP is the varying average cycle time, even after a big amount of entities have passed through the system. The half-width for the confidence interval indicated [see Table 8-4], that 10,000 entities would result in an accurate estimation of the cycle time. Although Figure 3-4 reveals that the average cycle time for 10,000 entities produced has approached a fairly stable value, it is still varying for bigger numbers.

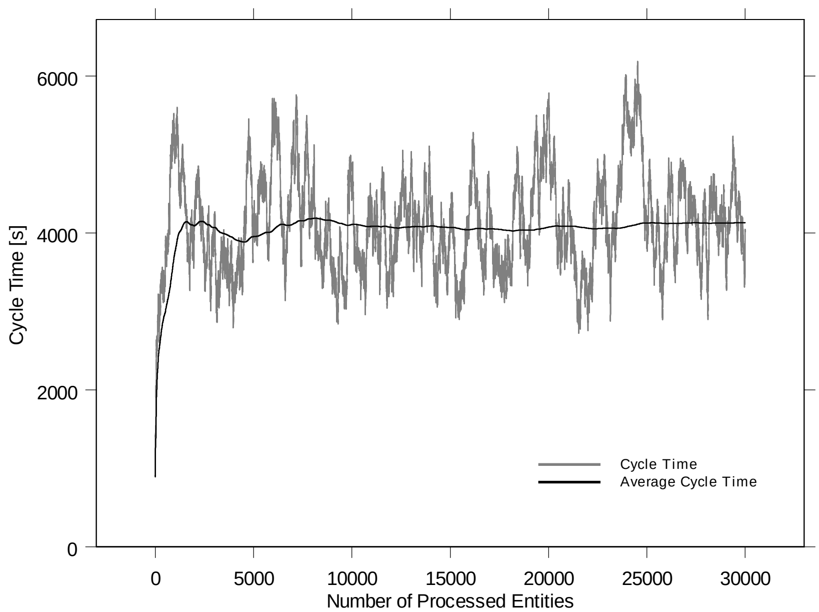


Figure 3-4: The cycle time per entity and the cumulative average cycle time dependent on the number of processed entities for MRP with Arena.

Even after 20,000 entities processed the average is moving, indicating that the random generator has an influence on the output for Arena. The same behavior is expected for EFML, as both simulation tools don't generate true random numbers. This fact could explain great deviations even for a high number of replications completed [see Figure 3-6].

To get an impression of how the systems would behave for different configurations, more simulations were run for varying batch size (1, 2, 4, 8, and 10) and number of cards (10, 15, 18, 10, and 22) or length of interarrival time (22 – 645).

The difference was measured as the percentage deviation in average cycle time, $\Delta\bar{t}_{cycle}$:

$$\Delta\bar{t}_{cycle} = 100\left(\frac{\bar{t}_{cycle}^{EFML} - \bar{t}_{cycle}^{Arena}}{\bar{t}_{cycle}^{EFML}}\right),$$

where $\bar{t}_{cycle}^{EFML}$ is the average cycle time for EFML and $\bar{t}_{cycle}^{Arena}$ is the average cycle time for Arena.

As not enough replications were run to evaluate the output data statistically, scatter diagrams were constructed to visualize the results.

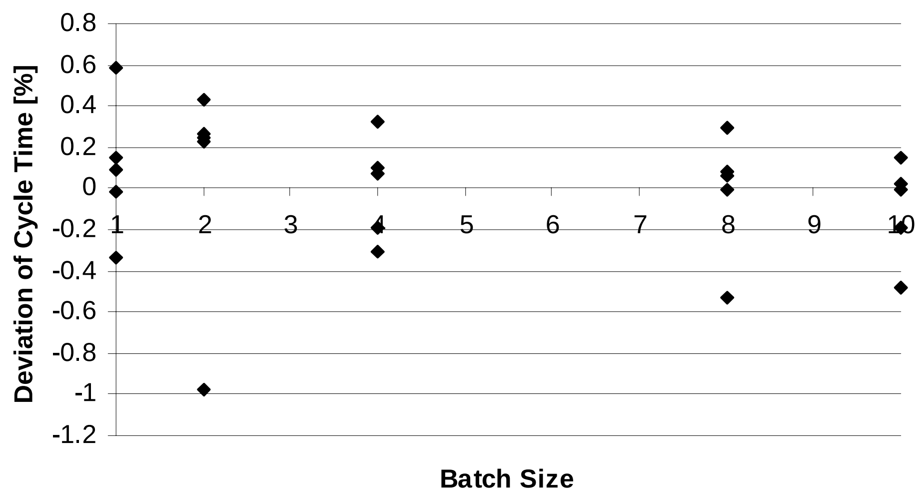


Figure 3-5: The deviation of the average cycle time between EFML and Arena for different configurations for CONWIP.

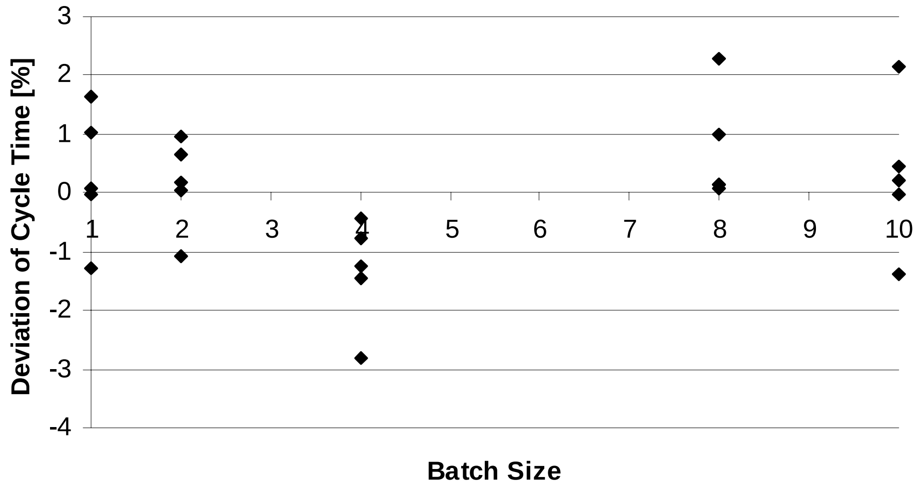


Figure 3-6: The deviation of the average cycle time between EFML and Arena for different configurations for Kanban.

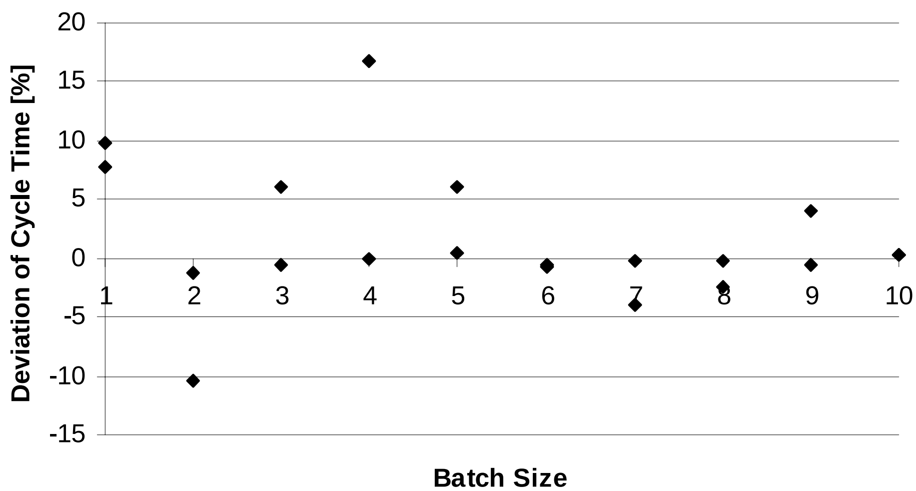


Figure 3-7: The deviation of the average cycle time between EFML and Arena for different configurations for MRP.

Figures 2-5, 2-6, and 2-7 indicate a fairly random output. The shift of data points for Kanban [see Figure 3-6] can probably not be associated to a software error, as the points lie above (batch size eight) and below (batch size four) the x-axis. The calculations done earlier reveal no significant difference between the outputs for batch size four and 20 cards assigned. Faulty input data would probably result in a bigger difference than 2%. The shift may again be attributed to the random generators.

### 3.2.5 Validation

The models under consideration were representing systems existing in theory only. Too many parameters were omitted to enable the simulation of a real system, making validation impossible. Yavuz and Satir mention, that the modeling of real-life manufacturing environment and usage of empirical data would provide a practical means of validation for the simulation models developed. The validation was missing in most of the articles reviewed. Validation would unravel intricacies of manufacturing that are demystified through mostly gross assumptions [YAV95].

### 3.2.6 Simulation Experiment Design

Experiments are performed by investigators in virtually all fields of inquiry, usually to discover something about a particular process or system. Literally, an experiment is a test. A designed experiment is a test or series of tests in which purposeful changes are made to the input variables of a process or system to observe and identify the reasons for changes in the output response [MON91, p.1].

The progression of choice of factors and levels included in the experiments is discussed at the beginning of the chapters covering the different stages of simulation:

- batch size,

- batch size and setup time,
- and batch size, setup time and failure.

The discussions comprise the following factors, henceforth called system parameters:

- Total number of cards assigned to the entire line, *c* [see 5.1.3],
- Batch size, *b* [see 5.1.2],
- The ratio of setup time to process time, $r_s$ [see 6.1.1],
- Time between failures (interfailure time), $t_{intfail}$ [see 7.1], and
- Repair duration, $t_{repair}$ [see 7.1].

The levels were determined according to practical applicability, primarily concerning average machine utilization. First, a high and a low level per factor was established. Then, the interval [low, high] was divided into segments with a certain amount of intermediate levels. As the average run time of one replication was approximately two minutes, the amount of levels was held high, mostly equal to ten.

The total cost could be selected as the primary response variable as the cost affects the basic goal of a company: making profit. The optimization of manufacturing resources can reduce costs considerably resulting in a higher profit margin or even a higher revenue as other market segments are conquered, in turn increasing the overall market share. This elevates cost to one of the most important indicators if not the most important indicator for the efficiency of a manufacturing system.

One characteristic makes cost even more useful. It can serve as an overall indicator, that takes different aspects into consideration, consolidating all the indicators. However, when several indicators are accumulated to be represented by one quantifier, the question, how to weigh the individual components, arises. The weights are most diverging for different industries. Even within one industry, they may differ considerably, representing the company's unique environment.

A wide variety of functions is available enabling a controller to construct a model perfectly fitting the needs. Unfortunately, often weights contain error terms and other parameters that are determined by subjective estimation, making a cost analysis at this point questionable.

Constructing functions for different scenarios would certainly give more insight into the problem [AFY98]. But, the gain in investigating other factors was classified as more important. Furthermore, the regression models can be transformed into cost functions without greater effort. The construction of more complex models would certainly be an interesting topic for another thesis that would probably be most rewarding when written in cooperation with industry.

Consequently, the performance measures were selected as the response variables. The control systems were evaluated on several criteria utilizing the following performance indicators:

- Work in process, *WIP*,
- Throughput, *Th*,
- Average utilization, $\bar{u}$ [see 5.2.1],
- Average cycle time, $\bar{t}_{cycle}$,
- Time spent in the system (analysis of dynamic response), $t_{system}$ [see 7.2.1 Indicators, Time Spent in System], and
- Recovery time (analysis of dynamic response), $t_{recover}$ [see 7.2.1 Indicators, Recovery Time].

The relationship,

$$Th = \frac{WIP}{\bar{t}_{cycle}},$$

is known as Little’s Law and is often referred to in manufacturing literature, being originally derived for a basic queuing system. It was found to be independent upon any specific assumptions regarding the arrival distribution, the service time distribution, the number of servers in the system or upon the particular queuing discipline within the system [KLE75, p. 17]. The formula existed as a “folk theorem” for many years before Little established its validity in a formal way [LIT61]. The formula is a useful tool as it can be used to calculate the third unknown indicator when two indicators are known, independent of system configurations.

The three basic principles of experimental design,

1. randomization,
2. blocking, and
3. replication

were taken into consideration in the following manner:

1. As the system variables and statistics were reinitiated after every replication and the random number generators were assumed to produce numbers confidently, behaving like numbers following a true random distribution, the order of the runs was not randomized.
2. The simulation software and the computer hardware provided an identical environment for every experiment performed, making experiment blocking [MON91, p. 9], unnecessary.

3. As most of the simulations were performed for non-terminating systems [see 3.2.7] a large number of entities was produced rather than completing several replications of the same configuration [see 4.2.2]. Only the analysis done on the dynamic behavior to failure involved a terminating system [see 4.2.1]. Here, the number of replications was established prior to the bulk of experiments [see 8.4.1].

### 3.2.7 Simulation Execution

Depending on the starting and stopping conditions, terminating or non-terminating simulations can be executed as a natural reflection of how the target system actually operates. The terminating simulation ends according to some model-specific rule or condition. For instance, a manufacturing line operates as long as it takes to produce 500 completed assemblies specified by order. According to Kelton et al. the key notion is that the time frame of the simulation has a well-defined and natural end, as well as a clearly defined way to start up. A steady-state of non-terminating simulation, on the other hand, is one in which the quantities to be estimated are defined in the long run, i.e., over a theoretically infinite time frame [KEL98, p. 177]. For a manufacturing line that never stops or restarts, a non-terminating simulation is appropriate.

After initial reflections on parameter settings and several model modifications, preliminary calculations of the confidence intervals [see 4.2 and CHAPTER 8] were conducted. These computations were done to ensure high accuracy on the estimation of the performance indicators. After the completion of the simulations on each of the levels, the confidence on the indicators was reevaluated. All the calculations carried out on the confidence were aggregated and documented in a separate chapter not to disrupt the analysis of the data.

### 3.2.8 Output Analysis and Interpretation of the Results

The output analysis and interpretation forms the major part of this documentation. Since simulation was the modeling tool in question, statistical output analyses were considered in a comprehensive manner. Yavuz and Satir, and Chu and Shih found these issues to be treated rather lightly in many studies reviewed [YAV95] [CHU92].

### 3.2.9 Conclusions and Implementation

At the end of the three chapters encompassing the discussions on the stepwise introduction of batch size, setup time, and machine failure the conclusions drawn from prior investigations are presented. Conclusions presented within the chapters, are clearly marked by a heading.

Unfortunately, a few additional factors have to be taken into consideration to enable simulations of an authentic manufacturing line. However, some findings may be translated into implementations able to improve productivity and efficiency of a real production system.

Before proceeding to the actual discussions of the simulations, a fairly comprehensive but short theoretical background on the statistical analysis methods used is given in the next chapter. The summary of the statistical theory in one chapter can serve as a review for some readers, but should primarily serve as the source of reference making explanations within the chapters redundant. Thus, several clarifications are reduced to one only, and the obstruction of narration is eliminated.

# CHAPTER 4
# STATISTICS

A simulation is a computer-based statistical sampling experiment [BAI98, p.97]. The results of a simulation have to be analyzed with the appropriate statistical techniques to reveal their full potential. Statistics cannot prove that a factor has a particular effect. They only provide guidelines as to the reliability and validity of results. Properly applied, statistical methods do not allow anything to be proved experimentally, but, they do allow us to measure the likely error in a conclusion or to attach a level of confidence to a statement. Thus, the primary advantage of statistical methods is that they add objectivity to the decision-making process. Unfortunately, the output processes of virtually all simulations are non-stationary and auto-correlated. Thus, classical statistical techniques based on identical independent distributed (IID) observations may not be directly applicable. Sometimes, special techniques have to be applied to ensure the statistical independence of the output data.

Let $x_{11}, x_{12}, ..., x_{1m}$ be a realization of the random variables $X_1, X_2, ..., X_m$ resulting from a simulation run of *m* replications using the random numbers $u_{11}, u_{12}, ...$. If the simulation is run with different sets of random numbers $u_{21}, u_{22}, ...$, a different realization $x_{21}, x_{22}, ..., x_{2m}$ of the random variables $X_1, X_2, ..., X_m$ will be obtained. For different runs of a simulation, different random numbers are used for each replication.

The statistical counters are reset at the beginning of each replication, which uses the same initial conditions. Suppose that we make *n* independent runs of length *m*, resulting in the observations:

$$\begin{matrix} x_{11} & \cdots & x_{1i} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2i} & \cdots & x_{2m} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots & x_{ni} & \cdots & x_{nm} \end{matrix}.$$

The observations from a particular replication (row) are not IID due to the nature of the random generators. However, the observations in the *i*th column are IID observations of the random variable $X_i$, *i*=1, 2, …, *m*. This independence across runs allows the statistical methods discussed below to be used. The goal it to make use of the observations to draw inferences about the random variables $X_1, X_2, ..., X_m$, the parameters influencing the performance of the different control systems [BAI98, p.98].

## 4.1 Transient and Steady-State Behavior

For the output stochastic process $X_1, X_2, ...$ let

$F_i(x|I) = P(X_i \leq x|I)$, *i*=1, 2, …,

where *x* is a real number an I represents the initial conditions. $F_i(x|I)$ is called the transient distribution of the output process at time *i* for initial conditions *I*.

For fixed *x* and *I*, the probabilities $F_1(x|I), F_2(x|I), ...$ are just a sequence of numbers.

If $F_i(x|I) \xrightarrow{i \to \infty} F(x)$ for all *x* and all initial conditions *I*, then *F(x)* is called the steady-state distribution of the output process $X_1, X_2, \ldots$ . Here, if the distributions are approximately the same after *k* steps in time, then steady-state is said to start at time *k*. However, steady-state does not mean the random variables $X_{k+1}, X_{k+2}, \ldots$ will take on the same value in a particular simulation run. It means that they will have approximately the same distribution [BAI98, p.98].

As mentioned earlier, statistics can not prove the correctness of a certain statement. Instead, they allow statements to be made with a certain confidence.

## 4.2 Confidence

The statistical analysis methods differ according to whether simulations are terminating or non-terminating [see 3.2.7].

### 4.2.1 Analysis for Terminating Simulations

The data set is given by *n* independent replications of a terminating simulation. Each replication is initiated with the same conditions and a different random generator seed and terminated by a certain event. Thus, independence of the observations is achieved by a different string of random numbers.

Let $X_i$ be the observation of the *i*th replication, *i*=1, 2, …, *n*. It is assumed that the $X_i$`s are comparable for different replications. Consequently, the $X_i$`s can be defined as identical independently distributed random variables.

For n data points $X_1, X_2, \ldots, X_n,$ the sample mean is an unbiased point estimator for the mean of X represented by the following formula:

$$\bar{X}(n) = \frac{\sum_{i=1}^{n} \bar{X}_i}{n}.$$

The 100(1-α)% confidence interval for the mean is given by

$$\bar{X}(n) = \pm t_{n-1,1-\alpha/2}\sqrt{\frac{s^2(n)}{n}},$$

where $s^2(n)$ is the sample variance given by

$$s^2(n) = \frac{\sum_{i=1}^{n}\left[X_i - \bar{X}(n)\right]^2}{n-1}$$

with n-1 degrees of freedom.

Let h be the half-width of the confidence interval of the point estimate,

$$h = t_{n-1,1-\alpha/2}\sqrt{\frac{s^2(n)}{n}}.$$

To ensure the desired accuracy of the estimation,

$$h \le \gamma\bar{X}(n),$$

where $\gamma$ is a given parameter, $0 < \gamma < 1$, here $\gamma = 0.1$ by default.

After an initial simulation with *n* replications this condition may not be satisfied. Additional $n_2$ replications have to be run to reduce the initial half-width $h_1$ to the desired half-width $h_2$ [BAI98, p.103].

For moderately large $n_1$, the sample statistics will remain relatively unchanged with respect to *n*, thus,

$$t_{n_1-1,1-\alpha/2} \approx t_{n_2-1,1-\alpha/2},$$

$$s^2(n_1) \approx s^2(n_2),$$

$\bar{X}(n_1) \approx \bar{X}(n_2)$.

Consequently,

$$n_2 \approx n_1 \left( \frac{h_1}{\gamma \bar{X}(n_1)} \right)^2 .$$

### 4.2.2 Analysis for Non-Terminating Simulations

Let $Y_1$ , $Y_2$, … be an output string from a single replication of a non-terminating simulation. $P(Y_i \le y) = F_i(y) \xrightarrow{i \to \infty} P(Y \le y) = F(y)$,

where $Y$ is the steady state random variable with distribution $F$. Due to the initial conditions, the observations near the beginning of the simulation usually are not representative of the steady-state behavior. For given observations $Y_1$ , $Y_2$, …, $Y_m$ the following formula gives a good point estimate of $E(Y)$:

$$\bar{Y}(m,l) = \frac{\sum_{i=l+1}^{m} Y_i}{m-l},$$

where $l$ stands for the warm-up period and $m$ for the number of observations. $l$ and $m$ are determined such that

$$\bar{Y}(m,l) \approx E(Y).$$

The Method of Batch Means is applied to ensure the accurate calculation of a point estimate for non-terminating systems.

A replication results in observations $Y_1$ , $Y_2$, …, $Y_m$ after removing the warm-up period $l$. The $m$ observations are divided into $n$ batches of length $k$, thus, $m=nk$. Let $\bar{Y}_j(k)$ be the sample mean of the $k$ observations in the $j$th batch. Let

$$\bar{Y}(n,k) = \frac{\sum_{j=1}^{n} \bar{Y}_j(k)}{n} = \frac{\sum_{i=1}^{m} Y_i}{m}$$

be the grand sample mean. Then $\bar{Y}(n,k)$ can be used as the estimate point for *E(Y).*

The batch size *k* can be determined by a correlation analysis. *k* is set equal to the lag length resulting in a minimal correlation of the data. Should

$$n = \frac{m}{k}$$

be non-integer, the excess amount of data, *e*,

$$e = m - \left\lfloor \frac{m}{k} \right\rfloor n$$

can be truncated.

### 4.2.3 Paired-t Confidence Interval

The following assumptions have to be made:

1. Each system provides an equal amount of data (n replications),
2. Observations are independent within the systems.

The following descriptions will refer to the two systems as System A and System B.

Table 4-1: For the paired-t test, comparing two systems is reduced to estimating a single parameter, the difference.

| Replication | System A | System B | Difference |
|---|---|---|---|
| 1 | $x_{a1}$ | $x_{b1}$ | $d_1$ |
| 2 | $x_{a2}$ | $x_{b2}$ | $d_2$ |
| … | … | … | … |
| *n* | $x_{an}$ | $x_{bn}$ | $d_n$ |

The confidence interval on the quantity $\delta$, which is the expected value of $d_{i,}$ will enable a comparison between the two systems. Thus, the problem of comparing two systems is reduced to estimating a single parameter, namely $d_i$ [see Table 4-1]. The resulting confidence interval is referred to as a *paired-t confidence interval*.

This method is particularly appealing as the following assumptions can be omitted:

1. Variance of $x_a$ = variance of $x_b$ (assumption for the two-sample-t method),
2. $x_{ai}$ and $x_{bi}$ are independent.

The confidence interval requires $x_{a1}$ and $x_{a2}$ to be independent, but correlations across rows are permitted. The procedure for computing the confidence interval on $\delta$ is exactly the same as for the single-system case:

$$\bar{d} = \sum_i \frac{d_i}{n},$$

$$s(d) = \sqrt{\sum_i \frac{\left(d_i - \bar{d}\right)^2}{n-1}} \text{ , and}$$

$$s(\bar{d}) = \frac{s(d)}{\sqrt{n}}.$$

The half-width for a (1-α) confidence interval on $\delta$ centered at $\bar{d}$ is then given by $h = t_{n-1,1-\alpha/2} s(\bar{d})$.

The statistic $\bar{d}$ is an estimate of the difference in the measured performance of the two systems: if the two systems perform identically, the expected value of $\bar{d}$ is 0. If the computed confidence interval contains 0, a difference between System A and System B can not be reliably stated. However, if the interval does not contain a 0, a difference between the two systems can be stated with the appropriate confidence level. If the confidence interval does not contain 0, the two systems differ and the appropriate system can be selected based on the sign of $\bar{d}$.

The authors elaborate on the fact, that if the interval on the difference between the systems contains 0, the two systems are not necessarily the same. Additional replications may be required to discern any difference [PEG95, pp. 177].

Another powerful tool to analyze data is regression. As regression describes statistical relations between variables, it also enables estimation and prediction of data points.

### 4.3 Multiple Regression

A regression model is a formal means of expressing the two essential ingredients of a statistical relation:

1. A tendency of the dependent variable to vary with the independent variable in a systematic fashion, and
2. A scattering of points around the curve of statistical relationship [NET90, p. 27].

Probabilistic models that include terms involving $x^2$, $x^3$ (or higher-order terms), or more than one independent variable are called multiple regression models. The general form of these models is

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon .$$

The dependent variable *y* is written as a function of k independent variables $x_1, x_2, ..., x_k$. $x_1, x_2, ..., x_k$ can be functions of variables as long as the functions do not contain unknown parameters. The random error term, $\varepsilon$, is added to make the model probabilistic rather than deterministic. The value of the coefficient $\beta_i$ determines the contribution of the independent variable $x_i$, and $\beta_0$ is the y-intercept. The coefficients $\beta_0, \beta_1, \ldots, \beta_k$ are usually unknown because they represent population parameters

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k}_{\text{Deterministic part of model}} + \underbrace{\varepsilon}_{\text{Random error}} .$$

The Least Squares Approach is used to fit the multiple regression models. The estimated model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + ... + \hat{\beta}_k x_k$$

minimizes

$$SSE = \sum (y - \hat{y})^2 ,$$

where *SSE* stands for the sum of square errors.

The sample estimates $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k$ are obtained as a solution to a set of simultaneous linear equations.

Model Assumptions:

1. For any given set of values of $x_1, x_2, \ldots, x_k$, the random error $\varepsilon$ has a normal probability distribution with mean equal to 0 and variance equal to $\sigma^2$.
2. The random errors are independent in a probabilistic sense [MCC94, p.744].

$\sigma^2$ represents the variance of the random error, $\varepsilon$. Thus it is an important measure of the usefulness of the model for the estimation of the mean and the prediction of actual values of $y$. If $\sigma^2 = 0$, all the random errors will equal 0 and the predicted values, $\hat{y}$, will be identical to $E(y)$, that is, $E(y)$ will be estimated without error. On the other hand a large value of $\sigma^2$ implies large values of $\varepsilon$ and larger deviations between the predicted values, $\hat{y}$, and the mean value, $E(y)$. Thus, $\sigma^2$ plays a major role in making inferences about $\beta_0, \beta_1, \ldots, \beta_k$, in estimating $E(y)$, and in predicting $y$ for specific values of $x_1, x_2, \ldots, x_k$..

Since the variance of the random error will rarely be known, the results of the regression analysis are used to estimate its value with the following formula

$$s^2 = \frac{\sum (y_i - \hat{y}_i)^2}{n-(k+1)},$$

($k$+1) indicating the number of $\beta$ parameters. This will be referred to as the mean square for error (MSE). To enable a meaningful interpretation, the standard deviation s is introduced as a measure of variability

$$s = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n-(k+1)}}.$$

### 4.3.1 Estimating and Testing Hypotheses about the $\beta$ Parameters

Some of the β parameters have practical significance in the models formulated in the following chapters. Thus, their values will be estimated and hypotheses will be tested about them. Considering the model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

the following hypothesis could be performed using a t-test:

null hypothesis $H_0 : \beta_2 = 0$ (No curvature in the response curve.)

against the

alternative hypothesis $H_a : \beta_2 < 0$ (Concavity exists in the response curve.).

The t-test utilizes a test statistic analogous to that used to make inferences about the slope of the straight-line regression model. The t statistic is formed by dividing the sample estimate, $\hat{\beta}_2$, of the parameter, $\beta_2$, by the estimated standard deviation of the sampling distribution of $\hat{\beta}_2$, $s_{\hat{\beta}_2}$:

Test statistic: $t = \dfrac{\hat{\beta}_2}{s_{\hat{\beta}_2}}$.

For relevant estimated model coefficients $\hat{\beta}_i$ the estimated standard deviation $s_{\hat{\beta}_i}$ and the calculated $t$ values will be given. To find the rejection region for the test the upper-tail value for t is retrieved from the *t*-table. This is a $t_\alpha$ such that $P(-t < -t_\alpha) = \alpha$. This value can then be used to construct rejection regions for either one-tailed [see Figure 4-1] or two-tailed tests.

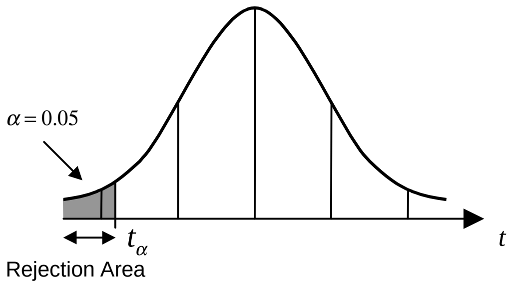


Figure 4-1: Rejection region for a test of $\beta_2$

The numbers given in the following chapters list the two-tailed significance levels for each $t$ value. The null hypothesis, that the parameter equals to zero, would be rejected in favor of the alternative hypothesis, that the parameter does not equal to zero, at any α level larger than the given number. A 100(1-$\alpha$)% confidence interval for a β parameter is given by

$$\hat{\beta}_i \pm t_{\alpha/2} s_{\hat{\beta}_i}$$

where $t_{\alpha/2}$ is based on $n$-($k$+1) degrees of freedom and n observations and ($k$+1) $\beta$ parameters in the model [MCC94, p.746].

### 4.3.2 Usefulness of a Model: $R^2$ and the Analysis of Variance *F*-Test

Conducting $t$-tests on each $\beta$ parameter in a model is not a good way to determine whether a model is contributing information for the prediction of $y$. When conducting a series of $t$-tests to determine whether the independent variables are contributing to the predictive relationship, it is most likely that an error would be made in deciding which terms to retain in the model and which to exclude. This may result in including a large number of insignificant variables and excluding some useful ones. Thus, a global test that encompasses all the $\beta$ parameters is needed. Furthermore, it would be useful to find a statistical quantity that measures how well the model fits the given data. As this statistical quantity $R^2$ , the multiple coefficient of determination, can be used to calculate the F value. $R^2$ will be introduced first.

### 4.3.3 Multiple Coefficient of Determination, $R^2$

As the name multiple coefficient of determination indicates, $R^2$ is the equivalent of $r^2$, the coefficient of determination for the straight-line model [see MCC94, p. 697]. It is defined as the following

$$R^2 = 1 - \frac{\sum (y-\hat{y})^2}{\sum (y-\bar{y})^2} = \frac{\text{Explained variability}}{\text{Total variablity}},$$

where $\hat{y}$ is the predicted value of $y$ for the model. $R^2$ represents the fraction of the sample variation of the $y$ values that is explained by the least squares prediction equation. $R^2 = 0$ implies a complete lack of fit of the model to the data and $R^2 = 1$ implies a perfect fit with the model passing through every data point. Thus, the larger the value of $R^2$, the better the model fits the data [MCC94, p. 759].

### 4.3.4 Variance *F*-Test

The following test would formally test the global usefulness of the model:

$H_o : \beta_1 = \beta_2 = \ldots = \beta_k = 0$

(All model terms are unimportant for predicting $y$.),

$H_a$ : At least one of the coefficients $\beta_i$ is nonzero

(At least one model term is useful for predicting *y*).

The test statistic used to test this hypothesis is an *F* statistic, which can be calculated with the following formula:

$$F = \frac{R^2 / k}{(1 - R^2)/[n - (k+1)]},$$

where *n* is the sample size and *k* is the number of terms in the model. The formula indicates that the *F* statistic is the ratio of the explained variability divided by the model degrees of freedom to the unexplained variability divided by the error degrees of freedom. The larger the proportion of the total variability accounted for by the model, the larger the *F* statistic.

To determine when the ratio becomes large enough that the null hypothesis can be rejected and the model is more useful than no model at all for predicting *y*, the calculated F value is compared to a tabled *F* value:

Rejection region: $F > F_{\alpha}$, where *F* is based on k numerator and $n-(k+1)$ denominator degrees of freedom.

McClave et al. caution the reader that a rejection of the null hypothesis leads to the conclusion, with 100(1-$\alpha$)% confidence, that the model is useful. However, useful does not necessarily mean best. Another model may prove even more useful in terms of providing more reliable estimates and predictions. Thus, this global F-test is usually regarded as a test that the model must pass to merit further consideration [MCC94, p.762]. It will only be used in this sense in the following chapters.

### 4.3.5 Comparison of two or more Regression Functions

Instead of fitting separate regressions for separate data sets, only one regression is fitted. This regression gives rise to the same response functions otherwise obtained. This has the following advantages:

1. Inferences can be made more precisely by working with one regression model containing indicator variables since more degrees of freedom will then be associated with the mean standard error (*MSE*)[NET90, p.355],
2. One regression run on the computer will yield both fitted regressions, and
3. Tests for comparing the regression functions for the different classes of the qualitative variable can be clearly seen to be tests of regression coefficients in a general linear model [NET90, p.358].

Here the data sets of the different control systems are accumulated to produce one data set. Indicator variables (or binary variables) that take on the values 0 and 1 are used to quantitatively identify the classes of the qualitative variables distinguishing the control systems. To prevent computational difficulties a qualitative variable with c classes will be represented by (*c*-1) indicator variables [see NET90, p.351].

Assuming that a first order model is to be employed it would give rise to the following function:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 i_1 + \varepsilon$$

where

$x_1$ = independent variable, and

$$i_1 = \begin{cases} 1 & controlsystem1 \\ 0 & controlsystem2 \end{cases}.$$

The response function of this regression model is:

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 i_1,$$

which can be interpreted as:

$$E(y) = (\beta_0 + \beta_2) + \beta_1 x_1$$

for the control system 1, and as:

$$E(y) = \beta_0 + \beta_1 x_1$$

for the control system 2. Thus, $\beta_2$ measures the differential effect of the type of control system. It shows how much higher (lower) the mean response line is for the class coded 1 than the line for the class coded 0, for any given level of $x_1$.

This approach is completely general. If three control systems are to be compared, additional variables are simply added to the model. Furthermore, the differentiation is not only limited to the y-intercept, but can be introduced to distinguish gradients or coefficients of variables with higher order.

However, the following assumption has to be made:

The error term variances in the regression models for the different populations are equal, otherwise transformations may be used to approximately equalize them.

### 4.3.6 Transformation

Simple transformations of either the dependent variable *y* or the independent variable *x*, or of both, are often sufficient to make the simple regression model appropriate for the transformed data. Unequal error variances and non-normality of the error terms frequently appear together. To reduce the departure from a simple linear regression model a transformation on *y* is needed, since the shapes and spreads of the distributions of *y* need to be changed. Such a transformation on *y* may help to linearize a curvilinear regression relation at the same time. At other times, a simultaneous transformation on x may also be needed to obtain or maintain a linear regression relation. However, it is very unlikely that such a transformation will be needed in the following chapters.

Box and Cox [COX58] have developed a procedure for choosing a transformation from the family of power transformations on *y*. This procedure is useful for correcting unequal error variances. The family of power transformations is of the form:

$y' = y^{\gamma}$,

where $\gamma$ is a parameter to be determined from the data. The family encompasses the following and widely used transformation:

$y' = \log_e y$.

The criterion for determining the appropriate parameter $\gamma$ of the transformation of *y* in the Box-Cox approach is to find the value of $\gamma$ that minimizes the error sum of squares *SSE* for a liner regression based on that transformation.

**4.3.7 Residual Analysis**

When regression analysis is applied deviations from the initial assumptions may result in incorrect reliabilities stated. The departures have to be detected and taken into account should they be big enough to alter the results. Fortunately, experience has shown that least squares regression analysis produces reliable statistical tests, confidence intervals, and prediction intervals as long as the departures from the assumptions are not too great [MCC94, p.784].

As the assumptions [see 3.2.3] concern the random error component, $\varepsilon$, of the model, a first step is to estimate the random error. Since the actual random error associated with a particular value of *y* is the difference between the actual *y* value and its unknown mean, the error is estimated by the difference between the actual *y* value and the estimated mean. This estimated error is called the regression residual, denoted by $\hat{\varepsilon}$.

$\varepsilon$ = actual random error

= (actual y value) – (mean of *y*)

$= y - E(y)$

$= y - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k)$

$\hat{\varepsilon}$ = estimated random error (residual)

= (actual y value) – (estimated mean of *y*)

$= y - \hat{y}$

$= y - (\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_k x_k).$

As the true mean of *y* (i.e., the true regression model) is not known, the actual random error can not be calculated. However, because the residual is based on the estimated mean (the least squares regression model), it can be calculated and used to estimate the random error and to check the regression assumptions. These checks are generally referred to as residual analyses [MCC94, p.784].

### 4.3.8 Influential Observations

When using regression, some subset of the observations may be found to be unusually influential. Sometimes these influential observations are relatively far away from the vicinity of the rest of the data. Dennis R. Cook developed an excellent diagnostic, the Cook's distance. This is a measure of the squared distance between the usual least squares estimate of $\beta$ based on all $n$ observations and the estimate obtained when the *i*th point is removed, say, $\hat{\beta}_i$ [NET90, p. 403].

The next chapter comprises a discussion of the influence of the batch size on the performance of the three manufacturing systems. A comparison between the systems introduces the chapter to give the reader a brief overview of the material. Then, the two pull systems are discussed in more detail to explain their behavior. The push system, MRP, is introduced separately due to its different attributes. After dealing with the material in more detail on a level where the interdependence of factors is more evident the discussion continues on a higher level by returning to the comparison of the systems.

# CHAPTER 5
# BATCH SIZE

Avoiding setups and facilitating material handling are the two primary reasons for batching jobs together in a manufacturing system. If large lots of similar products are run in batches, equipment setups are infrequently needed. If setups are long, large lots result in substantially more effective capacity. Furthermore, for process batches equal to move batches the material that is moved between workstations in large batches requires less handling than if it is moved in small lots [HOP96, p. 288].

The entities arrive at a workstation in a batch. While the first entity of that batch enters the machine, the remaining entities have to wait to be processed. The batch can be transported to the next stage in the system, only when the last entity of a batch is completed. Here, transportation is assumed infinitely fast resulting in zero transportation time.

A variety of single stage models and analytical techniques have been reviewed by Chaudhry and Templeton [CHA83]. The literature covers single stage manufacturing systems only, not applicable to a ten machine tandem line. Gold investigates sophisticated batch service systems in push and pull manufacturing environments as single stage systems by using embedded Markov chain techniques [GOL92]. Kim et al. focus on production scheduling in semiconductor wafer fabrication taking batch sizes into account. They use simulation to evaluate new scheduling rules [KIM98].

Schoening and Kahnt show how to extend the methodology of Mitra and Mitrani [MIT90] to model a one-card Kanban system with batch servers [SCG95]. However, in all three cases the batches could be processed simultaneously by batch servers, such as plating baths, drying facilities, and heat-treating ovens, not quite transferable to the tandem line with sequentially processing machines.

The model parameters and their levels are introduced to elaborate on the input data prior to the discussion of the simulation results.

## 5.1 Parameters

The process time was established at 20 seconds throughout all the simulations while the following parameters were varied to evaluate the performance of the manufacturing systems:

- Batch size,
- Total number of cards assigned to the line, and
- The interarrival time for MRP.

The levels of these parameters or factors are discussed briefly.

### 5.1.1 Process Time

A workstation which processes a batch size $r$ can be modeled as an $r$-stage Erlangian server. In such a system a customer enters the server, proceeds one stage at a time through the sequence of $r$ stages and departs at the end. Only then, a new customer enters. The total time that a customer spends in this service facility is the sum of $r$ independent identically distributed random variables, each chosen from an exponential distribution. The probability distribution function of the service time is an Erlangian distribution [KLE75, pp. 123-124]. Consequently, the process time for a batch of size $r$ is distributed according to an $r$-stage Erlangian distribution with a mean of the individual process time, viz. 20 seconds. The batch size and the mean process time per entity were given as an input.

### 5.1.2 Batch Size

The following batch sizes were selected:

1, 2, …, 10, and 20.

Initially, neutral experiments were conducted to establish differences of system behavior for batch size 20. The results were found to be compliant with the results obtained for batch size 1 to 10. Thus, batch size 20 was omitted for further experiments.

### 5.1.3 Number of Cards

The second design parameter portrays the number of cards assigned to the entire line. Naturally, this parameter applies to the pull systems only. Its pendant for MRP is the interarrival time. The parameter merely indicates the total amount of cards in the system. It does not specify the number of cards assigned to individual machines. Huang and Wang determine the number of cards in a CONWIP system, $\theta$, by applying Little's Law:

$$\theta = \mu t ,$$

where $\mu$ is the average throughput of the production line and $t$ is the average time for a card to pass through the production line. The formula is expanded to approximate the number of cards in a production line in series containing a bottleneck [HUA98].

Optimizing the number of kanbans in a line has been a popular research topic. According to Bonvik et al. most kanban implementations set the parameters by rules of thumb or simple formulas [BON97]. Sugimori et al. state Toyota's formula as an example:

$$c > \frac{DL(1+\alpha)}{p} ,$$

where $c$ is the number of cards, $D$ is the demand rate, $L$ the replenishment lead time, $\alpha$ a safety factor, and $p$ the number of parts in a container [SUG77]. During factory operation, the kanban numbers are steadily decreased by reducing the safety factor. According to Bonvik et al. the fact that the formula is based on standard lead times is less than satisfying, as it does not reflect the lead time consequences of shop floor congestion and limited machine capacities [BON97].

Liberopoulos and Dallery use an iterative heuristic to optimize the number of cards assigned to a conventional single-stage Kanban control system (KCS). They show that the computational complexity of optimizing a single-stage generalized Kanban control system (GKCS) is the same as that of optimizing the KCS, which can be considered a special case of the GKCS [LIB95]. However, the algorithm was found to be rather complex, making use of an analytically tractable approximation method or simulation for initialization. Dallery and Liberopoulos introduce the extended Kanban control system (EKCS) as a KCS accommodating $N$ stages in another publication [DAL95], which was recently generalized to assembly structures by Chaouiya et al. [CHY98]. However, these discussions have a pure comparative nature, not incorporating the number of cards assigned to the system.

Unlike CONWIP, the Kanban control system does not only vary with the number of cards assigned to the entire system, but, its performance is dependent on the number of cards assigned to the individual machines. To ensure a comparison of an optimal Kanban with CONWIP and MRP, some card allocation studies had to be carried out prior to the actual simulations.

#### 5.1.3.1 Card Allocation for Kanban

Card allocations can not be carried out according to a generally applicable algorithm. Some rules have been documented, applicable to specific manufacturing systems. Gsettner and Kuhn make use of a heuristic to determine the optimal allocation for a given production rate in a Kanban line with *m* stations. The production rate is calculated analytically underestimating the true production rate systematically. The procedure starts with assigning one card to every station. The number of cards is then increased at each station on a trial basis. The distribution which shows the best ratio between change in production rate and WIP is finally accepted (greedy procedure) [GST96].

The next sub-chapter constitutes an endeavor to specify general allocation rules relevant to the ten machine tandem line.

### 5.1.3.2 Card Allocation Rules

To visualize the material and to avoid ambiguity, the rules are explained with the assistance of statics, essential to any engineering education. The ten machine tandem line [see Figure 5-1] can be modeled as a beam supporting ten weights of equal distance to one another [see Figure 5-2].

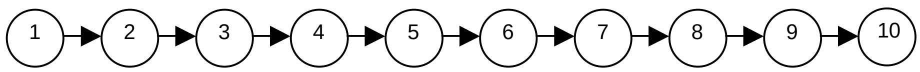


Figure 5-1: The ten machine tandem line.

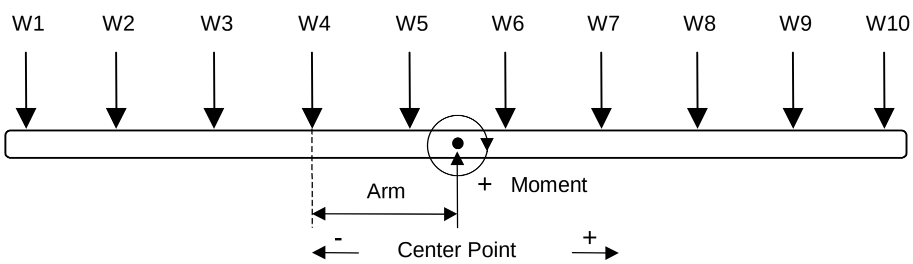


Figure 5-2: Free body diagram of the ten machine tandem line modeled as a beam.

The *moment* of a force is its tendency to produce rotation of the body on which it acts, about some axis. The measure of a moment is the product of the force and the perpendicular distance between the axis of rotation and the line of action of the force. This distance is called the *moment arm* [see Figure 5-2]. The intersection of the axis of rotation with the plane of the force and its moment arm is called the *center of moments* [JEN83, p. 15]. As it is a point, it is referred to as *center point* here. All the forces of a system may be regarded as the component of their resultant force. Hence, about any point, the moment of the resultant force (total weight) equals the algebraic sum of the moments of the separate forces (weights). This principle is known as Varignon's Theorem [JEN83, p. 17].

For the ten machine production line, the weight refers to the number of cards assigned to a machine. The weight increases with increasing number of cards allocated. Thus, the balance of the line can be expressed as the moment of the resultant force, a consequence of a specific card allocation.

As the rules are not applicable to all manufacturing lines, the following assumptions were made:

1. Identical machines,
2. All machines comprise the bottleneck,
3. Objective: maximum throughput, and
4. Center point: median of line (between machine 5 and 6).

Applying the statics analogy to the manufacturing line the following rules result:

1. Increase weight of last machine last,
2. Positive moment preferred to negative moment:
    - Increase weight on positive side of center point first,
    - Start increasing weight with smaller arm first,
3. Establish balance on line:
    - Symmetric structure relative to center point,
        - moment close to the absolute minimum (zero): same weight with certain arm on either side (positive and negative) of center point,

- Small difference (one card) in weight between the machines for the entire line, and

4. Minimize number of consecutive machines with same weight.

All rules are to be applied simultaneously. However, the importance of the rules decreases with increasing number. Thus, if the rules contradict one another, the rules with lower number override the rules with higher number. Initially, all machines get assigned the same amount of cards. Then, any additional cards are allocated according to the rules. All the additional cards previously positioned may have to be reallocated for one more card assigned to the line, thus satisfying an additional rule. For example: the card remaining from allocating one card to each machine, the 11th card, is assigned to the 6th machine [see Figure 5-3].

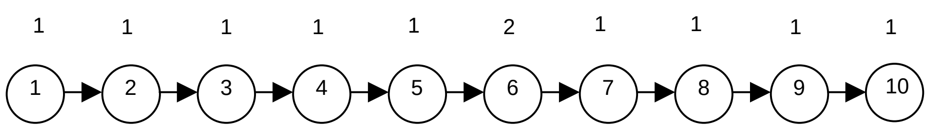


Figure 5-3: Number of cards per machine for 11 cards assigned to a ten machine line.

However, when a 12th card is assigned, the 11th card previously assigned has to be reallocated to machine seven while the 12th card is assigned to machine four [see Figure 5-4].

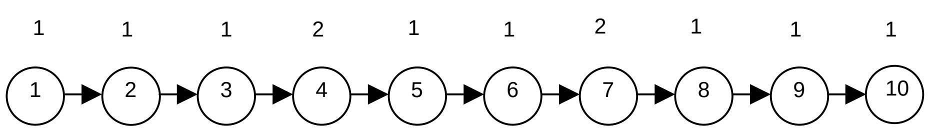


Figure 5-4: Number of cards per machine for 12 cards assigned to a ten machine line.

To test the correctness of the rules, some simulations were carried out. For these simulations batch size, setup time, and failure were not taken into consideration. It was assumed, that the card allocations were optimal independent of the above mentioned parameters.

The rules were found to result in a good approximation of the optimum. However, an approximation was not good enough to compare Kanban with the other two systems, both being able to perform at their optimal settings. Consequently, more simulations were run to establish optimal card allocations for 10 to 70 cards assigned to the line. Assuming that the performance of the line could not be improved otherwise, one rule was kept: small difference (one card) in weight between the machines for the entire line.

The following number of combinations, *m*, had to be run for 10 to 19 cards being assigned:

$$m = \sum_{j=0}^{9} \binom{10}{j} = 1023.$$

As it was found that even the optimal allocations for the interval 10 to 19 cards could not be applied to the lines with 20 to 70 cards, simulations had to be run for the following intervals:

1. [10, 19],
2. [20, 29],
3. [30, 39],
4. [40, 49],
5. [50, 59], and
6. [60, 69],

plus one last replication for 70 cards assigned. This resulted in

$n = 6m+1 = 6(1023)+1 = 6139$

experiments.

Thus, 6139 replications were completed resulting in the data to evaluate the rules quantitatively.

### 5.1.3.3 Deviation of Rules from Optimum

The performance of the line for 10 to 70 cards assigned was measured by the throughput. The percentage increase in throughput, $I$, for allocating optimally, $Th_o$, instead of allocating according to the rules, $Th_r$, was calculated according to the following formula:

$$I = \left( \frac{Th_o - Th_r}{Th_r} \right) 100\% .$$

Figure 5-5 indicates an increase in most of the cases. Only in a few cases the rules resulted in the optimal allocation. Naturally, there was no increase in throughput for 10, 20, …, 70 as with these numbers only one allocation was possible under the given assumptions [see p. 68].

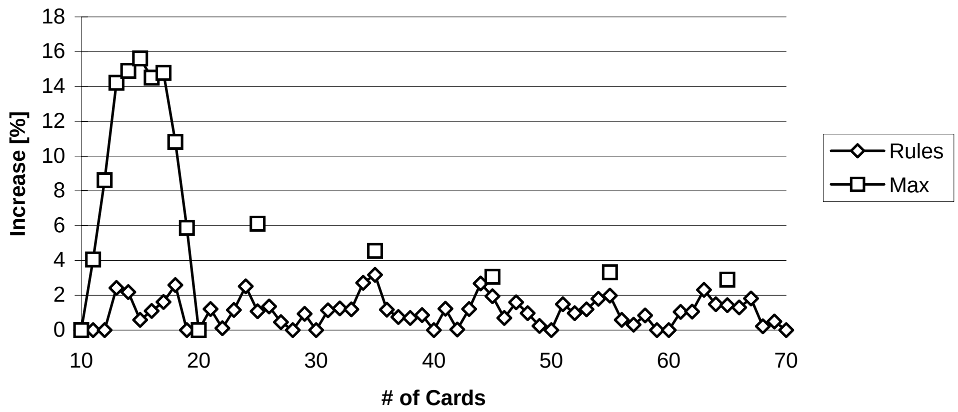


Figure 5-5: Increase in throughput by allocating cards optimally instead of simply applying the rules.

To put these percentage increases in a relative context, the maximal increases, i.e. the increases from the worst possible allocation to the optimal allocation, are indicated in Figure 5-5 (Max) as well. The graph shows all maximal increases for the first interval, 10 to 20 cards assigned, and only the maximal increase for 25, 35, …, 65 cards assigned per consecutive allocation interval. These numbers were expected to show the greatest deviation in throughput as they give rise to the greatest amount of different possible allocations, *a*:

$$a = \binom{10}{5} = 252\,,$$

where five additional cards had to be assigned after an equal amount of cards was allocated to all the machines.

The graph illustrates the good approximation of the optimum by the rules. This is especially true for a bigger number of cards assigned. It can clearly be seen that the maximal increase decreases with an increasing amount of cards in the system. This can be ascribed to the following:

- the machines are busy most of the time as enough cards have been allocated to them,
- the increase of utilization per additional card assigned to the system decreases with an increasing amount of cards allocated [see Figure 5-6], and
- the ratio,

  $$r = \frac{c_1}{c_2},$$

  where $c_1$ is the smallest number of cards assigned to any machine on the line and $c_2$ the largest number of cards assigned to a machine, decreases as the difference, $d^c = c_2 - c_1$, is kept constant and equal to 1.

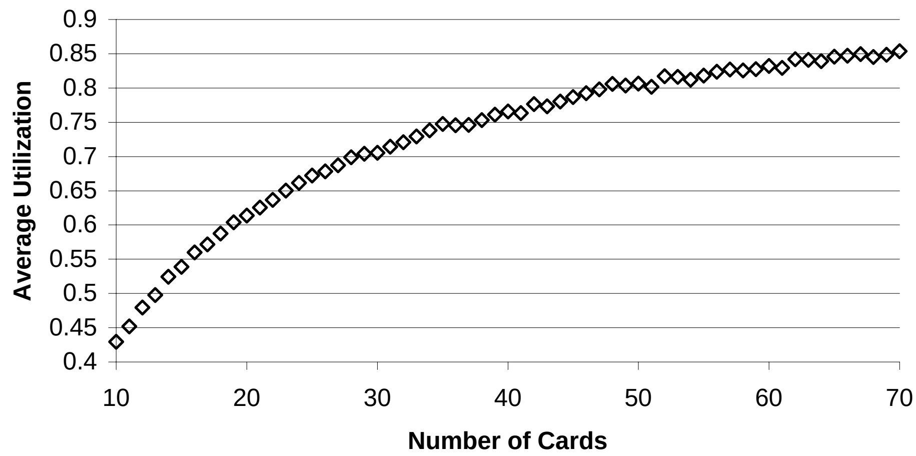


Figure 5-6: The average utilization dependent on the number of cards for Kanban.

The optimal card allocations for maximum throughput, minimal work in process and minimal average cycle time were carefully studied.

The optimal allocations for minimizing WIP and average cycle time were found to be very close to the rules applied. Note that these rules are different from those given above, as the primary objective to achieve minimal WIP and minimal average cycle time is to liberate the system of WIP. This is most efficiently done by placing more cards towards the end of the line to pull material out of the system. Less cards at the beginning of the line would result in raw material only being pulled into the system for processing, not keeping any excess material in the buffers.

However, trying to achieve maximal throughput resulted in a great variability of where the additional cards should be placed. Table 5-1 shows an extraction of the list obtained to illustrate this interesting phenomenon. The systems with the same amount of additional cards were grouped together. These additional cards were indicated as ones in their respective rows. Looking at the table unveils no obvious pattern. $M_{median}$ and $M_{beginning}$ are discussed below.

Table 5-1: Additional cards allocated to the system with ten machines.

| # of cards assigned to the system | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | $M_{median}$ | $M_{beginning}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| 21 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -2 | 4 |
| 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 8 |
| 41 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -5 | 1 |
| 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 10 |
| 61 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5 | 10 |
| 12 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 11 |
| 22 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 12 |
| 32 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 3 | 13 |
| 42 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 11 |
| 52 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | 10 |
| 62 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 12 |
| 13 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | -2 | 15 |
| 23 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | -2 | 15 |
| 33 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 16 |
| 43 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 17 |
| 53 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 2 | 18 |
| 63 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 17 |

As the research on card allocation was not the main topic of this research paper, a very limited amount of time was spent trying to find patterns that could explain this variation. Some calculations were done to express the findings mathematically. The interest was focused on the balance of the system. $M_{median}$ represents the moment of the line with the center point at the median (between machine 5 and machine 6):

$$M_{median} = \sum_{i=1}^{10} l_i w_i \ ,$$

where $w_i$ stands for the weight of machine $i$ [see 5.1.3, Card Allocation Rules] and $l_i$ stands for the arm of machine $i$. This was expected to be close to zero at all times, assuming the correctness of the rules. As can be seen in Table 5-1, this number greatly varies and sometimes equals to the maximum arm, $l_5 = 5$.

$M_{beginning}$ quantifies the moment of the line for additional cards with the center point at the beginning of the line, such that $l_i = i$:

$$M_{beginning} = \sum_{i=1}^{10} i d_i^c \ ,$$

where $d_i^c$ is the difference between the amount of cards of the different machines in the system [see 5.1.3, Card Allocation Rules]. This formula indicates the position of weight on the line. For 33 and 43 [see Table 5-1] $M_{median}$ is the same and indicates a balanced line. However, $M_{beginning}$ shows, that the weight is distributed differently, viz. more towards the end of the line for 43 cards assigned. Comparing $M_{median}$ and $M_{beginning}$ for the different allocations, shows no evident pattern. More research could be conducted to find explanations for this behavior.

### 5.1.4 Interarrival Time

This parameter stands for the time interval between two consecutive batch arrivals. Its inverse is the arrival rate. The interarrival time was favored to the arrival rate as it is understood more intuitively. Furthermore, it served as a direct input value for the software applied.

The selected levels resulted from setting the utilization interval $[\bar{u}_{min}, \bar{u}_{max}]$ for MRP equal to the utilizations for the pull systems. The levels selected divided the intervals into nine partitions.

As the average cycle time represents one of the primary indicators of the performance of a manufacturing line, its response to a change in batch size is discussed first.

### 5.2 Average Cycle Time

The following graph shows the influence of the batch size and the number of cards allocated to the system on the average cycle time for the three control systems: Kanban (1), CONWIP (2), and MRP (3) [see Figure 5-7].

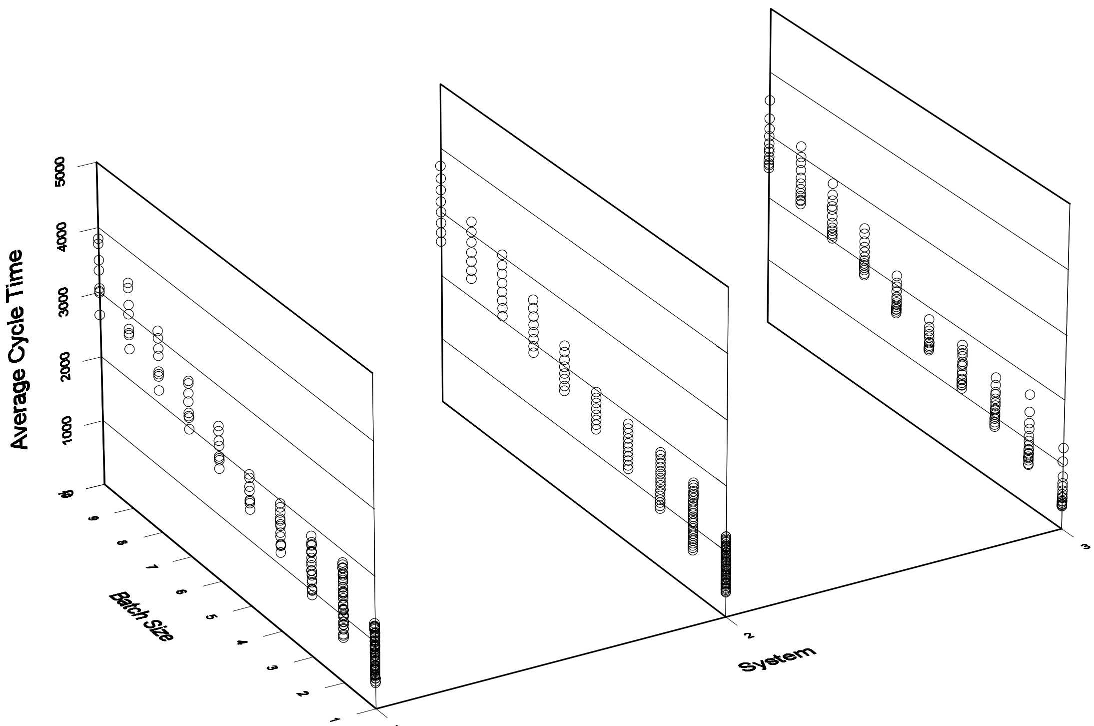


Figure 5-7: The average cycle time dependent on the batch size and number of cards allocated to the line for the three control systems: Kanban (1), CONWIP (2), and MRP (3).

The average cycle time increases with increasing batch size. For the vertically aligned data points, the number of cards assigned increases from bottom to top. As the material is pulled into the system in batches the last member of each batch has to wait until all the other members are processed. As the batch size increases, this waiting time increases.

The lowest values of the average cycle time per batch size were obtained for the least number of cards assigned to the system, viz. ten. Ten cards theoretically enable all the machines to be busy simultaneously. Furthermore, Kanban requires this minimal amount to function. For the upper bound at most 200 entities were chosen:

$$Wip_{max} = bc = (10)(20) = 200,$$

where $b$ is the batch size and $c$ is the number of cards assigned to the system.

However, another constraint enforces even stronger limitations on the systems: the average utilization of the machines.

### 5.2.1 The Average Machine Utilization

Little's Law relates the three parameters: throughput, cycle time, and work in process. This interdependence has proven practically to be the only stable observation for the turbulent stochastic manufacturing systems. Thus, it can easily be used to make conclusions about one of the parameters when one is kept constant and the other one is known.

The three parameters constitute ideal indicators of performance for a production system. The production engineers are most definitely interested in reducing work in process to decrease cycle time and increase the throughput of the line. Thus, these parameters serve as quantitative indicators enabling state of the art process control.

From these indicators other indicators can be derived. One of these indicators would be the machine utilization. The utilization, *u*, can be determined independently of the throughput, but, they are directly related:

$$u = \frac{Th_{average}}{Th_{theory}},$$

where $Th_{average}$ is the average throughput derived from the systems under study, and $Th_{theory}$ is the theoretical throughput, which can be determined by the following formula:

$$Th_{theory} = \frac{1}{t_{process}},$$

where $t_{process}$ is the process time of the bottle neck machine in minutes. Here, the machines are identical and can all be considered bottle neck machines with a process time of 20 seconds or $\frac{1}{3}$ minute resulting in the following:

$$Th_{theory} = \frac{1}{\frac{1}{3}} = 3$$

entities per minute.

The utilization gives a relative performance of a machine and can be calculated for the entire line. The average utilization, $\overline{u}$, of the line can be calculated by the following formula:

$$\overline{u} = \frac{\sum_{i=1}^{10} u_i}{10},$$

where $u_i$ is the utilization of machine *i*. For the simulations completed in this study, the utilization of a machine was determined by sampling the system in 20 second intervals. The sampling interval was set arbitrarily equal to the machine process time.

The average throughput $Th_{average}$ was computed at the end of each simulation replication using Little's Law:

$$Th_{average} = \frac{WIP_{average}}{\bar{t}_{cycle}},$$

where $\bar{t}_{cycle}$ is the average cycle time and

$$WIP_{average} = \frac{\sum_{i=1}^{10} WIP_i}{10}$$

is the average work in process calculated from the work in process per machine *i* sampled in 20 second intervals.

The average cycle time was computed by the following formula:

$$\bar{t}_{cycle} = \frac{\sum_{j=1}^{n} t_j^{cycle}}{n},$$

where $t_j^{cycle}$ is the cycle time of entity *j*, *j*=1, …, *n*, here *n*=10,000. Calculating the average utilization either way results in the same value. For the simulations completed the values differ in the third decimal after the comma, an insignificant difference.

The utilization serves as an ideal indicator for the practical applicability of the simulation data. A rule of thumb states that utilizations above 0.9 are unrealistic in industry. Utilizations below 0.65 are uneconomical. To make the observations and conclusions transferable to real life problems, all configurations were chosen to produce utilization within the interval [0.65, 0.9].

As the utilization was one of the constraints kept throughout all simulations, it unveiled itself as an ideal parameter to serve as a common factor for comparisons between the three control systems.

To enable a comparison between the systems the utilization interval was kept constant for all three of them [see Figure 5-8].

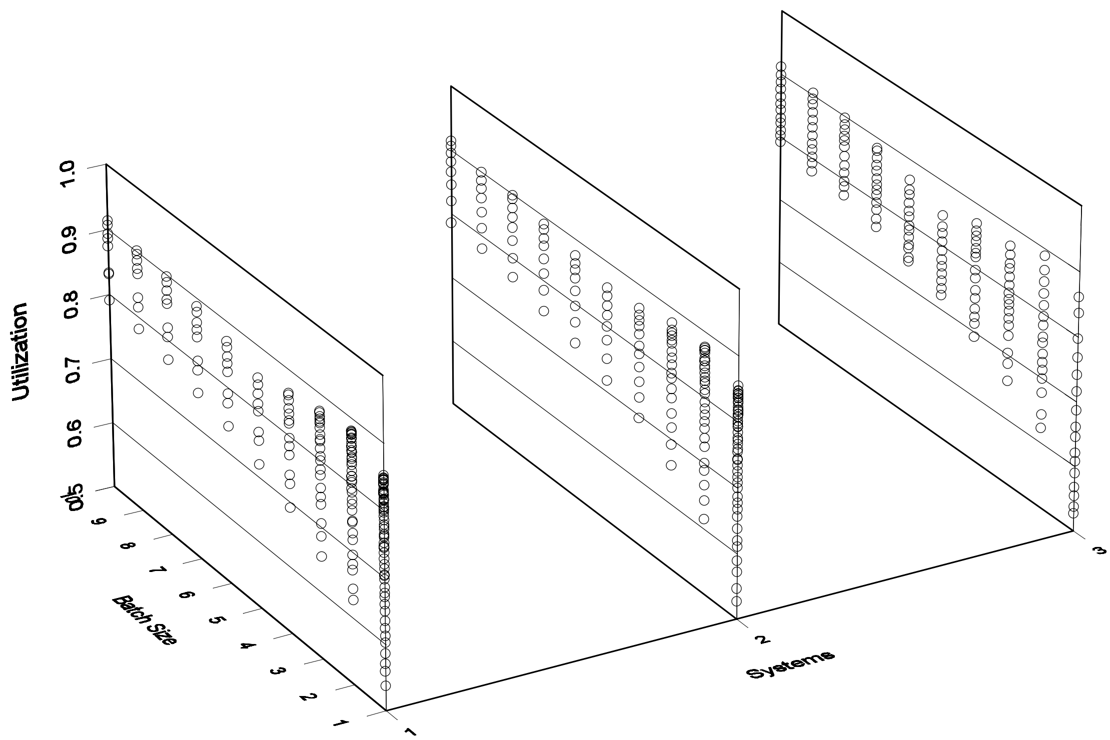


Figure 5-8: Utilization dependent on batch size and number of cards for Kanban (1), CONWIP (2), and MRP (3).

This utilization interval was created by controlling the number of cards assigned to the line for the pull systems and by adjusting the interarrival time for the push system, MRP.

### 5.2.2 Kanban and CONWIP

As the average WIP differs for Kanban and CONWIP for a certain amount of cards [see 2.6], the minimal amount of cards for Kanban could only be determined by additional replications. The data with utilization below the lower bound of the interval [0.65, 0.9] was dropped. In CONWIP all the cards are used at all times resulting in the same amount of WIP as number of cards in the system. In Kanban the amount of cards represents an upper limit of WIP in the system. Here, the cards are bound to a machine. Assuming that a machine at the beginning of the line has an exceptionally long processing time for a particular entity, the machines down the line, at least those close to the momentary bottleneck, are idle. These machines don’t carry WIP, leaving the system with less WIP than cards assigned. Figure 5-9 shows that the data points are clearly below the 45 degrees line.

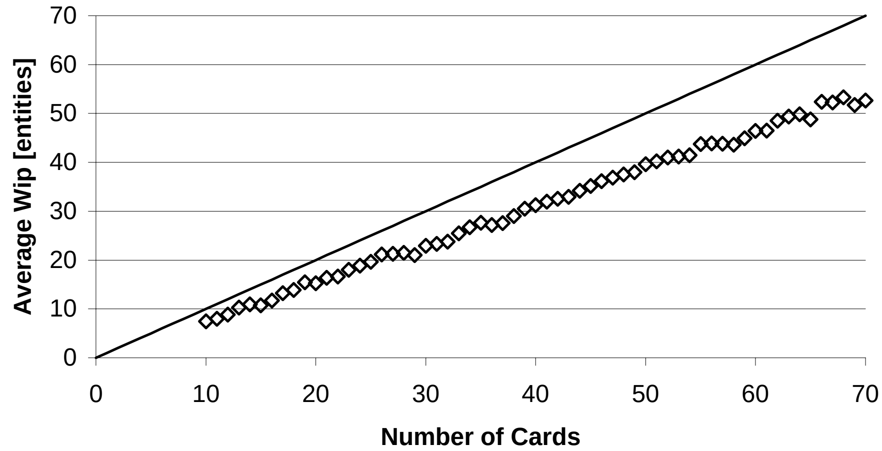


Figure 5-9: Average WIP dependent on the number of cards assigned to a Kanban system.

Figure 5-7 also shows that with an increasing number of cards assigned, average work in process increases and, consequently, the average cycle time increases as well [see 3.2.6]. The data points are aligned vertically, the number of cards assigned increases from bottom to top. The waiting time for a unit to be processed increases not only within the batches, but, it increases throughout the line as more material is caught in the system. Figure 5-10 illustrates this dependence for CONWIP.

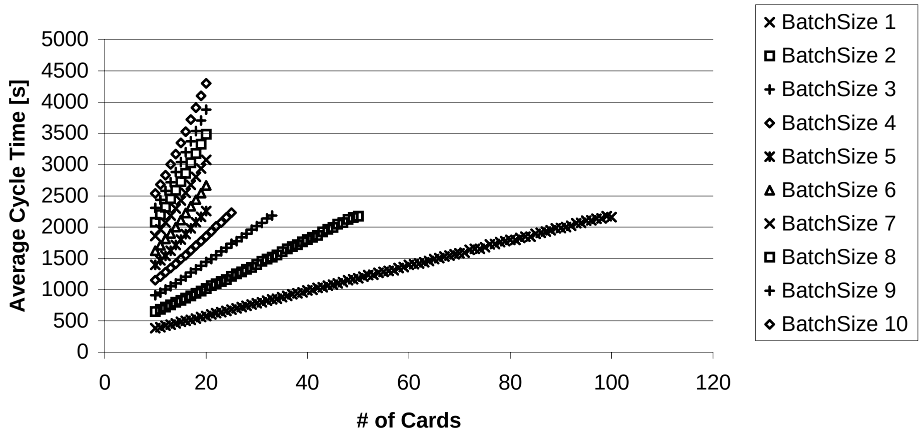


Figure 5-10: The average cycle time dependent on the number of cards assigned to the system for CONWIP.

The straight lines formed by the data points without any outliers make the linear dependency most evident and indicate that Little's Law holds: as the number of cards increases, the WIP increases, and the average cycle time increases. Naturally, Kanban shows the same behavior, although the lines are not quite as smooth [see Figure 5-11]. Figure 5-7 reveals this fact, too. The circles are not as evenly spread for Kanban as for CONWIP for specific batch sizes. Especially the higher batch sizes show a higher concentration of circles at certain average cycle times.

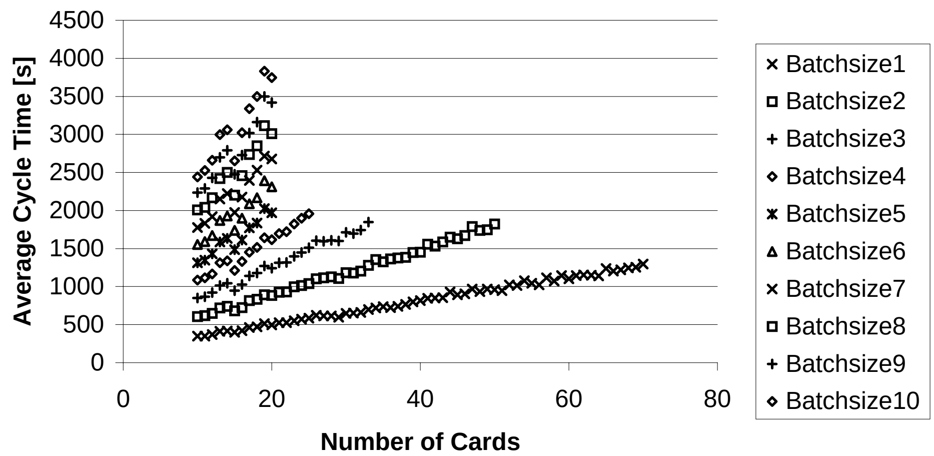


Figure 5-11: The average cycle time dependent on the number of cards assigned to the system for Kanban.

To find out how well the data fits the linear dependency assumption a multiple regression analysis was performed on both systems. As expected, the regression model for CONWIP proves the linear behavior of the dependent variable. The high value of the multiple coefficient of determination, $R^2 = 0.9999$ [see Row 24, Table 5-2], shows the perfect fit of the model. To give the reader an idea of the data output given by the software, it is tabulated below.

Table 5-2: Multiple regression output for CONWIP with the average cycle time (Avgct) dependent on the number of cards (Ccards).

| Formula | | | | | |
|---|---|---|---|---|---|
| Cavgct ~ Ccards + Ci2 + Ci3 + Ci4 + Ci5 + Ci6 + Ci7 + Ci8 + Ci9 + Ci10 + Ccardsci2 + Ccardsci3 + Ccardsci4 + Ccardsci5 + Ccardsci6 + Ccardsci7 + Ccardsci8 + Ccardsci9 + Ccardsci10 | | | | | 1 |
| **Residuals** | | | | | |
| Min | 1Q | Median | 3Q | Max | |
| -19.75 | -5.095 | -0.4204 | 4.093 | 28.57 | 2 |
| **Coefficients** | | | | | |
| Parameter | Value | Std.Error | t-value | Pr(>\|t\|) | |
| (Intercept) | 178.9593 | 3.8587 | 46.3780 | 0.0000 | 3 |
| Ccards | 20.0774 | 0.1136 | 176.7149 | 0.0000 | 4 |
| Ci2 | 73.5170 | 6.6358 | 11.0788 | 0.0000 | 5 |
| Ci3 | 156.7482 | 9.6439 | 16.2536 | 0.0000 | 6 |
| Ci4 | 242.2489 | 13.2812 | 18.2400 | 0.0000 | 7 |
| Ci5 | 375.3362 | 24.0987 | 15.5750 | 0.0000 | 8 |
| Ci6 | 408.7711 | 20.5851 | 19.8576 | 0.0000 | 9 |
| Ci7 | 465.8402 | 18.0067 | 25.8704 | 0.0000 | 10 |
| Ci8 | 519.8010 | 20.5851 | 25.2513 | 0.0000 | 11 |
| Ci9 | 569.4994 | 20.5851 | 27.6656 | 0.0000 | 12 |
| Ci10 | 636.8179 | 20.5851 | 30.9359 | 0.0000 | 13 |
| Ccardsci2 | 18.3685 | 0.2409 | 76.2555 | 0.0000 | 14 |
| Ccardsci3 | 35.1863 | 0.4872 | 72.2192 | 0.0000 | 15 |
| Ccardsci4 | 51.1600 | 0.8083 | 63.2931 | 0.0000 | 16 |
| Ccardsci5 | 62.7219 | 1.8121 | 34.6125 | 0.0000 | 17 |
| Ccardsci6 | 81.7514 | 1.4810 | 55.1986 | 0.0000 | 18 |
| Ccardsci7 | 99.1862 | 1.2407 | 79.9445 | 0.0000 | 19 |
| Ccardsci8 | 115.7481 | 1.4810 | 78.1533 | 0.0000 | 20 |
| Ccardsci9 | 133.1146 | 1.4810 | 89.8792 | 0.0000 | 21 |
| Ccardsci10 | 149.4230 | 1.4810 | 100.8906 | 0.0000 | 22 |
| **Residual standard error** | | | | | |
| 9.57 on 130 degrees of freedom | | | | | 23 |
| **Multiple R-Squared** | | | | | |
| 0.9999 | | | | | 24 |
| **F-statistic** | | | | | |
| 50270 on 19 and 130 degrees of freedom, the p-value is 0 | | | | | 25 |

The formula [see row 1] describes the linear dependence of the response variable to the nineteen independent variables that have a main effect on the model. It contains 9 indicator variables, Ci*j*, *j=2, 3, …, 10*, to fit the y-intercepts for the ten curves in the data set. This is one variable less than the number of fitted curves to enable calculation. The Ccardsci*j*, *j=2, 3,…, 10* variables enable the calculation of the gradients, expressing the differentiation of slopes. This results in the following model fitted to the data set:

$Cavgct = \beta_1 Ccards + \beta_2 Ci2 + \beta_3 Ci3 + \beta_4 Ci4 + \beta_5 Ci5 + \beta_6 Ci6 + \beta_7 Ci7 + \beta_8 Ci8 + \beta_9 Ci9 + \beta_{10} Ci10 + \beta_{11} Ccardsci2 + \beta_{12} Ccardsci3 + \beta_{13} Ccardsci4 + \beta_{14} Ccardsci5 + \beta_{15} Ccardsci6 + \beta_{16} Ccardsci7 + \beta_{17} Ccardsci8 + \beta_{18} Ccardsci9 + \beta_{19} Ccardsci10.$

Row 2 shows the distribution of the residuals. The given extremes, min and max, show the very good estimation of the line. The estimated regression line deviates from the given data by maximally 28.57 seconds, where the average cycle time lies in the interval [381.1972, 4298.0820]. The intercept of the line for batch size 1 is given in row 3, while the increase in coefficient $\beta_j$ from $\beta_{j-1}$ for $\beta_2$ , …, $\beta_{19}$ is given in rows 4 to 22. The residual standard error will be discussed shortly [row 23]. The p-value given for the f-statistic indicates, that for any $\alpha>0$, the null hypothesis, that $\beta_j = 0$, j=1, …, 19, is rejected.

As the lines for Kanban are less smooth, the regression was expected to show a greater variance for the residual error. Although the initial model had a high multiple coefficient of determination, $R^2=0.9892$, the residual graph [see Figure 5-12] clearly indicated an unequal residual variance making transformation necessary.

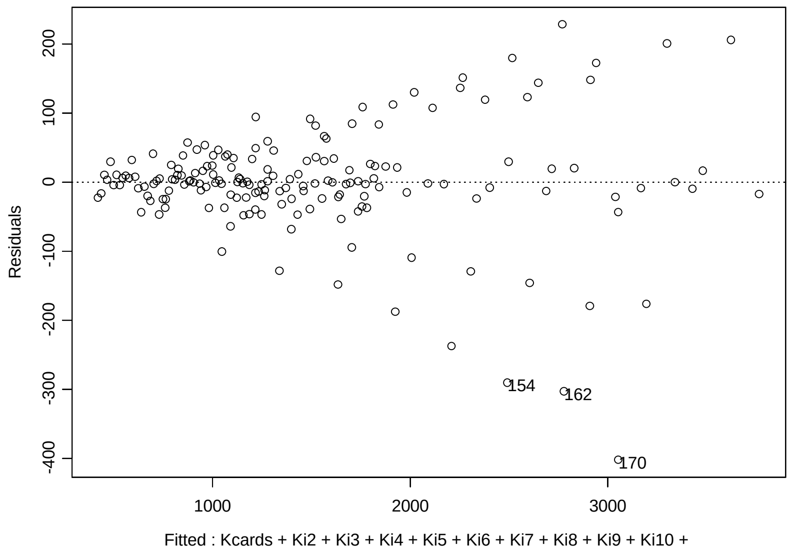


Figure 5-12: Unequal residual error variance for initial model fitted to Kanban.

From the transformations,

$y' = \log_2 y$,

$y' = \log_e y$, and

$y' = \sqrt{y}$

the following was chosen with the help of the Box and Cox procedure [see 4.3.6] and observation of the residual plots:

$y' = \log_e y$.

This transformation had the least residual standard error [see Table 5-3] and resulted in a considerable decrease in inequality [see Figure 5-13]. The distribution of the data points resembles the gun shot pattern.

Table 5-3: The residual standard error of different transformations for the multiple regression on Kanban.

| Indicator | $y$ | $y' = \sqrt{y}$ | $y' = \log_2 y$ | $y' = \log_e y$ |
|---|---|---|---|---|
| Residual Standard Error | 84.36 | 0.9206 | 0.07605 | 0.05272 |

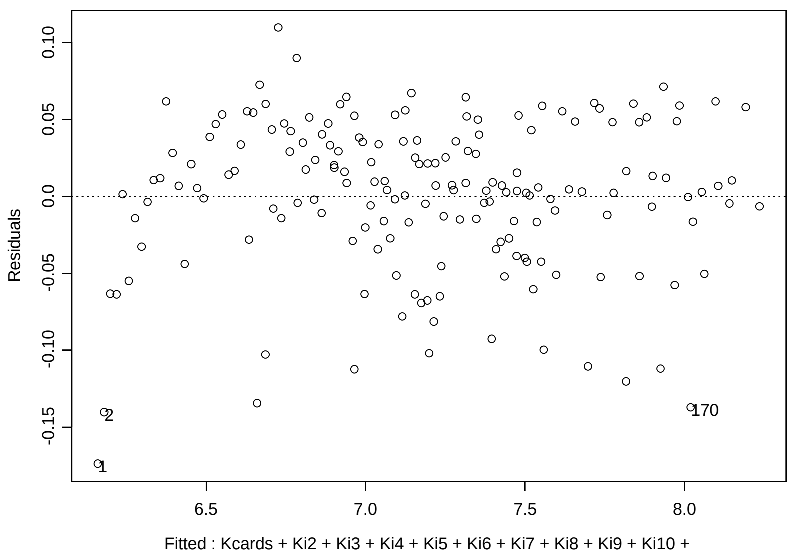


Figure 5-13: The distribution pattern for the residual error of the transformed multiple regression model for Kanban.

The regressions result in the following functions:

$$Avgct_i^{Conwip} = \beta_0 + \beta_1 c$$
$$Avgct_i^{Kanban} = e^{\beta_0 + \beta_1 c},$$

where $\beta_0$ is the y-intercept and $\beta_1$ is the gradient for batch size *i*, *i=1, 2, …, 10.* The values for the coefficients are given in the table below [see Table 5-4].

Table 5-4: Function coefficients describing the dependency of the average cycle time and the batch size derived by multiple linear regression for Kanban and CONWIP.

| CONWIP | | |
|---|---|---|
| *i* | $\beta_0$ | $\beta_1$ |
| 1 | 178.9593 | 20.0774 |
| 2 | 252.4763 | 38.4459 |
| 3 | 409.2245 | 73.6322 |
| 4 | 651.4734 | 124.7922 |
| 5 | 1026.81 | 187.5141 |
| 6 | 1435.581 | 269.2655 |
| 7 | 1901.421 | 368.4517 |
| 8 | 2421.222 | 484.1998 |
| 9 | 2990.721 | 617.3144 |
| 10 | 3627.539 | 766.7374 |
| Kanban | | |
| *i* | $\beta_0$ | $\beta_1$ |
| 1 | 5.8694 | 0.0195 |
| 2 | 6.2798 | 0.0254 |
| 3 | 6.9001 | 0.0377 |
| 4 | 7.662 | 0.0562 |
| 5 | 8.5958 | 0.0763 |
| 6 | 9.6922 | 0.0964 |
| 7 | 10.9192 | 0.117 |
| 8 | 12.24 | 0.1393 |
| 9 | 13.6507 | 0.1629 |
| 10 | 15.1567 | 0.1864 |

However, these functions can be represented by a single function, $Avgct = f(Bsize, Cards)$, per control system. The multiple coefficients of determination indicate a loss of accuracy, but, they are still exceptionally high, indicating a good fit of the model to the given data [see Table 5-5]. The two functions were derived by multiple regression and can be used to calculate the average cycle time dependent on the number of cards used and the batch size chosen.

Table 5-5: The derived functions for CONWIP and Kanban to estimate the average cycle time for given batch size and number of cards assigned to the system.

| System | Model | $R^2$ |
|---|---|---|
| CONWIP | $\bar{t}_{cycle} = 99.5873 + 89.2901b - 2.6088b^2 + 2.8994c + 17.1832bc$ | 0.9996 |
| Kanban | $\bar{t}_{cycle} = e^x$<br>$x = 5.6679 + 0.3204b - 0.0225b^2 + 0.0108c + 0.0073bc$ | 0.9764 |

The results for Kanban are visualized below. The graph gives an impression of how the transformation influences the spread of data points [see Figure 5-14]. The data can be considered a little blurred, reducing the distance between the data points, especially for the larger values of the average cycle time. Furthermore, the good fit of the model to the data can be seen.

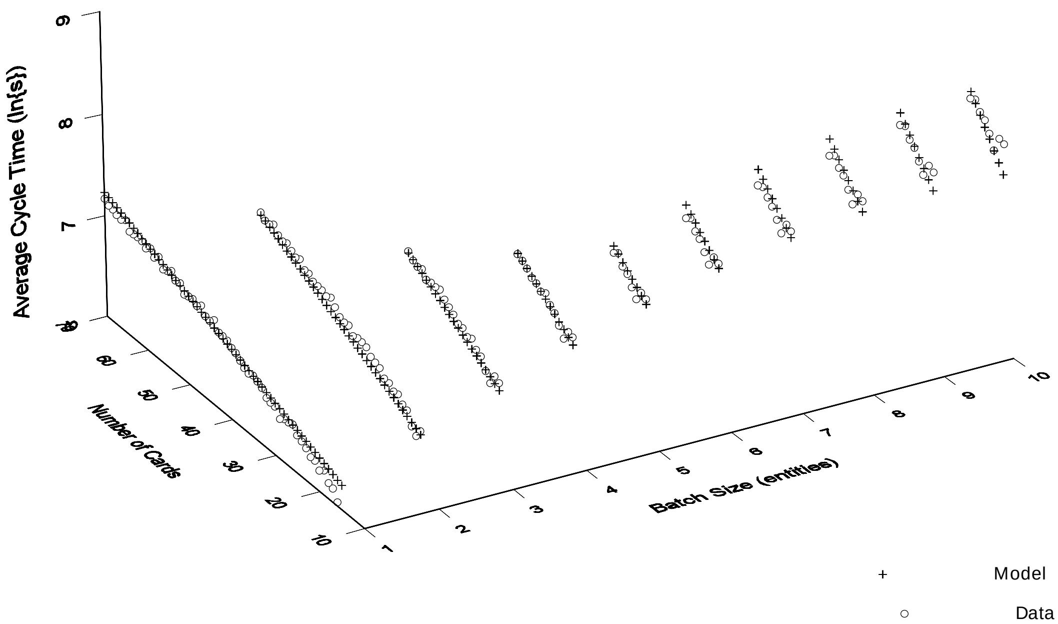


Figure 5-14: Three dimensional illustration of the ln-transformed data points of the average cycle time, dependent on the batch size and number of cards, and the data points computed with the regression model for Kanban.

### 5.2.3 Findings and Conclusions for Kanban and CONWIP

The models indicate the following:

- Linear dependence between the number of cards assigned and the average cycle time,
- Small quadratic influence of the batch size on the average cycle time, and
- Interaction of the batch size and the number of cards assigned.

The following conclusions can be made:

1. The batch size has a higher impact on the average cycle time than the number of cards assigned. This is true for the throughput as well, keeping the work in process constant (Little's Law).
2. For an increasing batch size, the increase in average cycle time increases for every additional card assigned to the system [see Figure 5-14]. To keep the average cycle time at a minimum the smallest batch size should be chosen.
3. The smaller batch size is superior to a larger one in every sense. The linear function of the intercepts [see Table 5-2] proves this. In other words: for a fixed amount of cards assigned the average cycle time increases with increasing batch size. There are no identical data points, the lines do not cross over [see Figure 5-10], which can be derived from conclusion 2.
4. The results shown reveal the dependence of the batch size and number of cards. Optimization should involve both parameters simultaneously. Here, a stepwise approach could lead to an optimal allocation, when one of the parameters is held constant as the gradients of the functions are positive in the given interval.

After showing the dependency of the average cycle time on the batch size and number of cards assigned for Kanban and CONWIP, the discussion is continued for MRP in the following chapter.

### 5.2.4 MRP

The average WIP can not be controlled as easily by the push systems as by the pull systems. As the batch size and the number of cards assigned increases, the WIP increases by the same amount (CONWIP) or proportionally (Kanban). As the WIP increases, the average cycle time increases, provided the throughput remains constant. With MRP the WIP can only be controlled indirectly. An increase in WIP is achieved by increasing the amount of material introduced to the system. This is controlled by increasing the release rate or decreasing its inverse, the interarrival time. Figure 5-15 shows the response of the WIP to the change in interarrival time.

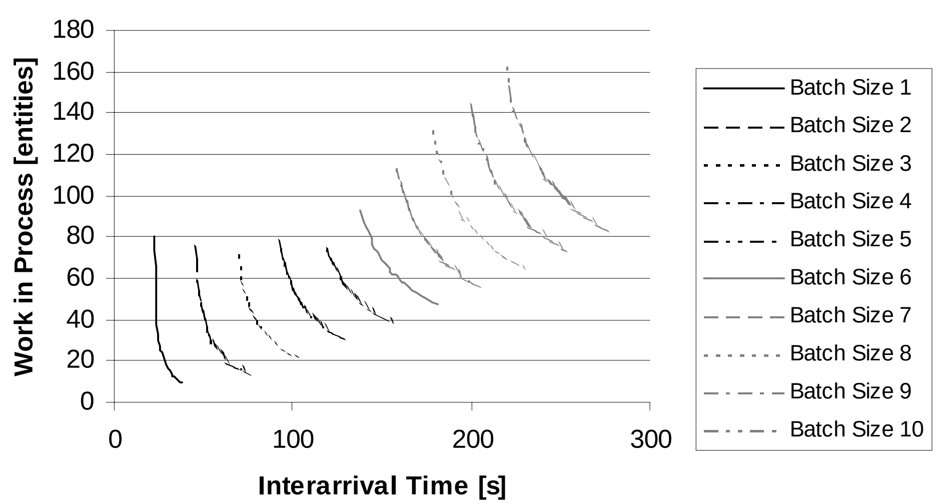


Figure 5-15: Work in process dependent on the interarrival time for different batch sizes for MRP.

Modeling the ten machine line by a M/M/1 system, the following condition has to be satisfied to reach a state of equilibrium [KLE75, p.95]:

$\lambda < \mu$,

where $\lambda$ is the birth rate or arrival rate and $\mu$ is the death rate or the inverse of the theoretical batch process time, $t_{process}^{batch}$,

$$\frac{1}{t_{intarr}} < \frac{1}{t_{process}^{batch}} \Leftrightarrow t_{intarr} > t_{process}^{batch},$$

where $t_{intarr}$ is the interarrival time.

As the interarrival time approaches the batch process time,

$$t_{process}^{batch} = bt_{process},$$

where $b$ is the batch size and $t_{process}$ is the process time of a single entity per machine, the work in process approaches infinity. For a batch size of 10 the following condition has to hold:

$t_{intarr} > 10(20) \Leftrightarrow t_{intarr} > 200$ seconds. These are the values the hyperbolas approach to the left in Figure 5-15. As the interarrival time approaches infinity, the WIP approaches zero. Thus, the hyperbolas all have y = 0 as an asymptote and can be elongated to the right. Assuming an interarrival time of 250 seconds, different WIP levels are reached for the different batch sizes. This results in flexible combinations of interarrival times and batch sizes to achieve certain WIP levels. However, the average utilization of the machines in the system has to be kept in the realistic interval of [0.65, 0.9] to make the findings relevant to practical application. Figure 5-16 shows the almost linear response of the utilization to the interarrival time. A closer look reveals a nonlinear influence of the interarrival time.

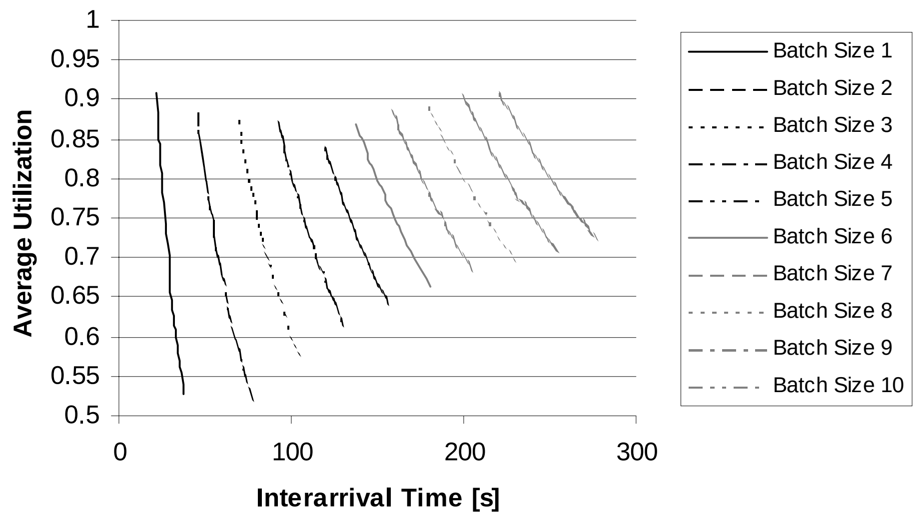


Figure 5-16: The average utilization of the line dependent on the interarrival time of the batches for MRP.

Thus, for the given utilization levels, and consequently the throughput, the average cycle time is expected to respond similarly to the work in process (Little's Law). Figure 5-17 proves this expectation to be correct.

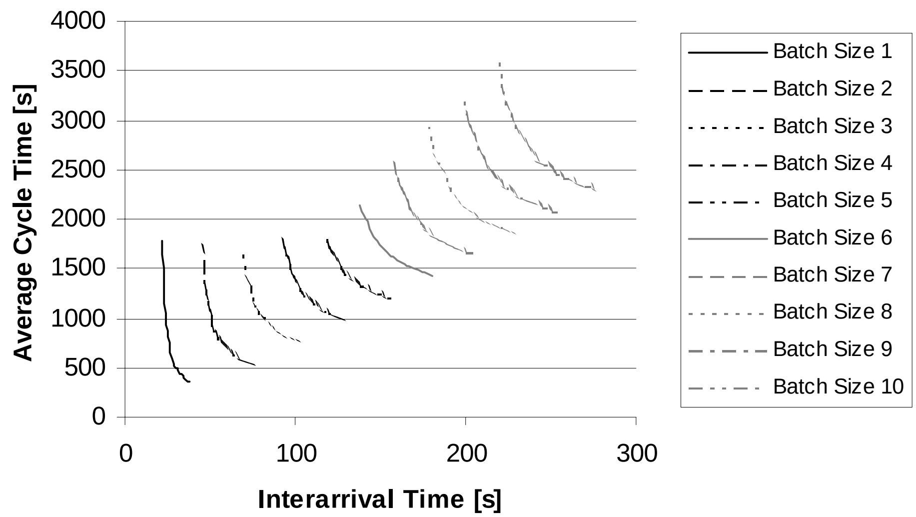


Figure 5-17: The average cycle time per entity dependent on the interarrival time for MRP.

However, the lowest cycle time per entity is dependent on the batch size. Thus, the y-asymptotes can be considered functions of the batch size. The reason was given at the beginning of chapter 5.2. The entities are introduced to the line in batches. The last entity of each batch has to wait until the other entities in the same batch are processed, increasing the average cycle time with increasing batch size.

A difference between MRP and the pull systems is the response of the average cycle time to the batch size. The models constructed for Kanban and CONWIP indicate a negative quadratic influence of the batch size [see Table 5-5]. As can be seen in Figure 5-18, the average cycle time responds linearly to an increase in the batch size. To determine this dependence, the throughput was held almost constant at 2.4 entities per minute or an average utilization of 0.8, keeping the theoretical throughput of 3 entities per minute in mind.

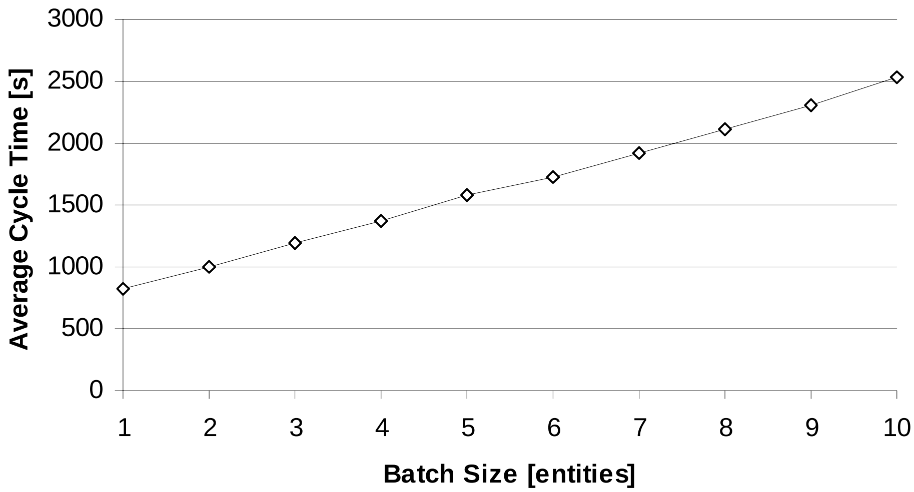


Figure 5-18: The average cycle time dependent on the batch size with a constant throughput for MRP.

### 5.2.5 Findings for MRP

The following points were observed:

1. If the interarrival time approaches the batch process time, the WIP-level increase exponentially:

   $$t_{intarr} \xrightarrow{t_{intarr} > t_{process}^{batch}} t_{process}^{batch} \Rightarrow Wip \rightarrow \infty .$$

   The system becomes unstable [see Figure 5-15],

2. The throughput or average utilization decreases at first rapidly and then slowly with increasing interarrival time,
3. The average cycle time decreases with an increasing interarrival time, but, it never reaches zero,
4. The batch size influences the average cycle time linearly, and

5. The average cycle time responds hyperbolically to the batch size, their asymptotes are functions of the batch size and interarrival time.

These observations lead to the following conclusions:

1. As the batch size increases, the average cycle time increases. For a given interarrival time, this is always true [see Figure 5-17]. The batch size should be chosen as small as possible,
2. Once the batch size is set to greater than one, a zero average cycle time can never be reached [see Figure 5-17, finding 5]. To ensure a high utilization of the line, an interarrival time close to batch process time should be implemented [see finding 1].

After discussing the response of the systems in more detail on a lower level of complexity, the following chapter continues the comparison of the systems on a higher level of abstraction.

## 5.3 Kanban, CONWIP, and MRP

In this section the performance of the different systems is briefly illustrated. At this point only first impressions are given. As the realism of the simulated configurations increases by adding setup times and failures, additional comparisons will reveal more information about their characteristics.

### 5.3.1 Average Cycle Time Dependent on Work in Process

For the comparison the optimal configurations, resulting in the lowest average cycle time, were extracted from the data set and illustrated [see Figure 5-19].

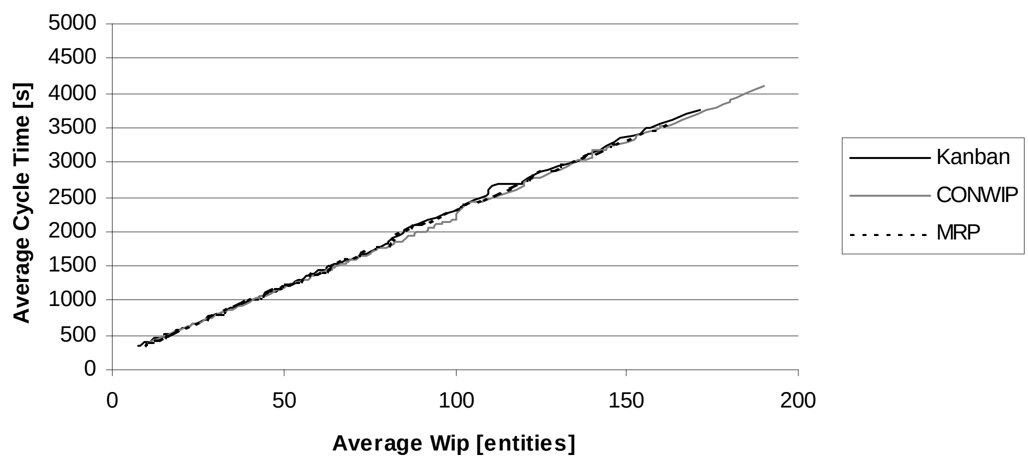


Figure 5-19: The minimal average cycle time dependent on the average work in process for the three control systems.

At this point maximum WIP and average WIP have to be clearly distinguished for Kanban. As previously mentioned, the smaller batch size results in a better performance of the system for a specific number of cards assigned, independent of the control mechanism [see Figure 5-7 and Figure 5-10]. However, for the pull systems the maximum WIP can be determined as a combination of batch size, *b*, and number of cards assigned, *c*,:

$WIP_{\max} = bc$.

Thus, as the maximum WIP level increases, the number of possible combinations of batch size and number of cards assigned increases.

In Figure 5-10 a WIP level of 60 can be obtained from 6 combinations for CONWIP. The extracted numbers are shown in Table 5-6.

Table 5-6: Combinations of batch size and number of cards for maximum WIP level 60 and the resulting average cycle times for Kanban and CONWIP.

| Combination | Batch Size | Number of Cards | CONWIP Average Cycle Time | Kanban Average Cycle Time |
|---|---|---|---|---|
| 1 | 1 | 60 | 1394.518 | 1101.557 |
| 2 | 2 | 30 | 1399.522 | 1183.894 |
| 3 | 3 | 20 | 1443.929 | 1243.223 |
| 4 | 4 | 15 | 1488.523 | 1210.624 |
| 5 | 5 | 12 | 1537.916 | 1432.689 |
| 6 | 6 | 10 | 1621.982 | 1554.002 |

Figure 5-20 shows a plot of the data points from simulation and the regression models. The difference between maximum WIP and average WIP can clearly be seen as the average WIP is lower for Kanban resulting in a shorter average cycle time. For a small batch size the model fits perfectly. Unfortunately, the quality of the fit decreases with increasing batch size. Most importantly, the response of the cycle time to the batch size is modeled well.

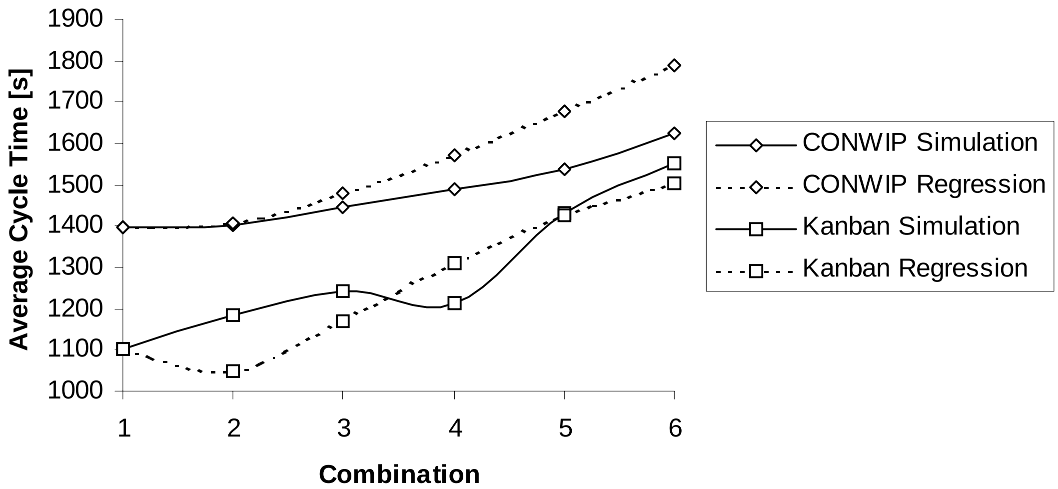


Figure 5-20: The average cycle time dependent on different combinations of batch size and number of cards assigned, simulation and regression model.

The term in the regression model expressing the interaction of the batch size and number of cards assigned is kept constant, resulting in the following equation for CONWIP:

$$\bar{t}_{cycle}^{Conwip} = 1130.5793 + 89.2901b - 2.6088b^2 + 2.8994c \,.$$

The influence of the $b^2$-term is compensated by the $c$-term, leaving the major influence on the average cycle time to the linear component of the batch size. Thus, the average cycle time increases with increasing batch size. The same interdependence can be derived for Kanban with the following regression model:

$$\bar{t}_{cycle}^{Kanban} = e^{6.1059 + 0.3204 Bsize - 0.0225 Bsize^2 + 0.0108c} \,.$$

In Figure 5-7 and Figure 5-10 the increase of the average cycle time with increasing batch size can be seen. However, the smooth line is broken by an exception. The combination batch size 4 and number of cards 15 is performing above expectations. The alert reader may have noticed this abnormality earlier. Looking closely at Figure 5-11 reveals that 15 cards assigned outperforms 14 cards, independent of the batch size. The clustering of data points for a specific batch size in Figure 5-7 (1) underlines this fact. Interestingly, the 15 cards assigned do not break the order of throughput. The throughput fits nicely between that of 14 and 16 cards assigned, independent of batch size. Disregarding this outlier and a few other exceptions that are dependent on batch size, an increase of average cycle time with an increase of batch size can be assumed.

Consequently, CONWIP's data extracted for Figure 5-19 represents the lowest batch size possible for a specific maximum WIP. At certain WIP levels, the difference in average cycle time between the combination with smallest batch size and the combination with higher batch size [see Table 5-6] is smaller, than the difference in average cycle time between two WIP levels *WIP* and (*WIP*+1) for the same batch size [see Table 5-7]. Adding another card to increase the WIP level of 77 to 78 results in a higher jump of the average cycle time than keeping the WIP level of 78 constant by decreasing the number of cards assigned while increasing the batch size.

Table 5-7: The increase in cycle time for increasing batch size and constant WIP.

| Batch Size | Work in Process | Average Cycle Time |
|---|---|---|
| 1 | 77 | 1726.513073 |
| 1 | 78 | 1758.674835 |
| 2 | 78 | 1763.326356 |
| 3 | 78 | 1775.199725 |
| 1 | 79 | 1775.474382 |

In Table 5-7 the data points with higher batch sizes 2 and 3 and WIP 78 would be dropped leaving the data point resulting from batch size one and WIP 78. Generally, the data points with higher batch sizes would be dropped, leaving those with the lowest cycle time. Thus, increasing the batch size (from 1 to 3) for a specific WIP level constant and reducing the number of cards assigned (78 to 26) may not be as harmful to the average cycle time as an increase in the WIP level by one (78 to 79). Here, the smaller batch size could be replaced by the higher batch size without considerable damage. This opportunity occurs at WIP 48 the first time. Batch size one is replaced by batch size two at WIP level 100, as simulations were only carried out for 100 or less cards assigned.

In Kanban batch size two outperforms batch size one at WIP level 49.46, where 32 cards are assigned. This card allocation seems to be superior in its WIP environment. Although the difference is small, batch size two manages to outperform batch size one for other card allocations as well.

MRP shows similarities to CONWIP. The smaller batch size outperforms the larger batch size without exceptions.

As the systems are strongly constrained they have little freedom of outperforming one another. Figure 5-19 shows almost no difference in performance for small work in process. Taking a closer look at the lower WIP level shows that the systems' performance is very much the same [see Figure 5-21].

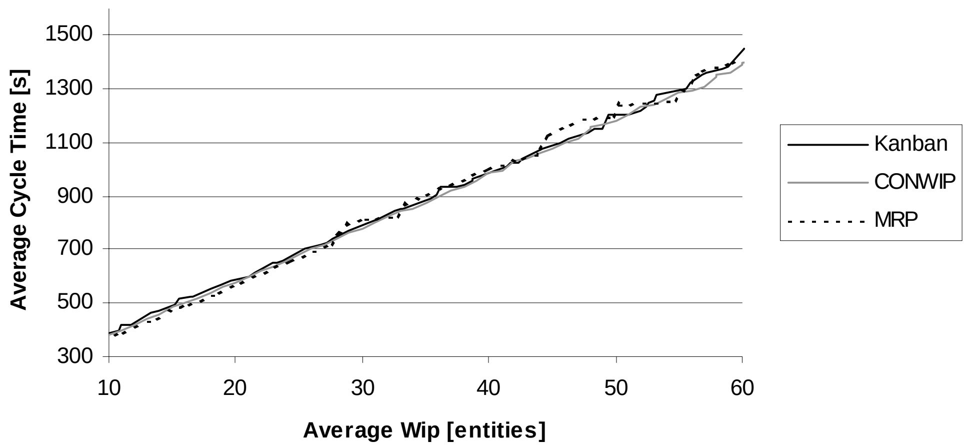


Figure 5-21: A closer look at the minimal average cycle time dependent on lower average work in process for the three control systems.

Figure 5-21 indicates that MRP performs best, CONWIP second best, and Kanban can be closely ranked on the third position for WIP levels less than 27. However, the differences are minute and not really worth considering. At WIP level 28.76 MRP introduces batch size 2, sacrificing its first position. The next data point with batch size 1 is at WIP level 32.74. Interpolation would reveal MRP’s behavior between the two WIP levels, suggesting MRP’s superiority. A regression analysis was performed for batch size one to statistically verify this ranking [see Table 5-8].

Table 5-8: The results for the regression analysis modeling the response of the average cycle time to the WIP for a comparison between Kanban, CONWIP, and MRP.

| Formula | | | | |
|---|---|---|---|---|
| Avgct ~ WIP + Ic + Im + IcWIP + ImWIP | | | | |
| Residuals | | | | |
| Min | 1Q | Median | 3Q | Max |
| -28.01 | -3.477 | 0.239 | 3.05 | 19.11 |
| Coefficients | | | | |
| Parameter | Value | Std.Error | t-value | Pr(>\|t\|) |
| (Intercept) | 195.0680 | 2.1733 | 89.7559 | 0.0000 |
| WIP | 19.8483 | 0.0638 | 310.9755 | 0.0000 |
| *Ic* | -16.1534 | 2.7468 | -5.8807 | 0.0000 |
| *Im* | -26.3223 | 3.5145 | -7.4896 | 0.0000 |
| *IcWIP* | 0.2272 | 0.0695 | 3.2683 | 0.0013 |
| *ImWIP* | 0.1573 | 0.1102 | 1.4266 | 0.1556 |
| Residual standard error | | | | |
| 6.906 on 163 degrees of freedom | | | | |
| Multiple R-Squared | | | | |
| 0.9998 | | | | |
| F-statistic | | | | |
| 181500 on 5 and 163 degrees of freedom, the p-value is 0 | | | | |

The regression verifies the ranking above. *Ic* and *Im* indicate the difference between the intercept for Kanban and CONWIP (*Ic*) and MRP (*Im*). MRP has the lowest intercept with a slightly higher gradient than Kanban. CONWIP has the second highest intercept and about the same gradient as MRP, as the probability of failing to reject the null hypothesis, stating a zero difference, is very high. The higher gradients for MRP and CONWIP than for Kanban indicate, that the lines cross over at some point. However, the difference is so small, that one can assume parallelism of the lines.

Figure 5-22 illustrates average cycle times for higher WIP levels. As already indicated in Figure 5-19 CONWIP performs best for the WIP levels from the upper fifties up to 100. This is attributed to the fact, that for CONWIP up to 100 cards were allocated for batch size one. As the other two control systems make use of at least batch size 2 at that level, they are outperformed by CONWIP. For WIP levels above 100 it becomes difficult to interpret the points attempting to rank the systems according to their performance. The data points may be misleading due to several factors:

- Different batch sizes are represented,
- Even though 10,000 units were produced narrowing the confidence interval on the mean down tremendously, the variance influences the output, and
- The output is discrete and not continuous. The systems could be performing equally well would their data points be in greater vicinity. As can be seen in Figure 5-22, drawing a straight line through the data points would only give an approximation.

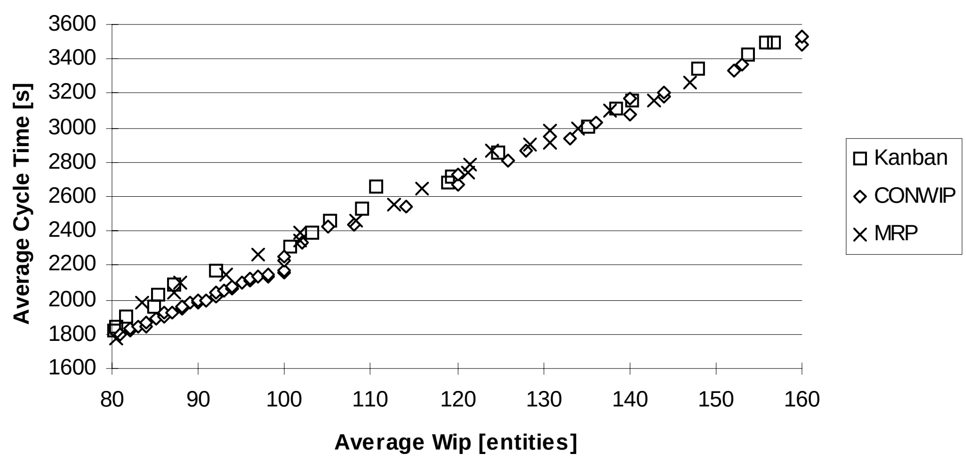


Figure 5-22: A closer look at the minimal average cycle time dependent on higher average work in process for the three control systems.

The previous investigation leads to the following conclusion: the smaller batch size outperforms the bigger batch size independent of the control system. In other words: although inferior to MRP, Kanban with batch size 3 would outperform MRP with batch size 4 for a given WIP level.

### 5.3.2 Conclusions

The observations lead to the following conclusions:

- For a given WIP level, an optimal configuration can be found for Kanban, CONWIP and MRP. With this configuration the systems can be ranked according to their performance:
  1. MRP,
  2. CONWIP, and
  3. Kanban.
- For a given WIP level a lower batch size is always superior to a higher batch size. This is true for all three control systems.
- The previous conclusion seems to hold even across manufacturing control systems as the difference in performance between the systems is minute.

After discussing the influence of the batch size on the performance of the manufacturing control systems, another parameter is added to increase the realism and the practical applicability of the investigations. In the following chapter setup is introduced.

# CHAPTER 6
# SETUP TIME

As mentioned earlier [see CHAPTER 5] batch size and setup are very much related. As the batch size increases, the setup time per unit decreases. Thus, more precisely, there is a trade off between the two parameters to optimize the performance of the manufacturing lines.

Spearman and Hopp distinguish between internal setup and external setup. Internal setup operations are those tasks that take place when the machine is stopped, while external setup operations are those tasks that can be completed while the machine is still running. Thus, the internal setup is disruptive to the production process and deserves the most attention [HOP96, p. 158]. Monden identifies four basic concepts to reduce setup:

1. Separate the internal setup from the external setup,
2. Convert as much as possible of the internal setup to the external setup,
3. Eliminate the adjustment process, and
4. Abolish the setup itself [MOD83].

Here, setup refers to the internal setup only.

Introducing setup to a system increases the variability. With similar reasoning as in process times [see 5.1.1], the setup times are also chosen to be exponentially distributed: $t_{setup} \sim M(setup\ time)$ and $t_{process} \sim M(process\ time)$, where $M$ stands for Markov. The setup can be viewed as another process step on a machine.

To begin with the actual processing of the first product, the machine has to wait for the setup to be completed. In the mean time the machine is running idle. Furthermore, the batch must be completed on the upstream machine before the setup on the downstream machine can begin. Baker refers to this feature as attached setup time, which can not be scheduled in anticipation of arriving work. The setups are classified as separable (or detached) when scheduling in anticipation is possible. In his article Baker considers the lot streaming model for a two-machine flow shop with setup times, transfer lots of size one, and a makespan objective. He expects the two-machine analysis to play a role in the development of heuristic procedures for the general *m* machine case, although he mentions that the efficient sequencing rules developed for the two machines will generally not extend to three-machine problems [BAK95]. Patterson focuses on constraint resources, building resources around bottlenecks. He concludes from the results of his study that a critical piece of information needed for finite scheduling is missing from the existing MRP database. The missing data field is a code to represent a setup procedure. His case study suggests that a setup procedure, and the required time, can be the same for multiple inventory items. Identifying these procedures can lead to the preparation of finite schedules that can improve due date performance and/or reduce overtime required to meet promised ship dates [PAT93]. Afyonoglu assesses the performance of a five machine line controlled by Kanban, CONWIP, and MRP with constant setup times. He evaluates the systems by determining their total costs including setup costs, inventory carrying costs, card costs, and penalty costs. His investigations result in practical thumb-rules, beneficial to practical applications [AFY98].

To express the availability of the machine, the setup time was simply added to the batch process time resulting in a new and longer delay time, $t_{delay}$, for which an arriving entity has to wait in the queue to be processed:

$$t_{delay} = t_{setup} + t_{process}, \tag{6-1}$$

where $t_{setup}$ and $t_{process}$ refer to batch setup times and batch process times. As the machine has to wait for the setup to be complete and can not process simultaneously, the theoretical throughput [see 3.2.6] has to be adjusted with equation (5-1):

$$Th_{theory} = \frac{1}{t_{delay}} = \frac{1}{t_{setup} + t_{process}}.$$

This results in the following machine utilization:

$$u = \frac{Th_{average}}{Th_{theory}} = Th_{average}\left(t_{setup} + t_{process}\right). \tag{6-2}$$

A new parameter is added to the models representing the setup while the other parameters remain the same.

## 6.1 Parameters

To enable a comparison, the average utilization was held constant across the three control systems. Furthermore, a low and a high utilization was chosen to illustrate the performance of the systems under high and low demand conditions. The number of cards assigned were held constant resulting in specific utilizations for varying batch sizes and setup times. If these parameters were not held constant, too many factors would have influenced the output and it would have become very difficult to unveil any dependencies. For MRP the interarrival time was adjusted to produce the utilization levels resulting from the card allocation for the pull systems.

The batch size, as explained earlier [see 5.1.2], and the setup ratio were chosen as the variable parameters.

### 6.1.1 Setup Ratio

As the process time was held constant throughout all the simulations, the setup time would have been a good parameter as well. However, the setup ratio,

$$r_s = \frac{t_{setup}}{t_{process}},$$

is an established parameter and has a higher information content as it expresses the relative length of the setup time to the process time. Table 6-1 shows the setup times and the corresponding ratios included in the simulations.

Table 6-1: The setup times and the corresponding setup ratios included in this study.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $t_i^{setup}$ | 2 | 10 | 20 | 40 | 60 | 80 | 100 | 120 | 140 | 160 | 180 | 200 |
| $r_{si}$ | 0.1 | 0.5 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

### 6.1.2 Utilization

For the pull systems the number of cards allocated and for MRP the interarrival time had to be determined resulting in the same utilization for all three control systems. The introduction of setup times to the models resulted in a decrease of utilization. As the number of cards allocated to the manufacturing lines was kept in the interval [10, 70] throughout all the experiments [see 5.1.3 Number of Cards, Card Allocation for Kanban] conducted in this study the initial utilization level interval of [0.65, 0.9] [see 5.2.1, p. 77] had to be altered to [0.55, 0.86]. Only more than 70 cards assigned could raise the upper utilization level to 0.9. The resulting configurations are presented in Table 6-2.

Table 6-2: Configuration chosen to establish high and low utilization levels.

| Utilization Level | Batch Size | # of cards for Kanban | # of Cards for CONWIP |
|---|---|---|---|
| high | 1 | 70 | 50 |
| high | 2 | 50 | 37 |
| high | 3 | 33 | 27 |
| high | 4 | 25 | 21 |
| high | 5, 6, 9, 10 | 20 | 17 |
| high | 7, 8 | 20 | 18 |
| low | 1 | 14 | 10 |
| low | 2 – 10 | 13 | 10 |

The number of cards assigned for Kanban exceed those assigned for CONWIP as for Kanban the number indicates the upper limit of WIP in the system, for CONWIP the number indicates the exact number of WIP in the system [see 4.2.2]. The chosen number of cards gave rise to certain utilization levels for the varying batch size and setup ratio. To determine the interarrival time of batches for MRP, the following calculations were done with equation (5-2):

$$\left.\begin{array}{l} \bar{u} = Th_{average}\left(t_{setup} + t_{process}\right) \Leftrightarrow Th_{average} = \dfrac{\bar{u}}{\left(t_{setup} + t_{process}\right)} \\ Th_{average} \approx \dfrac{1}{t_{intarr}} \Leftrightarrow t_{intarr} \approx \dfrac{1}{Th_{average}} \end{array}\right\} \Rightarrow t_{intarr} \approx \frac{\left(t_{setup} + t_{process}\right)}{\bar{u}},$$

for a stable system. Thus, the interarrival time for MRP was calculated from the setup time, process time, and average utilization. However, these were purely theoretical terms, that needed some fine tuning.

The fine-tuning was done by interpolation. This could be done as the utilization, or throughput, is linearly dependent on the interarrival time for most of the utilization intervals priorly chosen. Once the interarrival time equals or exceeds the bottleneck processing time (here, the theoretical average cycle time), the system's WIP grows exponentially [see Figure 5-15], finally resulting in an unstable system. Parallely, the average throughput approaches the theoretical throughput while the utilization approaches one. Here, linear interpolation would lead to the wrong results. The following calculations were done:

$$\frac{t_2^{intarr} - t_1^{intarr}}{\bar{u}_1 - \bar{u}_2} = \frac{t_2^{intarr} - t_{intarr}}{\bar{u} - \bar{u}_2} \Leftrightarrow t_{intarr} = \frac{\left(t_2^{intarr} - t_1^{intarr}\right)\left(\bar{u} - \bar{u}_2\right)}{\bar{u}_2 - \bar{u}_1} + t_2^{intarr},$$

where index 1 refers to the high level values and index 2 refers to the low level values and $\bar{u}$ is the wanted average utilization resulting from the calculated interarrival time, $t_{intarr}$. Fortunately, it was found that the average cycle times resulting from simulating for 1,000 entities was very close to the actual value resulting from processing 10,000 entities. Thus, the fine tuning could be done in less time.

Another approach to find the right utilization levels could have been to run simulations on intervals of interarrival times close to the computed theoretical value. The major advantage of this approach would be that the simulations could be run without attendance. Furthermore, no interpolation calculations would have to be done. The additional simulation time would be only a few hours.

The previous activities resulted in a data set based on roughly the same utilization for the three control systems. A difference of 0.03 in utilization was exceeded in only 8 cases out of 242, as the pull systems result in discrete values, making it impossible to obtain the exact same values [see Figure 6-1].

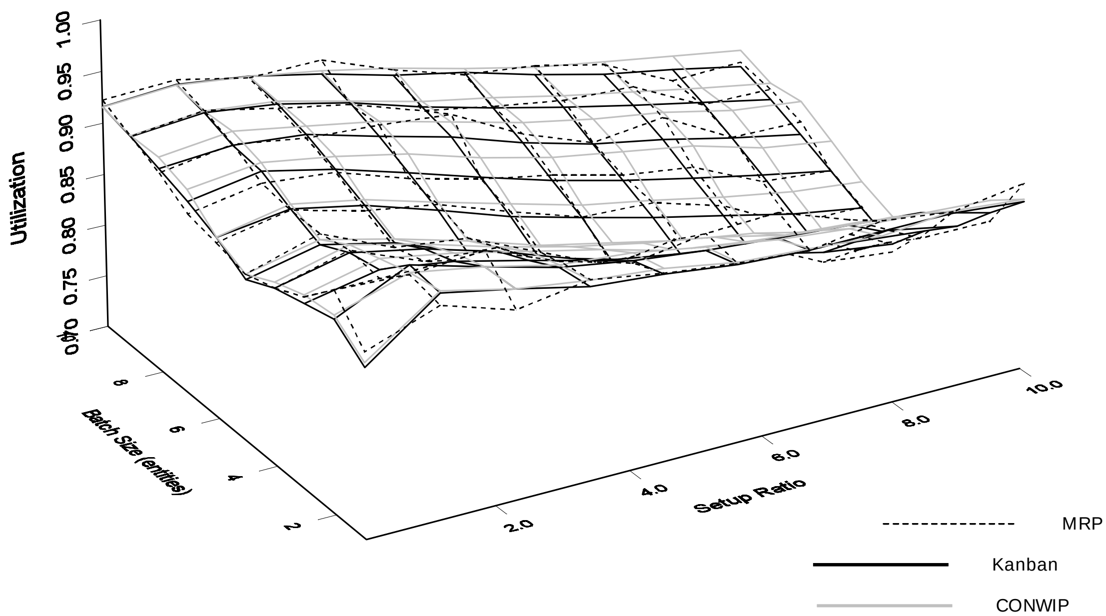


Figure 6-1: The higher utilization level dependent on the setup ratio and the batch size for Kanban, CONWIP, and MRP.

Although the graph appears to be overloaded with information, it gives a very good impression of the output data. Furthermore, it is easily read, as the individual data points for a given system are represented by two lines intersecting. To compare the three systems for a given setup ratio and batch size, an imaginary vertical line can be drawn through the three intersections and the data points can be read from top to bottom. Taking the data points for setup ratio 10 and batch size 9, thus $(10, 9, \bar{u}_{Kanban})$, $(10, 9, \bar{u}_{CONWIP})$, and $(10, 9, \bar{u}_{MRP})$, reading from top to bottom, it can easily be seen, that CONWIP has the highest value, MRP the intermediate value, and Kanban the lowest value. Although the deviation is small (the average utilizations are 0.8110, 0.8048, and 0.7945) it can be distinguished in the graph. Table 6-3 exhibits the output for the two-sided paired t-tests performed on the difference of utilization.

Table 6-3: Output for the paired t-tests on difference of high utilization including setup for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of differences estimate |
| Kanban-CONWIP | -12.8163 | 119 | 0 | (-0.0117; -0.0086) | -0.01016293 |
| CONWIP-MRP | 6.2253 | 119 | 0 | (0.0053; 0.0102) | 0.007766142 |
| Kanban-MRP | -2.3792 | 119 | 0.0189 | (-0.0044; -0.0004) | -0.002396792 |

The utilizations can not be considered the same, as zero is in none of the confidence intervals. However, the differences are found to be very small, indicated by the confidence interval and the estimate of the mean difference. The output data shows the following order of utilization:

$$\bar{u}_{CONWIP} > \bar{u}_{MRP} > \bar{u}_{Kanban}.$$

Keeping the setup ratio constant, the three systems unveil a linear response to the batch size with a minimum for batch size 5, increasing for both an increasing or decreasing batch size [see Figure 6-1]. The systems showed similar behavior not including the setup [see Figure 5-7]. Note that the number of cards assigned is constant, too. It is one of the fixed parameters mentioned earlier [see 6.1]. For an increasing setup ratio, the utilization decreases somewhat, while the increase to either side of batch size five, per unit increase of batch size, becomes bigger.

Figure 6-2 illustrates the difference between the utilization and the throughput as defined earlier [see 6.1.2]. The graph clearly indicates, that it is impossible to hold a high throughput with increasing batch size [see Figure 5-6] and setup ratio. The throughput does not incorporate the utilization of machine capacity for setup, which the utilization does. Furthermore, the graph states that a comparison between the different systems for a given setup ratio and batch size can be made, as the throughput is almost identical, the surfaces lie very close together.

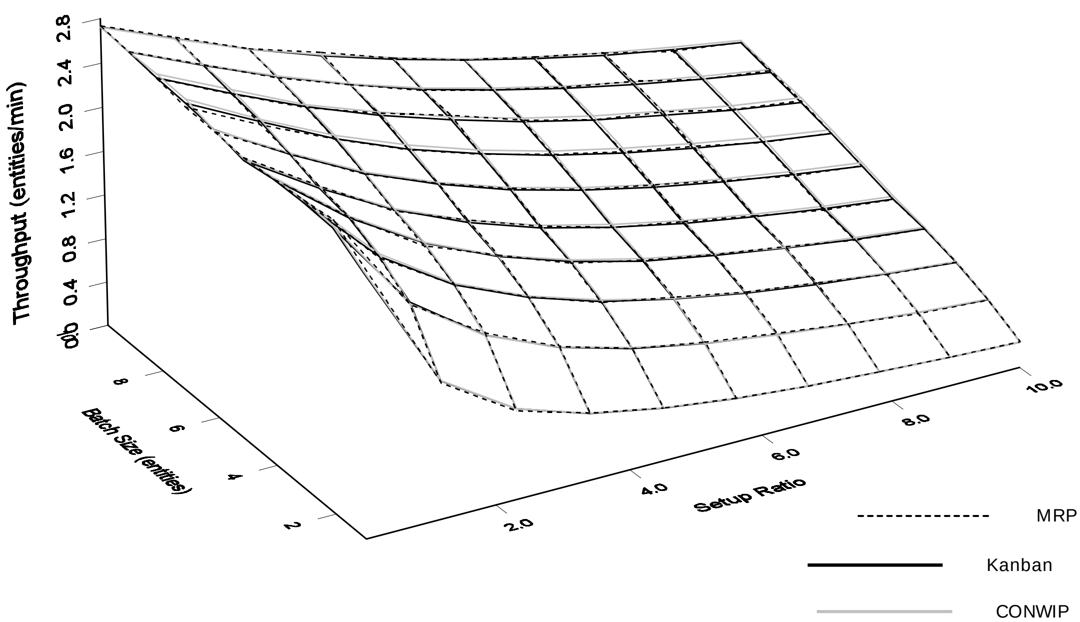


Figure 6-2: Throughput dependent on the setup ratio and the batch size for Kanban, CONWIP, and MRP.

For every batch entering a machine, the actual processing of the entities is delayed for the time needed to set the machine up for individual requirements. Obviously, this time has a great effect on the throughput when the batch size is small. For batch size one, the machine has to be set up for every single entity, slowing down the process and increasing the average cycle time tremendously. As the setup ratio increases, this effect is multiplied reducing the throughput to a mere 0.23 entities per minute for setup ratio 10. However, as the batch size increases, the influence of the setup time decreases. The delay time $t_l^{delay}(b)$ of the *l*th entity is dependent on the batch size *b*,

$$t_l^{delay}(b) = \sum_{j=1}^{h} t_j^{setup} + \sum_{i=1}^{l} t_i^{process}, \qquad (6\text{-}3)$$

where $h = \left\lceil \frac{l}{b} \right\rceil$ is the number of setups prior to the *l*th entity, $t_j^{setup}$ is the *j*th setup time, and $t_i^{process}$ is the process time of entity *i*. As the batch size decreases, *h* increases, increasing the amount of setups prior to the time when entity *l* reaches the machine. The setup time and the process time are entity dependent as they are probabilistic variables. But, for a large number of entities processed, these variables approach their expected value and can be assumed constant.

The throughput seems to respond quadratically to the setup ratio. As the ratio increases from below one, the throughput quickly decreases, evening out for higher setup ratios. The batch size has an inversely multiplicative effect on the increasing throughput for a decreasing setup ratio.

Keeping one parameter, the utilization, constant for the three control systems, their performance can easily be compared. The performance of the control systems is compared by analyzing the average cycle time for a high utilization level and for a low utilization level.

## 6.2 Average Cycle Time (High Utilization)

To continue the discussion from previous chapters about the average cycle time, its response to the setup ratio and batch size is illustrated below [see Figure 6-3]. For a low setup ratio, the cycle time shows the patterns formerly discussed [see Figure 5-7].

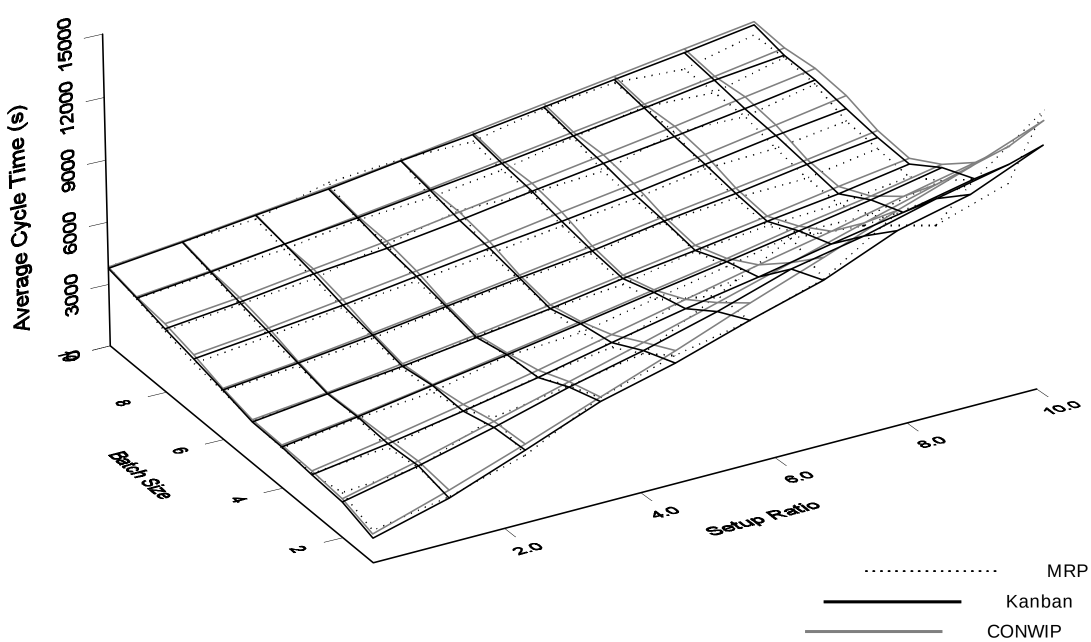


Figure 6-3: The average cycle time dependent on the setup ratio and the batch size for Kanban, CONWIP, and MRP.

The line shows a slight negative quadratic curvature with a dent for batch size five continuously increasing with increasing batch size [see Figure 6-4].

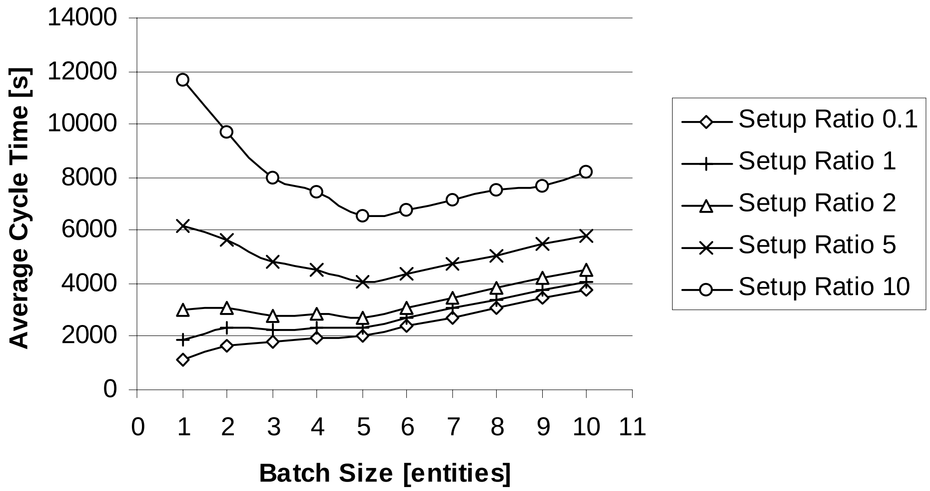


Figure 6-4: The average cycle time dependent on the batch size and setup ratio for Kanban.

As the setup ratio increases, its influence on especially the smaller batch sizes augments while shifting the lines up parallelly. The lines start to curve up for batch sizes smaller than five. As the setup ratio reaches ten, the cycle time for batch size one surmounts the other batch sizes. The dent in the line for batch size five remains, making it the superior configuration. This phenomenon can be easily explained with the assistance of the formula for the delay time per entity given before [see equation 5-3 on p. 113]. The smaller the batch size, the more setups have to be performed for a given amount of entities. As the setup time increases, the influence of these setups grows.

After the discussion of the system behavior, a comparison of the performance follows in the next section.

### 6.2.1 Comparison

Figure 6-3 reveals very similar behavior of the three systems. At first sight it is impossible to determine the best performer. Especially for the lower setup ratios the average cycle time seems to be almost identical. As the ratio increases, the systems commence to distinguish themselves. Drawing imaginary lines, the cycle times unveil the following order:

$$\bar{t}_{cycle}^{CONWIP} > \bar{t}_{cycle}^{Kanban} > \bar{t}_{cycle}^{MRP} .$$

To prove this finding, paired t-tests were performed. The output is listed in Table 6-4.

Table 6-4: The output of the paired t-tests on the difference between the average cycle times for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -8.7197 | 119 | 0 | (-246.4751; -155.2500) | -200.8625 |
| CONWIP-MRP | 9.2561 | 119 | 0 | (295.6708; 456.5986) | 376.1347 |
| Kanban-MRP | 4.5673 | 119 | 0 | (99.2851; 251.2593) | 175.2722 |

The tests support the former findings and permit the construction of the following ranking:

1. MRP,
2. Kanban, and
3. CONWIP.

Figure 6-5 illustrates the mean of differences graphically.

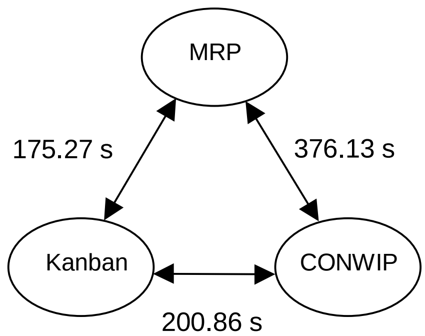


Figure 6-5: The mean differences of the average cycle times between Kanban, CONWIP, and MRP for the high utilization level.

Recalling the difference in utilization and the resulting relation,

$\bar{u}_{CONWIP} > \bar{u}_{MRP} > \bar{u}_{Kanban}$ ,

MRP is truly the best performer. The push system has the least average cycle times although it has to settle with a higher utilization than Kanban. However, CONWIP has the highest utilization which may result in the longest cycle times. Thus, position 2 and 3 can not be reliably stated. Table 6-5 shows the mean of the average cycle time relative to the mean of the utilization for the three systems.

Table 6-5: The mean of the average cycle time relative to the mean of the utilization for Kanban, CONWIP, and MRP.

| System | $E(\bar{t}_{cycle})$ | $E(\bar{u})$ | $E(\bar{t}_{cycle})/E(\bar{u})$ |
|---|---|---|---|
| Kanban | 4913.26 | 0.85 | 5780.306 |
| CONWIP | 5114.12 | 0.86 | 5946.651 |
| MRP | 4737.98 | 0.85 | 5574.094 |

The calculations were done to standardize the average cycle times with the expected utilization enabling a comparison. The numbers support the previous ranking.

### 6.2.2 Conclusions

The following conclusions summarize the former discussions:

The influence of the setup time on the average cycle time increases with decreasing batch size. Thus, for a big batch size an increase in setup time is not as detrimental for the performance of a system as for a small batch size.

Batch size five separates the batch sizes into those with a higher sensibility to change and those with a lower one. It encompasses the advantages of a small batch size, viz. a lower average cycle time, and the good attributes of the big batch sizes, viz. a lower reactivity to a change in setup time.

Observing the average cycle time dependent on the batch size and the setup ratio, MRP outperforms the two pull systems. In contradiction with the previous findings [see 5.3.2], Kanban shows a better performance than CONWIP.

## 6.3 Average Cycle Time (Low Utilization)

Although less relevant to its practical applicability, the average cycle time resulting from the low utilization level is investigated briefly. Again, the utilizations have to be compared. The output for the paired t-test is given in Table 6-6.

Table 6-6: Output for the paired t-tests on difference of low utilization including setup for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -5.6057 | 119 | 0 | (-0.0057; -0.0027) | -0.004232567 |
| CONWIP-MRP | -2.3045 | 119 | 0.0229 | (-0.0045; -0.0003) | -0.002426025 |
| Kanban-MRP | -9.2092 | 119 | 0 | (-0.0081; -0.0052) | -0.006658592 |

The p-value for comparing the average utilization of CONWIP and MRP is fairly high indicating that the difference in utilization is very small. The mean of differences estimate equals -0.002426025. As the difference is very small, one can assume equal utilization for the three systems for the low level.

The average value for the low utilization level of 0.67 can be considered a fairly relevant level, resembling a manufacturing system barely working at full capacity. Figure 6-6 illustrates the resulting average cycle time planes for the two utilization levels.

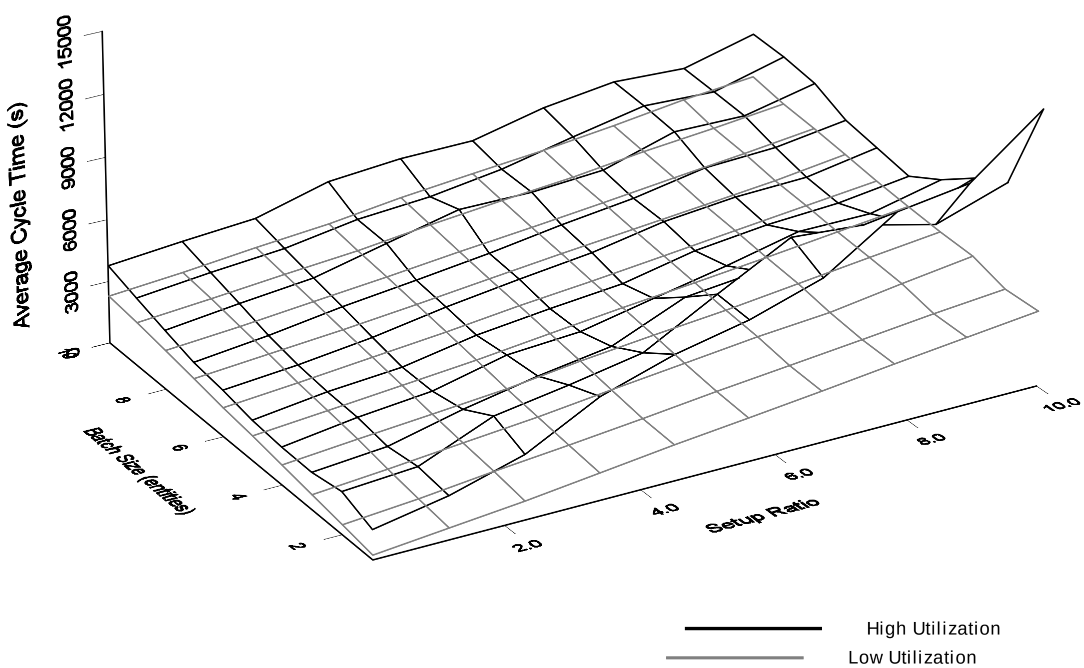


Figure 6-6: The average cycle time dependent on the setup ratio and the batch size for the high (0.85) and the low utilization level (0.67).

Naturally, there is a decrease in average cycle time for the lower utilization level, as a lower throughput results in a shorter cycle time (Little’s Law). Interestingly, the quadratic response of the cycle time to the batch size has vanished. A mere linear dependency is the result. The average cycle time increases linearly with an increasing setup ratio and an increasing batch size. Furthermore, a high setup ratio for a small batch size does not have the immense impact as with a high utilization level. Note that the utilization includes the setup as well. For an increasing setup time, the machines decrease processing, which results in a throughput of 0.15 entities per minute for setup ratio ten and batch size one. For this configuration the average cycle time closely surpasses 4,000 seconds per entity. These 4,000 seconds include 10 times setup for 200 seconds, which adds up to 2,000 seconds. As the throughput indicates, the system is fairly empty allowing the entities to pass through rather rapidly. Thus, 2,000 seconds remain for pure processing time, which sounds very reasonable. As the batch size increases, the entities spend more time waiting for the other members of their batch to be processed. Furthermore, more entities are pulled into the system, as the number of cards stands for the amount of batches in the system. Seemingly, at this utilization level, which counterbalances the increasing setup with a decreasing throughput, the effects resulting from system congestion remain absent.

### 6.3.1 Comparison

Understanding the system behavior better, the performance comparison remains. Figure 6-7 illustrates the average cycle time planes for Kanban, CONWIP, and MRP.

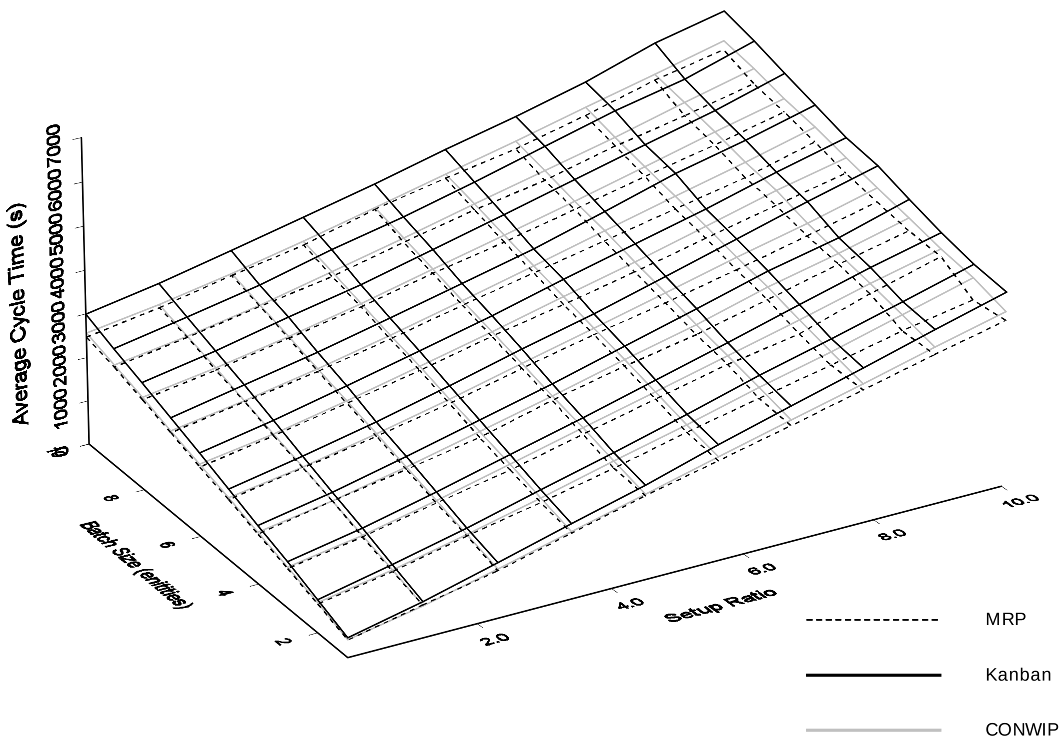


Figure 6-7: The average cycle time for the low utilization level dependent on the setup ratio and the batch size for Kanban, CONWIP, and MRP.

Even a brief look at the graph reveals the superiority of MRP to the two pull systems. As the batch size and the setup ratio increases, this domination becomes more apparent. However, CONWIP manages to cling very close to its related pendant. Although the order,

1. MRP,
2. CONWIP, and
3. Kanban,

was palpable, paired t-tests were performed to quantify the difference. The result is illustrated in Figure 6-8.

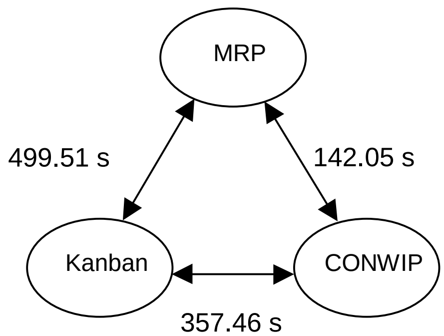


Figure 6-8: The mean differences of the average cycle times between Kanban, CONWIP, and MRP for the low utilization level.

### 6.3.2 Conclusions

The low utilization ranking differs from the high utilization ranking. However, it corresponds to the ranking not incorporating setup [see 5.3.2]. MRP performs well, outranking CONWIP and Kanban. CONWIP produces lower average cycle times than Kanban, positioning this control system at number two.

## 6.4 Regression Models

The models reveal a very complex interdependency of the different variables. For the regression, the fixed parameter, number of cards and interarrival time, were introduced as variables as well. This resulted in more than 1,000 data points for the pull systems and close to that number for MRP. In the model the average cycle time is represented as a function of the batch size, number of cards assigned to the system or the interarrival time for MRP, and the setup ratio [see Table 6-7].

Table 6-7: The regression models for the average cycle time as functions of the batch size, the number of cards assigned to the system or the interarrival time for MRP, and the setup ratio and their corresponding multiple coefficients of determination, $R^2$.

| System | Model | $R^2$ |
|---|---|---|
| Kanban | $\bar{t}_{cycle} = e^{x}$<br>$x = 0.4136\mathrm{r_s} + 0.2492\mathrm{b} + 0.0093\mathrm{c} - 0.0355\mathrm{br_s} + 0.0066\mathrm{bc} +$<br>$0.0023\mathrm{cr_s} + (0.0011\mathrm{b} - 0.0002\mathrm{c})\,\mathrm{r_s^2} + (-0.0003\mathrm{c} + 0.0008\mathrm{r_s})\mathrm{b}^2 +$<br>$0.0014\mathrm{r_s^3} + 0.0004\mathrm{b}$ | 0.9856 |
| CONWIP | $\bar{t}_{cycle} = 122.5767 + 97.8285r_s + 64.7392b + 2.15c - 3.358br_s +$<br>$17.0986bc + 20.0374cr_s + 6.1521r_s^2 - 0.3728b^2$ | 0.9998 |
| MRP | $\bar{t}_{cycle} = 806.2587 + 1501.8485\mathrm{r_s} + 1184.4921\mathrm{b} - 44.1563\mathrm{t}_{intarr}$<br>$+ 445.7524\mathrm{br_s} - 14.7163\mathrm{bt}_{intarr} - 23.4810\mathrm{r_s t}_{intarr} + 419.7524\mathrm{r_s^2}$<br>$+ 142.0208\mathrm{b}^2 - 0.3793\mathrm{t}_{intarr}^2 - 2.9357\mathrm{r_s^3}$ | 0.9818 |

The multiple coefficients of variation indicate a good fit of the models. Unfortunately, the cubic terms for modeling Kanban were needed, as $R^2$ would otherwise decrease by almost 2 percentage points. The residual plot indicated a cubic response of the average cycle time for Kanban only. Introducing the cubic regressor to MRP could not necessarily be derived from the residual plots. By stepwise regression and trial and error, the mean square error could be reduced, simultaneously increasing the multiple coefficient of determination while keeping the p-values all equal to zero. CONWIP could easily be modeled with the findings derived from the former analyses. No interaction between a quadratic regressor and a linear regressor could be found for CONWIP and MRP in contrast to Kanban. As the utilization for MRP increased and approached the theoretical maximum of one, the model was found not to be representative anymore. The values resulting were categorized as outliers. Thus, for very high utilization levels, the given model should not be applied.

After investigating the influence of batch sizing and setup, another step is taken to explore a manufacturing system's behavior. The next step in improving the realism of the simulated system is including machine failure.

# CHAPTER 7
# MACHINE FAILURE

In the previous experiments machines are assumed to be available full time. However, realistic systems are imperfect, machines suffer break down obstructing the flow of material in manufacturing. Thus, for every break down the availability of a machine is reduced. Other unscheduled downtimes may result from shortages caused by human failure. They result in the machine being inactive.The term failure will be used to indicate the unavailability of a machine. The availability of the machine can be calculated by the following formula:

$$a = 1 - \frac{t_{repair}}{t_{repair} + t_{intfail}} \Leftrightarrow a = \frac{t_{intfail}}{t_{repair} + t_{intfail}},$$

where $t_{repair}$ is the *repair duration* and $t_{infail}$ is the *interfailure time*. Hopp and Spearman refer to these times as mean time to failure and mean time to repair [HOP96, p.261].

Figure 7-1 provides a dynamic graph of the state of an unreliable machine. The failures occur at certain time instances [$t_0$ and $t_2$ ]. After a failure, it takes some time to make the machine available for production again. This time interval [$t_1$ - $t_0$], will be called the *repair duration*. The time instance when the machine can continue processing [$t_1$ ] will be referred to as *reactivation*. Finally, the *interfailure time* represents the time interval between reactivation and the next failure [$t_2$ - $t_1$].

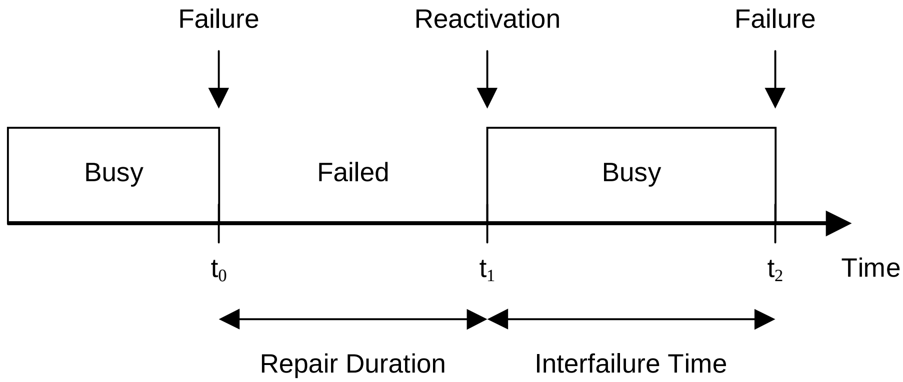


Figure 7-1: Resource states and their occurence times.

Here, it will be assumed that the product will not be damaged or scrapped if a failure occurs. The machine will resume processing the entity as soon as it becomes available again [see Figure 7-2].

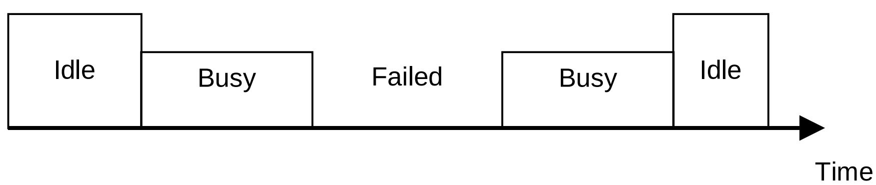


Figure 7-2: The effect of failure on the entity.

Gupta and Al-Turki explore the impact of sudden breakdown of a material handling system on the performance of a traditional Kanban system (TKS) with constant processing times. In addition, they also study a newly developed Kanban system, which dynamically and systematically manipulates the number of kanbans in order to offset the blocking and starvation caused by these factors during a production cycle (FKS), under the same conditions. They compare the overall performance of the

TKS and the FKS by considering a variety of cases. They come to the conclusion that the overall performance of the FKS exceeds that of the TKS [GUP98].

Hopp and Spearman consider a tandem line with a CONWIP control strategy. They also assume the processing times to be deterministic, but, machines are subject to exponential failures and repairs: deterministic processing and random outages (DPRO). They model the system as a closed queuing network and develop an approximate regenerative model (ARM) for estimating throughput and cycle time as a function of WIP level. Furthermore, Hopp and Spearman use an analytical alternative to simulation, mean value approximation (MVA), for analyzing the system. For MVA the processing times are chosen to match the mean processing times of the real system. Their comparisons show that ARM is more robust than MVA. They also observe that in order to get MVA to work significantly better than ARM, quite unrealistic parameters have to be chosen. In their opinion the ARM approach to approximating throughput in a DPRO system under a CONWIP control system appears promising [HOP91].

Duenyas et al. model a CONWIP production line with deterministic processing times and exponential failure and repair times as a closed queuing network as well. They derive an approximation for the mean and variance of the output during a specified interval and give computable conditions under which this approximation performs well. They show through empirical tests that the approximation is robust and illustrate its usefulness as the basis for a procedure for selecting an economic production quota and card count for a CONWIP line. Duenyas et al. admit to the simplicity of the system studied, consisting of a single product line of single machine work stations. In more realistic cases, with additional complicating factors, detailed analysis of the quota and card count issue is likely to be integrated with analysis of capacity and staffing issues, and as such are likely to require simulation [DUE93].

Tan determines the variance of the throughput on an *N*-station production line with constant processing time, no intermediate buffers, and time dependent failures analytically. Time to failure and time to repair distributions are assumed to be exponential. Tan mentions that state-space based methods are very flexible allowing various assumptions to be implemented in a model. However, these methods are computationally very demanding. Instead, he applies an inflexible method, only considering time dependent failures, which is computationally very efficient. With the given procedure, a numerical result for a production line with a given number of stations can easily be obtained [TAN97].

Unfortunately, all of the studies above use deterministic processing times. As Duenyas et al. remarks, more realistic systems with additional complicating factors including probabilistic processing times and setup times are likely to require simulation. Here, simulation is applied to study the effects of failure on the performance of the different control systems.

## 7.1 Parameters

To include failure in the system, the parameters discussed above were introduced to the models:

1. Interfailure time and

2. Repair duration.

By manipulating these parameters, the performance of the three control systems could be evaluated. Due to the growing number of parameters influencing the systems, initial investigations were carried out for less complex cases. As previously, the amount of cards assigned to the line was kept constant. Additionally, only one machine was subjected to failure. To get better insight into the systems' behavior, the observations were done dynamically.

## 7.2 Dynamics of Failure

Instead of looking at the results at the end of each replication, the data was output into a file continuously, revealing dynamic behavior. To unveil the reaction of the systems to a failure, one single downtime was induced at a known time. This point in time had to exceed the time instance for reaching steady state enabling an objective comparison. The failure time was chosen such that 5,000 entities would have passed through the system, resulting in the desired steady state and producing stable statistics. A repair duration of 3,600 seconds or one hour would represent a realistic event.

Good indicators had to be chosen to give unblurred information about the systems’ behavior.

### 7.2.1 Indicators

Cumulative average values would be blurred by the large amount of entities produced prior to failure. Furthermore, data revealing the state of the entire line would be cumulative as well. The data on the individual machines would be accumulated, loosing precious information. Thus, data based on individual entities would provide the highest information content. This data could later be compressed to give insight into the systems' performance.

As throughput (utilization) was fixed, the WIP level and cycle time remained as performance measures to compare alternatives. Generally, WIP is a difficult measure to obtain in a real life setting. However, the cycle time is commonly available in a shopfloor. Therefore, the cycle time was chosen as the performance indicator to observe system behavior. The following information could be revealed from the cycle time per entity:

1. The time spent at certain points in the system, and
2. The time taken by the system to recover from the disruption.

After deciding on the indicator, the following methods of data collection were chosen to obtain the desired information.

#### 7.2.1.1 Time Spent in System

The cycle time per entity had to be collected at different points in the line only after the failure occurred. Collecting the data before the occurance would result in excess data irrelevant to the investigation. Machine 5 was designated as the machine to fail as it is situated in the middle of the line. Consequently, the cycle time was calculated at six different passing points, before each downstream machine and after the last machine in the line [see Figure 7-3]. These points would reveal the time spent in every machine and the time of departure from the system.

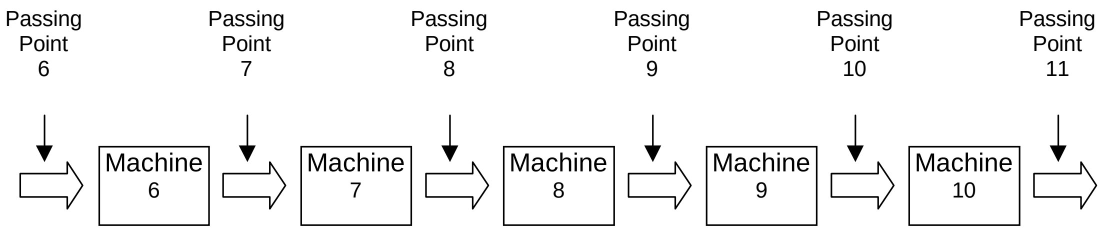


Figure 7-3: The points of data collection for the investigation on the dynamics of failure.

The passing points are named according to the machines they precede, making them easy to locate. Passing point 11 allows the calculation of the departure time.

### 7.2.1.2 Recovery Time

The average cycle time was determined prior to failure for 5,000 entities processed. All entities trapped upstream from machine 5 would increase their cycle time by at least the duration of failure. As these entities depart the system, the cycle times approach their average and the system reestablishes steady state, indicating the time of recovery. Looking at the individual cycle times would subject the determination of the time of recovery to too much variability. Thus, the exponentially smoothed cycle time,

$$\bar{t}^{exps}_{cycle,i} = 0.5(\bar{t}^{exps}_{cycle,i-1}) + 0.5(t_{cycle,i}) ,$$

was computed for the *i*th entity departing the system, where $t_{cycle,i}$ is the cycle time of the *i*th entity and $\bar{t}^{exps}_{cycle,1} = t_{cycle,1}$, starting with the first entity leaving machine 5 after reactivation. The recovery time, $t_{recover}$, was determined as the time taken until the moving average of the cycle times exceeded the average cycle time by less than ten percent:

$\bar{t}^{exps}_{cyle,i} \leq 1.1(\bar{t}_{cycle})$. Given the variability of the cycle times the system was assumed to have approached steady state once the exponentially smoothed cycle time was within a 10% band of the average cycle time.

### 7.2.2 Configuration

A configuration resulting in a high variability of the performance indicators would test the systems ability to cope with even more unreliability best. However, to enable a fair comparison, the system's utilizations had to be nearly identical. These requests resulted in the configurations given in Table 7-1.

Table 7-1: The configurations chosen for the investigation on the dynamics of failure.

| System | Batch Size | Setup Time | Number of Cards | Interarrival Time | Utilization |
|---|---|---|---|---|---|
| Kanban | 5 | 200 | 15 | - | 0.6553 |
| CONWIP | 5 | 200 | 11 | - | 0.6504 |
| MRP | 5 | 200 | - | 456 | 0.6561 |

The following section covers observations for the time an entity spends at different points in the system.

### 7.2.3 Time Spent in the System

The failure duration of one hour was chosen to possibly force the system to empty the buffers of the machines downstream from the failed machine. A batch of five entities would need on average 100 seconds to pass through one machine. For the Kanban system at most eight batches would be present downstream during the occurance of failure, resulting in 800 seconds. Adding the setup time of 200 seconds per batch times eight batches to the 500 seconds would equal to 2,100 seconds on average. As the process times and setup times are exponentially distributed, this sum may easily double leaving a few entities in the system unlikely to obstruct the batches that had to wait in the buffer of machine 5 for its reactivation.

Figure 7-4 shows 20 replications for the first entity passing through the downstream half of the line after the reactivation of machine 5 for Kanban. The average of the time after failure for passing point six for the 20 replications is 3,851.96 seconds, with a standard deviation of 159.66 seconds. For replication 16 the time after failure is minimal with a length of 3,616.73 seconds [see Table 7-2]. It thus takes the first entity 3,616.73 – 3,600 = 16.73 seconds after failure to depart machine 5. Either this entity was worked on before the failure occurred and it took 16.73 seconds to complete processing or the process time was shorter than the mean of 20 seconds. The maximum is 4,446.35 seconds for replication 10. This entity was delayed 846 seconds at machine five, a truly long period of time, which can most probably be associated to the random number given by the exponential distribution. As the entity continues travelling through the downstream half of the line, the coefficient of variation increases from 0.05 to 0.10. This can easily be seen in the graph comparing the local maxima and minima and their difference. For passing point 11 or the system departure, this difference equals to 1,889.25 seconds or half an hour.

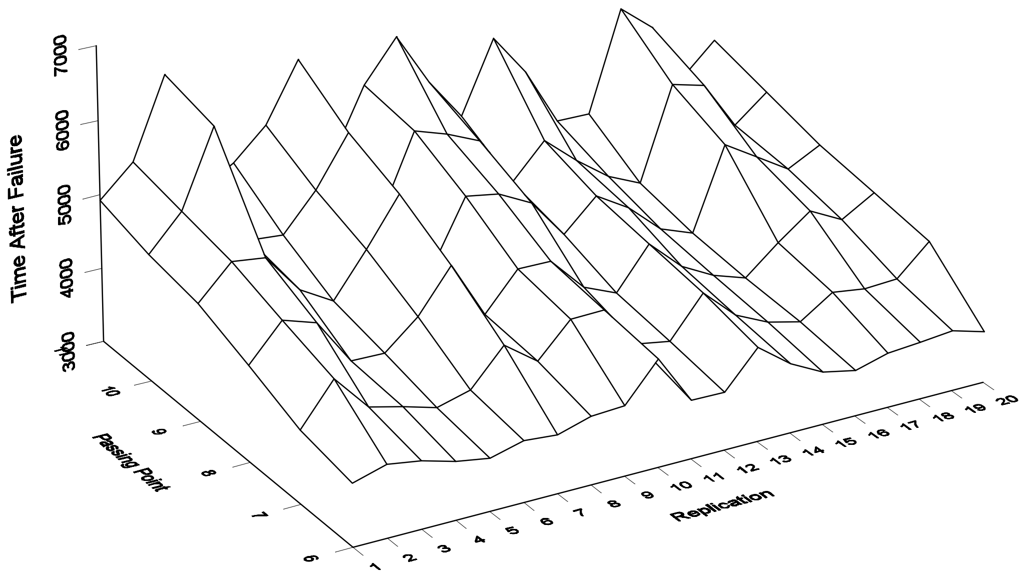


Figure 7-4: 20 replications showing the first entity passing through the downstream half of the line after the reactivation of machine 5 for Kanban.

CONWIP shows similar patterns [see Figure 7-5]. Replication 11 reveals an extremely early departure time from machine 5. Only 3,604.558 seconds after the failure of the machine, the entity passes through passing point 6, indicating a process time of 4.558 seconds [see Table 7-2]. This short time may be due to processing of the entity before the failure occurred. However, the value could have been generated by the random distribution as well.

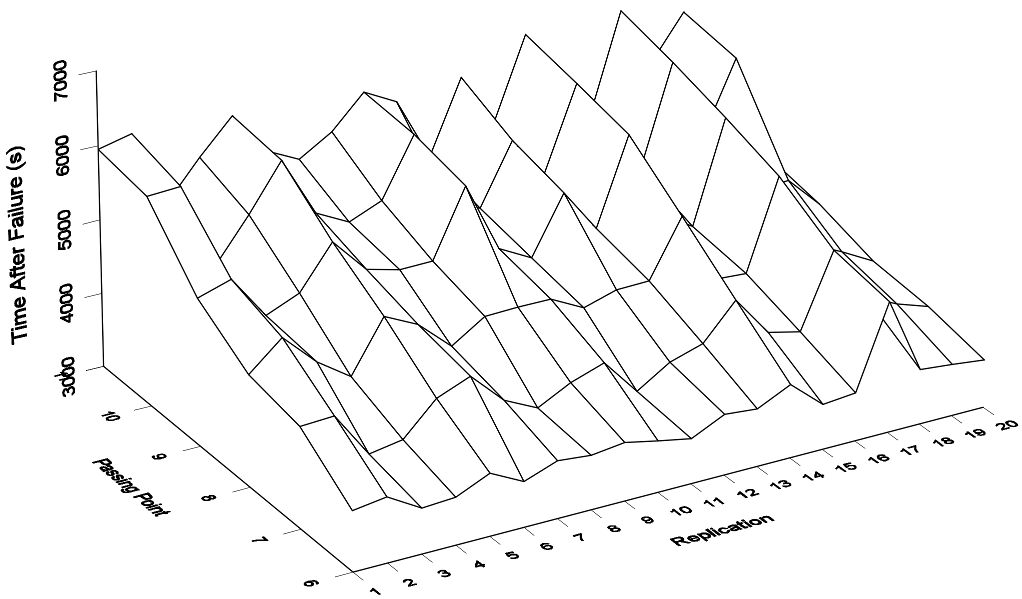


Figure 7-5: 20 replications showing the first entity passing through the downstream half of the line after reactivation of machine 5 for CONWIP.

As usual, MRP shows a somewhat different pattern [see Figure 7-6]. The variation increases tremendously resulting in a maximum of 7,624.563 seconds, which is about 1,200 seconds higher than Kanban's maximum [see Table 7-2].

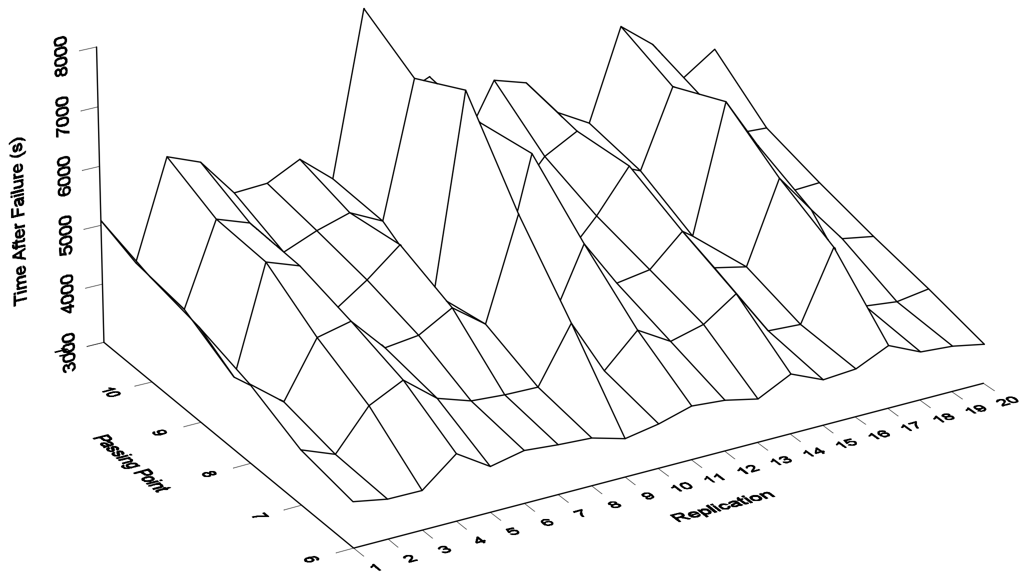


Figure 7-6: 20 replications showing the first entity passing through the downstream half of the line after reactivation of machine 5 for MRP.

The coefficients of variation of the time after failure for MRP are clearly higher than those for Kanban of CONWIP [see Table 7-2].

Table 7-2: The coefficients of variation, the minimal and the maximal times after the failure of machine 5 for Kanban, CONWIP, and MRP.

| Passing Point | $\tilde{c}_{Kanban}$ | $\tilde{c}_{CONWIP}$ | $\tilde{c}_{MRP}$ | $t^{min}_{Kanban}$ | $t^{min}_{CONWIP}$ | $t^{min}_{MRP}$ | $t^{max}_{Kanban}$ | $t^{max}_{CONWIP}$ | $t^{max}_{MRP}$ |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 0.0477 | 0.0611 | 0.0399 | 3616.07 | 3604.56 | 3639.90 | 4446.35 | 4761.03 | 4276.69 |
| 7 | 0.0672 | 0.0670 | 0.0890 | 3732.42 | 3694.91 | 3704.37 | 4652.95 | 4931.82 | 5226.18 |
| 8 | 0.0732 | 0.0732 | 0.1041 | 3799.85 | 3936.57 | 3828.04 | 4982.78 | 5414.80 | 5847.88 |
| 9 | 0.0765 | 0.0826 | 0.1091 | 3890.76 | 4210.94 | 3942.76 | 5396.01 | 5790.14 | 6255.38 |
| 10 | 0.0967 | 0.0961 | 0.1150 | 4065.97 | 4385.84 | 4097.58 | 6217.88 | 6359.00 | 7060.23 |
| 11 | 0.0959 | 0.0928 | 0.1191 | 4521.75 | 4474.03 | 4224.08 | 6411.00 | 6786.05 | 7624.56 |

Figure 7-7 gives a comparative view of the three control systems and their average times after failure. The average was calculated for 30 replications ensuring a half-width of less than 10% of the mean [see 8.4.1 Dynamics of Failure, Time Spent in the System]. According to the data points in the graph, Kanban outperforms the other two systems at nearly every passing point. As the variability of the times increases, Kanban maintains its superior performance. This can be explained by pointing to the fact that MRP and CONWIP continue sending units into the system during repair whereas Kanban terminates the entry of new batches into the system.

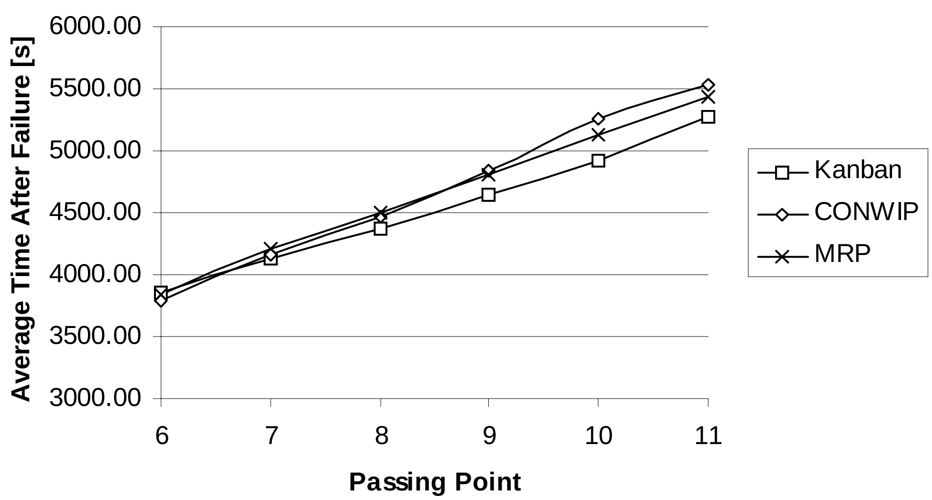


Figure 7-7: The average time after failure at the passing points for Kanban, CONWIP, and MRP.

The paired t-test for the time after failure at the passing point 11 indicates, that there is no significant difference between CONWIP and MRP [see Table 7-3]. Even for Kanban and CONWIP or MRP the confidence interval of the two sided test includes zero, indicating that there is no significant difference. The p-values support this finding.

Table 7-3: The results of the paired t-test for the time after failure at passing point 11 or departure of the system for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -1.7553 | 30 | 0.0894 | (-524.2984; 39.6207) | -242.3388 |
| CONWIP-MRP | 0.5156 | 30 | 0.6099 | (-232.3546; 389.2861) | 78.4657 |
| Kanban-MRP | -1.5290 | 30 | 0.1368 | (-382.7632; 55.0169) | -163.8731 |

Thus, it can not be stated with a high confidence that Kanban outperforms the other two systems. More replications would have to be run to narrow the confidence intervals down. It is very likely, that zero will drop out of the interval for a comparison between Kanban and CONWIP or MRP, as the given intervals are shifted to the left around the negative mean values.

At first the individual entity was observed as it passed through the system. The performance of the control system was judged by the time taken for the first entity to depart the line after reactivation of the failed machine. A second investigation looks at the system with a higher degree of abstraction. The performance will be rated according to the time the system takes to recover from the failure.

### 7.2.4 System Recovery

As described earlier, the exponentially smoothed time taken to depart the line is calculated and scaled with the time taken for the smoothed time to get within a 10% range of the average cycle time. Figure 7-8 illustrates the exponentially smoothed cycle time versus the time after the failure occurred for 15 replications, where Kanban, CONWIP, and MRP are represented by five replications each.

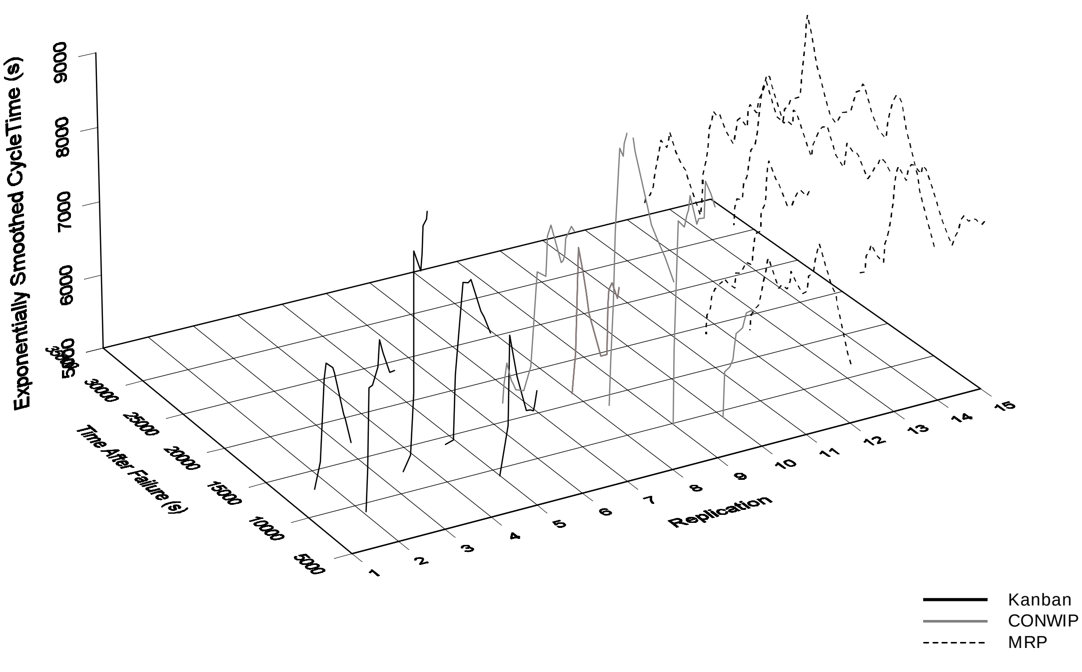


Figure 7-8: The moving average of the cycle time dependent on the time after failure for five replications per Kanban, CONWIP, and MRP.

A quick look reveals that it takes MRP very much longer to approach the average cycle time than it takes the pull systems. A closer look unveils that the curves have different shapes, too. With Kanban and CONWIP controlling the manufacturing line, a negative quadratic shape can be seen. The first entity can pass through the downstream half of the line fairly quickly without much obstruction. The second entity has to wait until the first one is processed resulting in an increase of its cycle time. Thus, the exponentially smoothed time increases incorporating the increased cycle times. Depending on the exponentially distributed process times, another peak may occur in the lines. As the amount of entities trapped in the system is limited to the number of cards assigned to the upstream half of the line, a decrease of the moving average can soon be expected. The decrease is caused by the entities entering the system after the failure and passing through unobstructedly.

With MRP the situation looks somewhat different. As the entities are constantly introduced to the system, independent of failure, a fairly big amount of material builds up during failure. A large amount of entities with high cycle times resides in the system waiting to be processed. These entities have similar cycle times only slightly differing in their process history. The sum of process times and setup times adds up to about half of the average cycle time, about 2,500 seconds. The repair duration of 3,600 seconds is added to this amount making up more than half of the total. Consequently, several long cycle times influence the moving average resulting in several peaks. The peaks occur randomly and not necessarily soon after the reactivation of the failed machine [see Figure 7-8, replications 10 to 15]. A quadratic structure can not be discerned, rather a steady decrease.

A regression analysis was performed to support the findings. The resulting multiple linear regression models are shown in Table 7-4.

Table 7-4: Output for the multiple linear regression models fitting the moving average cycle time dependent on the time after failure for Kanban, CONWIP, and MRP.

| Model | | | | | | |
|---|---|---|---|---|---|---|
| $\bar{t}_{system}^{move} = \beta_0 + \beta_1 t + \beta_{i+1} I_i + \beta_{i+k} I_i t + \beta_{2(k+1)} t^2 + \beta_{i+2(k+1)} I_i t^2$ | | | | | | |
| System | $i$ | $k$ | $R^2$ | $\beta$ intercept | $\beta$ linear $t$ | $\beta$ quadratic $t$ |
| Kanban | 1, 2, …, 29 | 29 | 0.826 | (-32425.30; 15218.36) | (-3.850; 9.571) | (-0.007; 0.006) |
| CONWIP | 1, 2, …, 29 | 29 | 0.819 | (-25573.86; 227.67) | (-0.039; 6.525) | (-0.004; 0) |
| MRP | 1, 2, …, 34 | 34 | 0.881 | (-4659.49; 8471.05) | (-0.479; 0.690) | 0 |

Multiple regression models were chosen to compensate for the differences in intercept, slope and curvature per replication. A simple model would have required an enormous amount of data to describe the response of the system to failure as the variability is quite big. The individual intercepts per replication reflect the time of departure for the first entity after failure. These times were found to be quite variable. The same behavior was found for the curvature. The p-values for the t-test calculating the significance of the individual regressors range from very low to very high, indicating that a difference of intercept, slope, and curvature could not always be found between the individual replications. However, the multiple coefficients of determination are fairly high, which can not be ascribed to the high number of observations, indicating a good fit of the model.

Interpreting the output data in Table 7-4, the following can be stated:

1. Kanban has a higher variability than CONWIP judged by the coefficient intervals. The variability is not necessarily comparable with MRP, as MRP produced a much higher amount of data points due to the long time of recovery.

2. The gradients indicate, that Kanban recovers faster than CONWIP, which in turn recovers faster than MRP.
3. For MRP no curvature can be measured, due to the frequent peaks or local maximums in the moving average time line [see Figure 7-8]. Kanban even shows a positive quadratic response, indicating that the peak is reached early and the moving average decreases faster at the beginning and slower as it approaches the average cycle time.

These findings were verified with the calculated means of the time after failure when the exponentially smoothed average of the cycle times exceeded the average cycle time by less than 10% [see Figure 7-9].

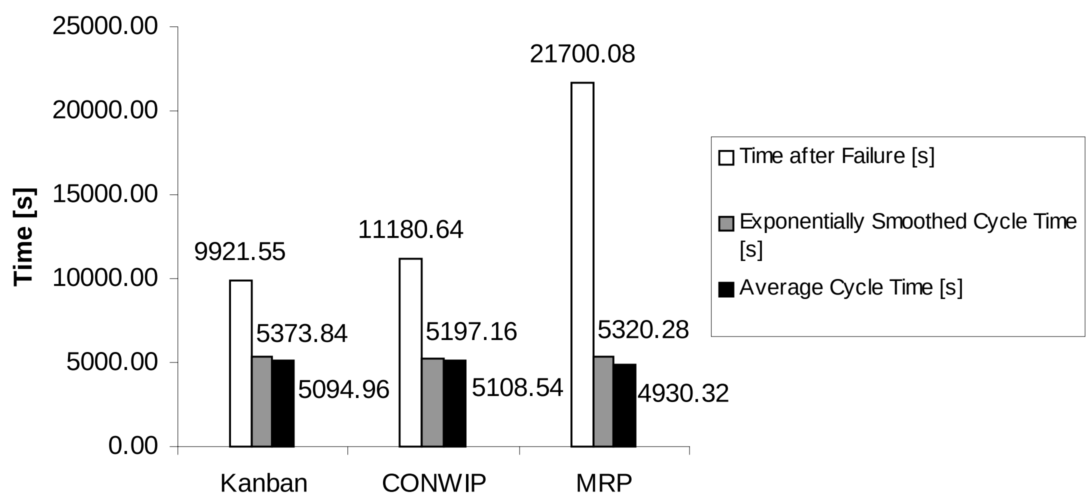


Figure 7-9: The time after failure for which the exponentially smoothed average of the cycle times exceeds the average cycle time by less than 10% for Kanban, CONWIP, and MRP.

To calculate the means, 20 replications sufficed for the pull systems. For MRP 40 replications were needed to meet the 10% criterion for the half-width of the confidence intervals [see 8.4.1 Dynamics of Failure, Recovery Time]. Figure 7-9 clearly illustrates the superiority of the pull systems over the push system. As expected, Kanban performs best, taking on average 9,921.55 seconds after failure to enter the 10% band width of the average cycle time. As moving average of cycle time is the biggest of the three control systems, 5,373.84 seconds, Kanban naturally takes a shorter time to meet the stop criterion. This may be the main reason, why CONWIP is clearly outperformed, needing 11,180.64 seconds to come within 10% of its average cycle time. CONWIP has the shortest moving average, which is almost 200 seconds shorter than the shortest time for Kanban. MRP shows an easily noticeable difference in time after failure from its pendants. It takes nearly twice as long as the pull systems to approach the average cycle time. This can be ascribed to the enormous buildup of WIP during failure. Similarly, CONWIP builds up more WIP than Kanban, as the cards are not bound to specific machines, accumulating entities in front of the failed machine. Theoretically, Kanban keeps only about half of the total WIP in the upstream half of the line, being able to quickly reduce the moving average with short cycle times produced by entities entering the system after failure.

### 7.2.5 Conclusions

The following conclusions can be made from the investigations above:

1. Looking at the time spent in the system the ranking,

    1. Kanban,

    2. MRP, and

    3. CONWIP,

can be derived. However, the ranking can not be stated with high confidence for the number of replications completed, as the differences in the time spent in the system are not highly significant.

2. A closer look at the time needed for recovery after failure reveals the following order:

   1. Kanban,
   2. CONWIP, and
   3. MRP.

   This order can be ascribed to the amount of WIP accumulating in front of the failed machine increasing the cumulative cycle time. This amount is the largest for MRP.

After investigating the dynamic response of the manufacturing systems to failure, the influence of machine failure on the performance of a system in steady-state is observed next.

### 7.3 Failure in Steady-State

In comparison to the simulations carried out to observe the dynamic response where failure was induced at one machine only, in this section all the machines were caused to fail at exponentially distributed time instances lasting for exponentially distributed time durations. A large number of entities was processed to reach a steady-state for which the indicators could be estimated fairly accurately on a 95% confidence level [see Table 8-16].

The previous models were altered to represent more realistic systems with machine failure, including the old set of parameters as well as the additional parameters introduced above [see 7.1].

### 7.3.1 Parameters

The parameters

- Process time,
- Number of cards for the pull systems,
- The interarrival time for MRP,
- Setup ratio, and
- Batch size

were kept from the previous models constructed for experiments including setup time [see 6.1]. The simulations were performed with the same levels. Additionally,

- The interfailure time and
- The repair duration

were introduced.

Realistic machine availability had to be determined to ensure the practical applicability of the performed simulations. A range of availability was established, that would keep the machine utilization fairly high. As the card configurations chosen for setup were kept for the current experiments, very low availability would reduce the machine utilizations to unacceptable low values.

However, the availability had to be represented by specific interfailure times and repair durations. These parameters should represent different scenarios to consider minor failures, which can be resolved within minutes, and major failures, that may take several hours to repair. Thus, short interfailure times with short repair durations as well as long interfailure times with longer repair durations should be taken into account. Unfortunately, the influence of the length of failure and its frequency on the system's performance was not well understood. These influences were studied first.

### 7.3.2 Influence of Interfailure Time and Repair Duration

For the investigations on the influence of interfailure time and repair duration a set of experiments was performed using CONWIP. At that point in time, CONWIP seemed to unite characteristics from both Kanban and MRP.

As the influence of the availability on the average utilization was unknown, a fairly low availability was chosen:

$$a = \frac{7200}{7200 + 600} = 0.923,$$

where the interfailure time equals 7,200 seconds or two hours and the repair duration 600 seconds or 10 minutes, thus, both values being easily divisible by 60. The availability was kept constant and the interfailure time and the repair duration were increased resulting in the combinations given in Table 7-5.

Table 7-5: Combinations of interfailure time and repair duration resulting in a constant availability.

| Configuration | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Interfailure Time | 7200 | 8640 | 10080 | 11520 | 12960 | 14400 | 15840 | 17280 | 18720 | 20160 | 21600 |
| Repair Duration | 600 | 720 | 840 | 960 | 1080 | 1200 | 1320 | 1440 | 1560 | 1680 | 1800 |

Furthermore, the simulations were run for low and high utilization levels, keeping the amount of cards assigned constant. The setup time was kept constant at 2 seconds, not to distort the output data even more. The batch size, $b$, was altered, $b = 1, 2, \ldots,$ 10. This resulted in

$$n = n_a n_u n_b = 11(2)(10) = 220$$

simulations, where $n_a$ is the number of combinations of interfailure time and repair duration, $n_u$ is the number of utilization levels, high and low, and $n_b$ is the number of different batch sizes chosen. The batch size was altered rather than running several replications with a fixed size, as its influence on the performance was not known and the results were supposed to reveal information generally applicable. The average utilizations were then averaged for a given availability combination and utilization level:

$$\bar{u}_{il}^{avg} = \sum_{j=1}^{10} \bar{u}_j ,$$

for the $i$th configuration, $i = 1, 2, \ldots, 11$, and the $l$th utilization level, $l$ = low, high, where $\bar{u}_j$ stands for the $j$th average utilization per batch size and replication. The resulting data is illustrated in Figure 7-10.

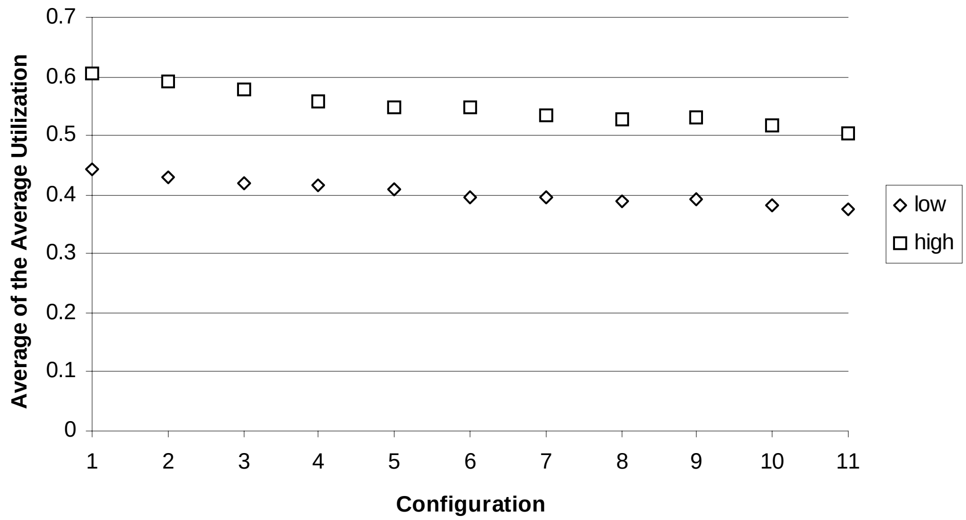


Figure 7-10: The average of the average utilizations per batch size and replication versus the configuration for increasing interfailure times and repair durations.

The graph clearly illustrates the decrease in utilization with increasing interfailure times and repair durations. To observe the availability closely, the state of the machines was determined every twenty seconds to calculate the average availability. The average availability throughout the entire simulation time was given as an output at the end. This output proved the availability to be almost constant across the different configurations.

The decrease in utilization has a logical explanation. The downstream buffers must be maintained at all times to provide protection against the loss in throughput. However, long repair durations starve the machines downstream for a considerable amount of time resulting in empty buffers and possibly a production stop. It takes quite some time to replenish this WIP. Shorter failures can rather be viewed as long process or setup times, momentarily creating a vacuum of WIP at certain places in the manufacturing line. The downstream buffers may still be able to provide the starved part of the line with enough material to continue processing. Hopp and Spearman discuss a one machine line example revealing similar behavior [see HOP96, p. 263].

### 7.3.3 Conclusions for Interfailure Time and Repair Duration

As the interfailure times and the repair durations increase, keeping the availability constant, the average utilization decreases. Thus, it seems to be of advantage to implement preventive maintenance done frequently with a shorter duration. Waiting for the failure to occur resulting in bigger damage and longer repair times can reduce the performance of a system.

As the length of the interfailure time, the length of the repair duration, and the degree of machine availability influence the performance of the system, two experiments were conducted:

1. the interfailure time was increased while the repair duration was held constant, and
2. the interfailure time was held constant while the repair duration was decreased

to reach certain degrees of availability. The resulting scenarios are shown in Table 7-6.

Table 7-6: The interfailure times and repair durations in different time units representing the scenarios for the range of availability simulated.

| Scenario | $t_{infail}$ [s] | $t_{infail}$ [h] | $t_{infail}$ [d] | $t_{repair}$ [s] | $t_{repair}$ [min] | Availability |
|---|---|---|---|---|---|---|
| 1 | 28200 | 7.83 | 0.33 | 1800 | 30 | 0.94 |
| 2 | 34200 | 9.50 | 0.40 | 1800 | 30 | 0.95 |
| 3 | 43200 | 12.00 | 0.50 | 1800 | 30 | 0.96 |
| 4 | 58200 | 16.17 | 0.67 | 1800 | 30 | 0.97 |
| 5 | 88200 | 24.50 | 1.02 | 1800 | 30 | 0.98 |
| 6 | 178200 | 49.50 | 2.06 | 1800 | 30 | 0.99 |
| 7 | 86400 | 24.00 | 1.00 | 5515 | 91.91 | 0.94 |
| 8 | 86400 | 24.00 | 1.00 | 4547 | 75.79 | 0.95 |
| 9 | 86400 | 24.00 | 1.00 | 3600 | 60 | 0.96 |
| 10 | 86400 | 24.00 | 1.00 | 2672 | 44.54 | 0.97 |
| 11 | 86400 | 24.00 | 1.00 | 1763 | 29.39 | 0.98 |
| 12 | 86400 | 24.00 | 1.00 | 873 | 14.55 | 0.99 |

Thus, the availability was kept within the interval [0.94; 0.99]. The repair time makes up at most 6% of total time. Here, this means that for scenario 1 28,200 seconds or 0.33 hours after every reactivation of a machine a failure occurs, which takes 1,800 seconds or 30 minutes to repair [see Table 7-6].

As before, the utilization was kept constant across all the manufacturing lines controlled by the three different control mechanisms.

### 7.3.4 Utilization

The number of cards allocated to the systems, resulting in a high and a low utilization level, was retained throughout the simulation studies. As more variability was introduced to the systems, including setup time and machine failure, the minimum utilization level decreased to an unrealistic level [see Table 7-7]. This was accepted as a consequence of the trade-off between a low variation of the parameter levels and the realism of the systems.

Table 7-7: The minimum, mean, and maximum values for the low and high utilization levels as a summary for the simulations completed, including machine failure for Kanban, CONWIP, and MRP.

| System | Utilization Level | Min | Mean | Max |
|---|---|---|---|---|
| Kanban | low | 0.318 | 0.59 | 0.753 |
| CONWIP | low | 0.31 | 0.60 | 0.741 |
| MRP | low | 0.31 | 0.60 | 0.741 |
| Kanban | high | 0.445 | 0.66 | 0.866 |
| CONWIP | high | 0.46 | 0.68 | 0.872 |
| MRP | high | 0.46 | 0.68 | 0.87 |

However, the response within the utilization interval [0.65; 0.9] was observed to be almost linear, making the results obtained here for the higher utilization level only transformable to utilization levels within an interval. Table 7-8 lists the output for the paired t-test to establish the difference in utilization between the three control systems.

Table 7-8: The output for the paired t-test to establish the difference between the utilization including machine failure for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | | |
|---|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | | |
| Output | | | | | | |
| Comparison between | Level | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | Low | -23.5278 | 1199 | 0 | (-0.00947, -0.00801) | -0.00874 |
| CONWIP-MRP | Low | 18.9 | 1199 | 0 | (0.00035, 0.00043) | 0.00039 |
| Kanban-MRP | Low | -22.3771 | 1199 | 0 | (-0.00908, -0.00762) | -0.00835 |
| Kanban-CONWIP | High | -44.2013 | 1199 | 0 | (-0.02358, -0.02158) | -0.02258 |
| CONWIP-MRP | High | 12.7387 | 1199 | 0 | (0.00033, 0.00045) | 0.00039 |
| Kanban-MRP | High | -43.3191 | 1199 | 0 | (-0.02319, -0.02118) | -0.02219 |

For the high utilization level the difference between Kanban and CONWIP is much greater than for the low utilization level. The relative difference between the control systems Kanban and CONWIP can be calculated with the following formula:

$$\delta\bar{u}_{Kanban}(\bar{u}_{CONWIP}) = \frac{\bar{u}_{CONWIP} - \bar{u}_{Kanban}}{\bar{u}_{Kanban}},$$

where $\bar{u}_{Kanban}$ is the average utilization for Kanban and $\bar{u}_{CONWIP}$ is the average utilization for CONWIP, here $\delta\bar{u}_{Kanban}(\bar{u}_{CONWIP}) = 0.0342$, $\delta\bar{u}_{Kanban}(\bar{u}_{MRP}) = 0.0336$, and $\delta\bar{u}_{CONWIP}(\bar{u}_{MRP}) = 0.00006$.

The small difference between CONWIP and MRP is a result of adjusting MRP's interarrival time to the utilization obtained for CONWIP. The resulting relation

$$\bar{u}_{CONWIP} > \bar{u}_{MRP} > \bar{u}_{Kanban}$$

agrees with the relation obtained for the systems with batch size and setup time included and machine failure excluded [see 6.1.2].

The cycle time is investigated to establish the influence of machine failure on the performance of the control systems. Initially, the average cycle time is discussed. Then, the maximum cycle time and the standard deviation of cycle time are taken into consideration.

### 7.3.5 Average Cycle Time

As the reduction of machine availability decreases the utilization (the throughput) of the manufacturing line, it has an increasing effect on the average cycle time. Figure 7-11 illustrates the response of the average cycle time to the batch size, the setup ratio and the availability level for the high utilization level. Different availability levels form layers of the response surface and the average cycle time increases with decreasing availability. Thus, looking at a vertical data point row in Figure 7-11 the availability decreases from bottom to top. The individual layers resemble the response surfaces shown for the models including batch size and setup time only [see Figure 6-1]. The surfaces lie nearly parallel to one another, the gradients increase slightly for decreasing machine availability. As the batch size and the setup ratio increase, the variance of the average cycle time increases. The data points are spread further apart for higher values of batch size and setup ratio.

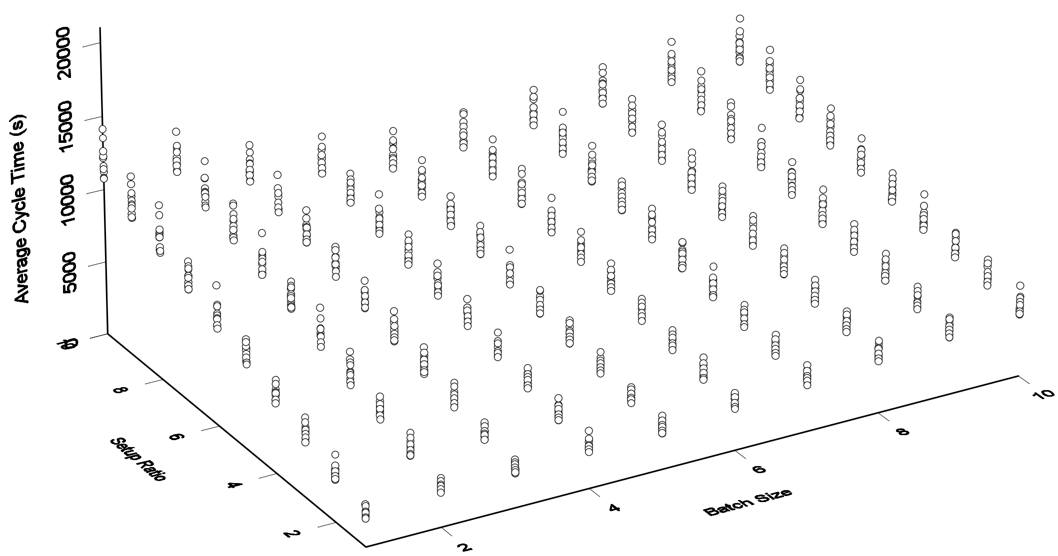


Figure 7-11: The average cycle time versus the batch size and the setup ratio for the six availability levels for Kanban.

Figure 7-12 reveals the response of the average cycle time to the batch size, setup time and availability level for CONWIP. The reason for displaying the graph is to illustrate the difference in variance between Kanban and CONWIP. The data points are spread further apart for CONWIP than for Kanban. Further, observing the average cycle time for a batch size of one and a setup ratio of ten, CONWIP unveils a spread of data points above the 15,000-second mark. This makes an overall increase of average cycle time for CONWIP in comparison to Kanban discernible.

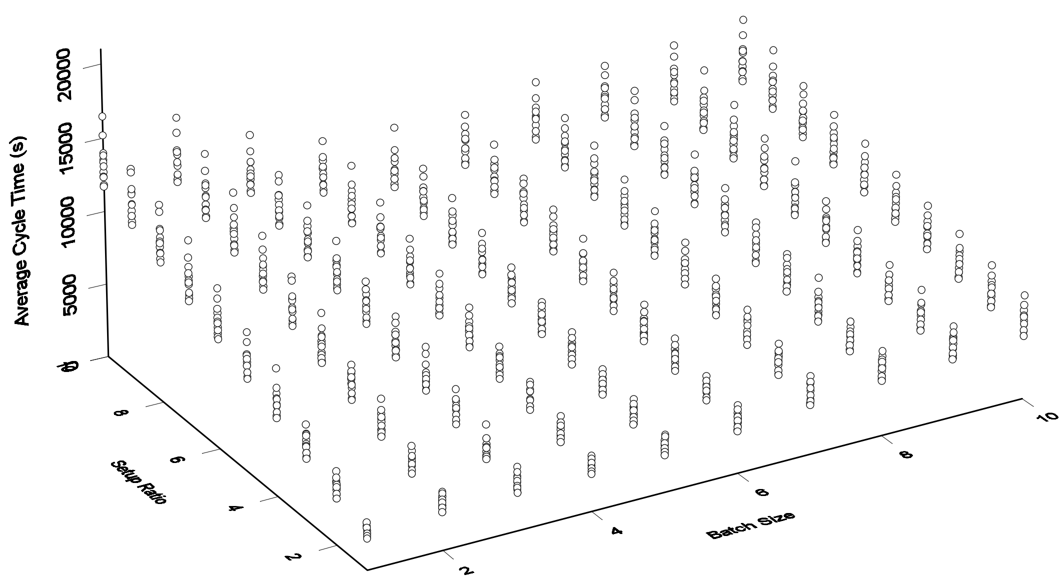


Figure 7-12: The average cycle time versus the batch size and the setup ratio for the six availability levels for CONWIP.

Figure 7-13 shows the response of the average cycle time to the batch size, the setup ratio and the different availability levels for MRP. The increase in variance can easily be seen. Further, the unequal spread of data points for a fixed batch size and setup ratio indicates a non-linear response of the average cycle time to the availability. Drawing an imaginary vertical line through a given combination of batch size and setup ratio reveals unequal distances between the data points on that line. The average cycle time for a batch size of one and a setup ratio of one even exceeds the 20,000 second mark, indicating the inferiority of the pull system, MRP, to the push systems, that stay close to the 15,000 mark for the lowest availability of 0.94.

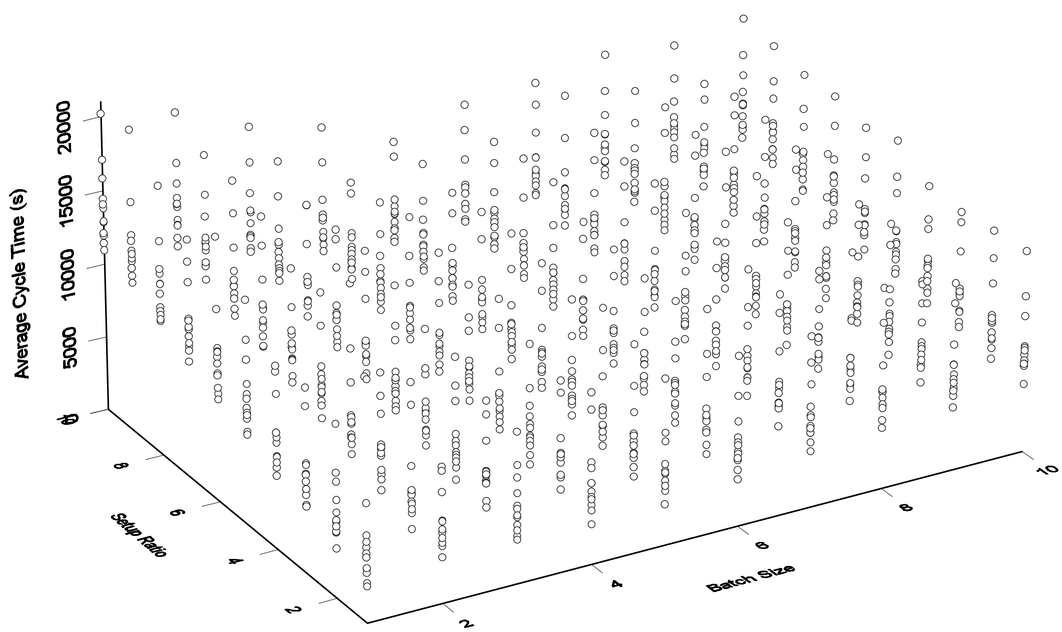


Figure 7-13: The average cycle time versus the batch size and the setup ratio for the six availability levels for MRP.

The three graphs shown above reveal the response of the cycle time to the batch size, setup time and availability level for the high utilization level that results from higher card assignments to the line for the pull systems and from a lower interarrival time for the push system, MRP. The low utilization level is not investigated in detail, as its mean below 0.65 makes the results hardly applicable to reality. For the low utilization level, the response surfaces resemble the surface obtained for setup and batch size experiments [see Figure 6-3]. The variance of the data points for given combinations of batch size and setup ratio was found to be less than the variance obtained for the higher utilization level. Henceforth, only the output data for the high utilization level will be discussed.

Table 7-9 provides a short summary of statistics on the average cycle time for Kanban, CONWIP, and MRP.

Table 7-9: A summary of statistics on the average cycle time for Kanban, CONWIP, and MRP including machine failure.

| Control System | Minimum | 1. Quartile | Mean | Median | 3. Quartile | Maximum |
|---|---|---|---|---|---|---|
| Kanban | 1888.96 | 4681.02 | 6298.4 | 6185.48 | 7768.58 | 14116.33 |
| CONWIP | 2101.54 | 5129.98 | 6959.04 | 6762.1 | 8513.11 | 16456.66 |
| MRP | 2457.91 | 6292.81 | 8469.86 | 7959.91 | 10107.02 | 20555.58 |

The output for the paired t-test performed to establish the difference between the means of the average cycle times for the three control systems given in Table 7-10 supports the data given above [see Table 7-9].

Table 7-10: The output for the paired t-test to establish the difference between the average cycle time including machine failure for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -55.001 | 1199 | 0 | (-684.2086 ,-637.0774) | -660.643 |
| CONWIP-MRP | -30.057 | 1199 | 0 | (-1609.433, -1412.199) | -1510.816 |
| Kanban-MRP | -39.698 | 1199 | 0 | (-2278.777, -2064.141) | -2171.459 |

The 95% confidence intervals for the difference between Kanban and CONWIP and for the difference between Kanban and MRP are narrow, their half-width is below 10% of the estimate of the mean of difference, indicating an accurate estimation of the true difference.

The results are not surprising at all and can be explained by looking at the WIP in the line. For Kanban the material is assigned to certain machines with the number of cards as an upper bound. During a machine failure, the material downstream the failed machine is pulled out of the system, while the material trapped upstream the failed machine is limited to a fixed amount. The total number of cards assigned to the line provides an upper limit to the WIP trapped in the system. For CONWIP, the material accumulates in front of the failed machine, as the material pulled out of the system is replaced by new releases at the beginning of the line [see 2.3.1]. The amount trapped upstream the failed machine may amount to the total number of cards assigned to the line, provided the repair duration is long enough for the entities downstream the failed machine to pass out of the system to be replaced by new releases. For MRP no upper limit exists, the releases are controlled by the interarrival rate of raw material to the line. Independent of a machine failure, new orders are released at the beginning of the line. During a machine failure the material is trapped in front of the failed machine for the entire repair duration. The material downstream the failed machine passes through the line, however, depending on the repair duration, the amount trapped upstream may result in a high WIP level for the entire line.

Thus, the amount of material trapped in the system during failure is the smallest for Kanban, followed by CONWIP. For MRP, the WIP level is the highest, primarily limited due to the fact that the utilization level is synchronized for all three systems. As the number of entities trapped in the system during failure increases, the influence of their cycle time on the average cycle time increases. This results in the highest average cycle time for the biggest average amount of WIP trapped during failure.

### 7.3.6 Conclusions for Average Cycle Time

The following relation summarizes the information obtained from the graphs and tables above:

$$\bar{t}_{cycle}^{MRP} > \bar{t}_{cycle}^{CONWIP} > \bar{t}_{cycle}^{Kanban}.$$

Assuming equal utilization for the three control systems, Kanban clearly outranks its pendants. However, the relation

$$\bar{u}_{CONWIP} > \bar{u}_{MRP} > \bar{u}_{Kanban}$$

indicates, that the line controlled by Kanban had a smaller utilization and thus a smaller throughput than the lines controlled by the other two systems. A two percent difference may well influence the performance. However, assuming an almost linear response of the average cycle time to utilization may result in a difference of about

300 seconds, a difference of 660 [see Table 7-10] is found to be most unlikely. MRP shows a definite inferiority to the pull systems, responding with an average cycle time of more than 1,500 seconds.

The resulting ranking illustrates the superiority of Kanban over the other systems, even though the difference between Kanban and CONWIP is not as obvious as that between the pull system and the push systems:

1. Kanban,
2. CONWIP, and
3. MRP.

The average cycle time represents a good indicator for the performance of a manufacturing system. Other indicators should be considered as well to obtain a better understanding for a mechanism's characteristics. The maximum of the cycle time and the standard deviation of cycle time give more insight into a system's behavior.

### 7.3.7 The Maximum Cycle Time

A low average cycle time is an indication for a good performance of a control system. However, a low average may result from a majority of short cycle times and a minority of very long cycle times. Assuming that an important order is trapped in the manufacturing system resulting in a very long cycle time with a first come first serve scheduling policy, the due date can not be made. As a consequence, the customer may be lost. Thus, knowledge on the maximum cycle time may prevent such unfortunate situations. Table 7-11 summarizes statistics on the maximum cycle times for the simulations completed.

Table 7-11: Statistics on the maximum cycle time including machine failure for Kanban, CONWIP, and MRP.

| Control System | Minimum | 1. Quartile | Mean | Median | 3. Quartile | Maximum |
|---|---|---|---|---|---|---|
| Kanban | 5865.9 | 17880.6 | 27091.8 | 23073.8 | 31693.5 | 88870.4 |
| CONWIP | 5939.3 | 19132.7 | 28202.8 | 24698.4 | 33690.3 | 98645.7 |
| MRP | 8928.2 | 24941.4 | 36361.3 | 30514.8 | 42545.1 | 116682.2 |

The box plots in Figure 7-14 illustrate the data from Table 7-11. The small difference between the first quartile and the median and the large distance between the median and the third quartile indicate a concentration of data below the median. There are no outliers below the first quartile. On the other hand, the maximum values for the maximum cycle times can be classified as outliers indicating a minority of data points with extremely large values. The interquartile range for the pull systems is far less than the range for MRP. The small difference between Kanban and CONWIP does not have a notable effect on the customer satisfaction concerning on time delivery.

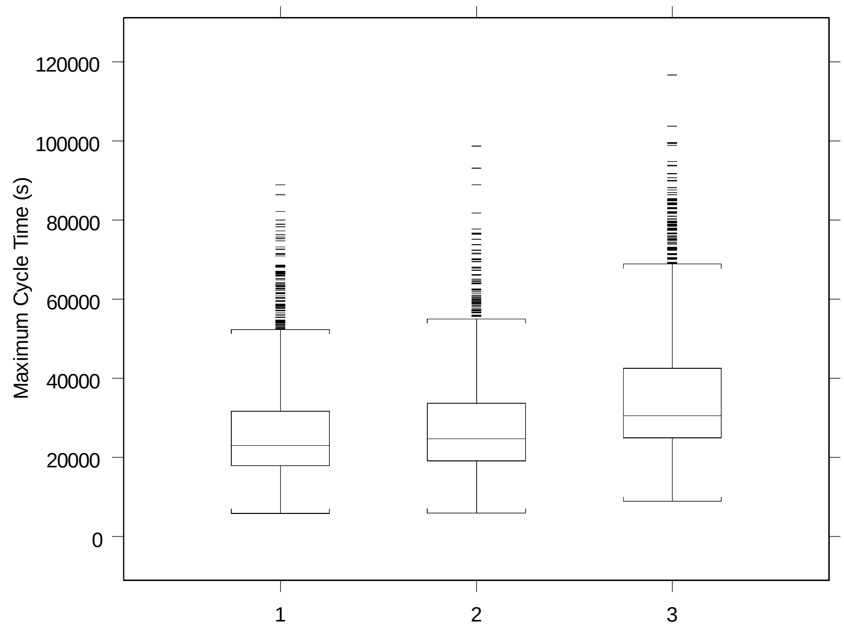


Figure 7-14: Box plots of the maximum cycle time including machine failure for Kanban (1), CONWIP (2), and MRP (3).

The large values for the maximum cycle time result from a low availability and unfortunate variability in the manufacturing system. The maximum cycle time decreases with increasing availability. The decrease is most predominant for MRP. The paired t-test performed on the maximum cycle times for the 1,200 configurations simulated reveals a significant (on a 95% confidence level) difference between the manufacturing lines controlled by the three mechanisms [see Table 7-12].

Table 7-12: The output for the paired t-test on the maximum cycle time including machine failure for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -5.4447 | 1199 | 0 | (-1511.4478, -710.7101) | -1111.079 |
| CONWIP-MRP | -33.2093 | 1199 | 0 | (-8640.428, -7676.457) | -8158.442 |
| Kanban-MRP | -38.3098 | 1199 | 0 | (-9744.237, -8794.805) | -9269.521 |

Although the p-value of zero indicates a significant (95% confidence level) difference between the maximum cycle times for Kanban and CONWIP, the small t-value of –5.447 indicates a lesser significance on a higher confidence level. Further, the difference of 1,111.079 between the two systems looks rather small when set relative to the mean:

$$\delta t_{cycle,\,Kanban}^{max}\left(t_{cycle,\,CONWIP}^{max}\right)=\frac{t_{cycle,\,CONWIP}^{max}-t_{cycle,\,Kanban}^{max}}{t_{cycle,\,Kanban}^{max}}\Leftrightarrow$$

$$\delta t_{cycle}^{max}\left(t_{cycle,\,CONWIP}^{max}\right)=\frac{28202.8-27091.8}{27091.8}\Leftrightarrow$$

$$\delta t_{cycle}^{max}\left(t_{cycle,\,CONWIP}^{max}\right)=0.041,$$

a relative improvement of less than 5%. The difference between the pull systems and the push system would certainly influence the business of a manufacturing company, considering the relative increase of the maximum cycle time of

$\delta t_{cycle,\,Kanban}^{max}\left(t_{cycle,\,MRP}^{max}\right)=0.342$ and

$\delta t_{cycle,\,CONWIP}^{max}\left(t_{cycle,\,MRP}^{max}\right)=0.289$ .

The result is not surprising. In an MRP controlled manufacturing line a large amount of material is trapped upstream the failed machine [see 7.3.5, p. 160]. The last entity in the queue in front of the failed machine has to wait until all the other entities have been processed. The waiting time increases with an increasing amount of material waiting in the queue. As the queue is expected to be longer for CONWIP than for Kanban, CONWIP results in slightly higher maximum cycle times.

### 7.3.8 Conclusions for the Maximum Cycle Time

Summarizing the information above and assuming an almost linear response of the maximum cycle time to utilization, the following ranking results:

1. Kanban, closely followed by
2. CONWIP, and
3. MRP.

The last aspect of cycle time investigated is the standard deviation of cycle time for the number of entities processed per configuration.

### 7.3.9 Standard Deviation of Cycle Times

Both the average cycle time and the maximum cycle time indicate the performance of the manufacturing control systems by measuring the central tendency of the data distribution. They do not indicate the variation or spread of data, revealing the dynamic behavior of a system. The standard deviation of cycle time enables the estimation of how the manufacturing process varies resulting in different cycle times for the entities leaving the system. Table 7-13 summarizes the statistics on the standard deviation of cycle time for the 1,200 simulations run per control system.

Table 7-13: Statistics on the standard deviation of cycle time including machine failure for Kanban, CONWIP, and MRP.

| Control System | Minimum | 1. Quartile | Mean | Median | 3. Quartile | Maximum |
|---|---|---|---|---|---|---|
| Kanban | 512.75 | 1510.77 | 2451.14 | 2172.26 | 2930.1 | 6765.17 |
| CONWIP | 461.95 | 1690.82 | 2762.69 | 2422.58 | 3259.18 | 7797.59 |
| MRP | 1155.77 | 3155.76 | 4754.63 | 3837.42 | 5786.49 | 12874.55 |

Figure 7-14 visualizes the behavior of the standard deviation of the cycle time for Kanban (1), CONWIP (2), and MRP (3). The box plots display the most obvious differences between the three control systems.

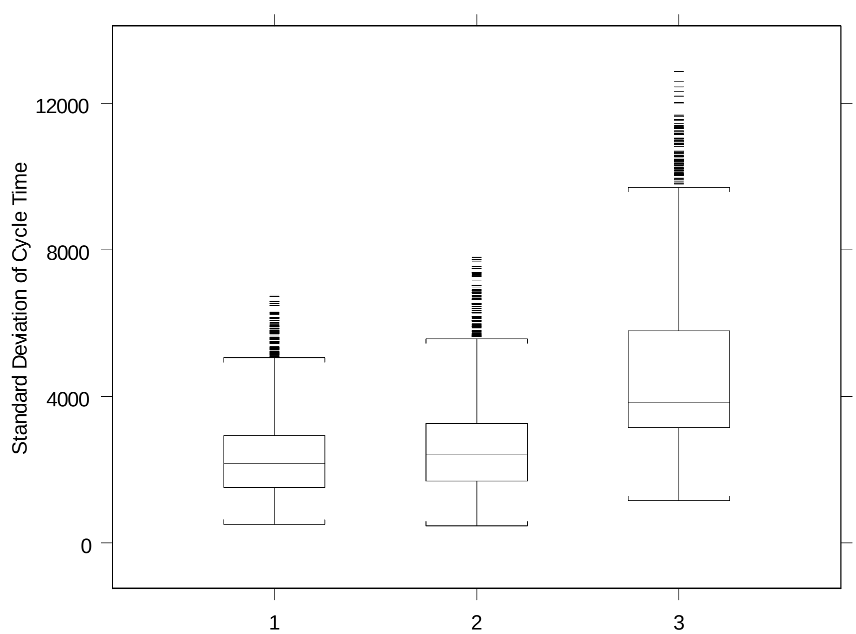


Figure 7-15: Boxplots of the standard deviation of cycle time including machine failure for Kanban (1), CONWIP (2), and MRP (3).

For the pull systems the plots reveal a fairly symmetric distribution of data points around the median between the first and the third quartile. The median for MRP lies in the lower half of the box denoting the first and third quartile. Thus, the lower half of the data is more condensed than the upper half of the data, indicating a few configurations with quite a big standard deviation in cycle times. Further, the data points for MRP lie further apart. Considering the slightly bigger minimum value as well, the process unveils a much higher variability than its pendants. As with the maximum cycle time, the mean of the standard deviation of cycle times for the different configurations lies above the median [see Table 7-13], denoting a few configurations with exceptionally high outputs.

Figure 7-15 does not show the big difference in mean between Kanban and CONWIP. The medians for the two systems are almost the same. Thus, CONWIP produces more extreme outliers increasing the mean. A paired t-test reveals a significant (95% confidence level) difference between the two pull systems [see Table 7-14].

Table 7-14: The output for the paired t-test for the standard deviation of cycle time including machine failure for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | |
|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | |
| Output | | | | | |
| Comparison between | t | df | p-value | 95% Confidence Interval | Mean of diff. estimate |
| Kanban-CONWIP | -30.2428 | 1199 | 0 | (-331.7594, -291.3373) | -311.5483 |
| CONWIP-MRP | -55.745 | 1199 | 0 | (-2062.049, -1921.836) | -1991.943 |
| Kanban-MRP | -56.8838 | 1199 | 0 | (-2382.939, -2224.043) | -2303.491 |

Table 7-15 shows the relative increase of standard deviation of cycle time. The values are high in comparison to the average cycle time and the maximum cycle time.

Table 7-15: The relative increase of the standard deviation in cycle time including machine failure for Kanban, CONWIP, and MRP.

| Indicator | Value |
|---|---|
| $\delta t_{cycle,\,Kanban}^{stdev}\left(t_{cycle,\,CONWIP}^{stdev}\right)$ | 0.127 |
| $\delta t_{cycle,\,Kanban}^{stdev}\left(t_{cycle,\,MRP}^{stdev}\right)$ | 0.721 |
| $\delta t_{cycle,\,\mathrm{CONWIP}}^{stdev}\left(t_{cycle,\,MRP}^{stdev}\right)$ | 0.940 |

The standard deviation of cycle time increases by as much as 94% if MRP is chosen instead of Kanban. Even the increase between the two pull systems reaches 12.7%, a notable difference.

### 7.3.10 Conclusions for the Standard Deviation of Cycle Times

The difference in standard deviations of cycle times between the three control systems is considerable resulting in the following ranking of the mechanisms:

1. Kanban,
2. CONWIP, and
3. MRP.

The initial investigation in the response of the cycle time to varying batch size, setup time and machine failure were done to primarily compare the performance of the three control systems. The regression analysis in the next subchapter provides more information on the individual systems as it reveals quantitative dependencies between the parameters and the indicator average cycle time.

### 7.3.11 Regression

The models were constructed with the aim to enable the calculation of the expected average cycle time with the help of the following parameters:

- Amount of cards assigned for the pull systems,
- Interarrival time for MRP
- Batch size,
- Setup ratio,
- Machine availability,
- Interfailure time, and
- Repair duration.

Either both the availability and the repair duration (for Kanban and CONWIP) or both the availability and the interfailure time (for MRP) were included in the models to take the decrease of utilization for an increase in interfailure time and repair duration, while keeping the availability constant, into consideration [see 7.3.2].

Further, the models were expected to show significant (on a 95% confidence level) regressors and interactions of regressors to provide more insight into the influence of the different parameters on the performance of the control systems. By doing simple arithmetics, the effect of the individual regressors was approximated to establish the most influential parameter or interaction of parameters. A model validation was performed to reveal the correctness of the regression models.

**7.3.11.1 Models**

Table 7-16 lists the regression models for Kanban, CONWIP, and MRP. The coefficients of determination indicate a good fit as all three exceed 0.97. The models were constructed by including regressor terms by trial and error. The best model given by the software's stepwise regression algorithm was found to represent the data points insufficiently.

Table 7-16: The regression models for the average cycle time including machine failure for Kanban, CONWIP, and MRP.

| System | Model | $R^2$ |
|---|---|---|
| Kanban | $\bar{t}_{cycle} = e^x$<br>$x = 6.2202 + 0.245r_s + 0.4542b + 0.0436c + 0.0011t_{repair} -$<br>$0.0011at_{repair} + 0.0049cr_s + 0.0031bc - 0.3139ab - 0.0005abr_s -$<br>$0.0038acr_s - 0.0137r_s^2 - 0.0102b^2 - 0.0007c^2 + 0.0004r_s^3 +$<br>$0.0004b^3 + yc^3$<br>$y < 0.0000$ | 0.9833 |
| CONWIP | $\bar{t}_{cycle} = e^x$<br>$x = 14.5216 - 8.7099a + 0.0002t_{repair} - 0.0002at_{repair} +$<br>$0.2508ar_s + 0.0011bc + 0.1694ab + 0.0582ac - 0.0001bcr_s -$<br>$0.0097abr_s - 0.0065r_s^2 - 0.0038b^2 - 0.0005c^2$ | 0.9919 |
| MRP | $\bar{t}_{cycle} = x^2$<br>$x = 1785.1101 + 0.1648t_{intarr} - 10.4123r_s + 7.4480b +$<br>$0.0103t_{intfail} - 3132.0306a - 0.0103at_{intfail} + 22.7093ar_s +$<br>$0.0048bt_{intarr} - 0.4997at_{intarr} - 0.001br_st_{intarr} + 0.0002t_{intarr}^2 +$<br>$1383.7879a^2 - 0.0494b^2$ | 0.9712 |

The models were assembled from regressors, which were found to be significant on a 99% confidence level. The probability for the t-value being greater than the tabled value were lower than 0.01 for the t-tests performed on the coefficients. The p-value for the F-test was zero for the three models obtained. Naturally, the models given in Table 7-16 are not necessarily optimal, but, they seem to represent the data set quite well.

After transformation, the residuals were found to be independently and normally distributed with a fairly equal variance. The most significant outliers were identified, to analyze the influence of the data points on the model. Figure 7-16 shows the Cook's distance [see 4.3.8] versus the index of the data points, the twenty most notable outliers are labeled.

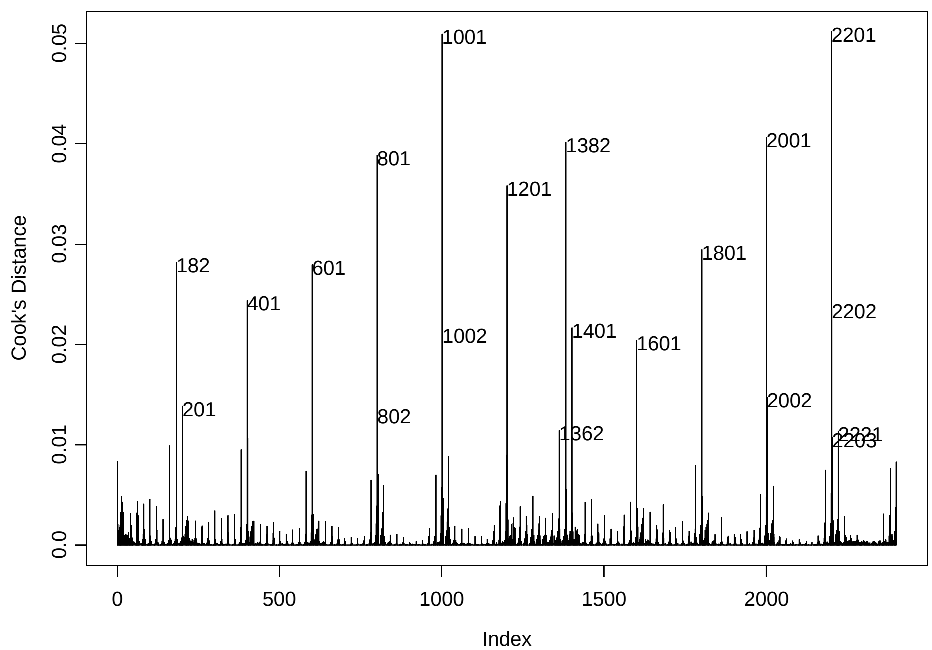


Figure 7-16: The Cook's Distance versus the index of the data points for the regression model for Kanban, including machine failure.

Eleven of the twenty outliers were identified as data points with an index ending on 01. Tracing the data points back to the output data, revealed configurations with batch size one, setup ratio one and the smaller number of cards assigned. These configurations resulted in unusually short average cycle times. However, the throughputs or utilizations were found to be exceptionally small as well, making these configurations not applicable to reality. For the other outliers an obvious pattern could not be distinguished.

#### 7.3.11.2 Effects of the Regressors

A small coefficient does not necessarily indicate, that the corresponding regressor has a small effect on the response variable. The domain of the regressor has to be taken into consideration as well. Suppose regressor A has a coefficient of 1, but represents values in the domain [1,000, 2,000]. On the other hand, regressor B has a coefficient of 1,000, but represents values in the domain [0.1, 0.2]. Multiplying the coefficient with the average of the lower and upper bound of the given domains, results in 1,500 for A and 150 for B. Thus, although the coefficient of B is much higher than the coefficient of A, A has a greater effect on the response variable. For the example, only two levels were taken to calculate the average of the values in the domain. 2,400 replications were used to construct the models. Thus, the averages were calculated from 2,400 values using the following formula:

$$Average_r = \frac{1}{2400} \sum_{i=1}^{2400} \prod_{j=1}^{m_r} l_j \ ,$$

where $l_j$ stands for the level of the *j*th variable in the *r*th regressor term in replication *i*. Table 7-17, Table 7-18, and Table 7-19 list the domain for the regressor terms, the average of the levels for the regressor and the corresponding effects on the average cycle time for Kanban, CONWIP, and MRP, respectively.

Table 7-17: The domain and the corresponding effects for the regressor terms including machine failure for Kanban's regression model.

| Regressor | Domain of Regressor | Average | Coefficient | Effect |
|---|---|---|---|---|
| $t_{repair}$ | [873, 5515] | 2480.833 | 0.0011 | 2.728917 |
| $b$ | [1,10] | 5.5 | 0.4542 | 2.4981 |
| $r_s$ | [1, 10] | 5.5 | 0.245 | 1.3475 |
| $c$ | [13,60] | 20.95 | 0.0436 | 0.91342 |
| $cr_s$ | [13, 600] | 115.225 | 0.0049 | 0.564603 |
| $bc$ | [13, 600] | 98.75 | 0.0031 | 0.306125 |
| $r_s^3$ | [1, 1000] | 302.5 | 0.0004 | 0.121 |
| $b^3$ | [1, 1000] | 302.5 | 0.0004 | 0.121 |
| $abr_s$ | [0.94, 99] | 29.1936 | -0.0005 | -0.0146 |
| $r_s^2$ | [1, 100] | 38.5 | -0.0102 | -0.3927 |
| $c^2$ | [169, 3600] | 596.55 | -0.0007 | -0.41759 |
| $acr_s$ | [12.22, 594] | 111.2066 | -0.0038 | -0.42258 |
| $r_s^2$ | [1, 100] | 38.5 | -0.0137 | -0.52745 |
| $ab$ | [0.94, 9.9] | 5.307979 | -0.3139 | -1.66617 |
| $at_{repair}$ | (820, 5460) | 2380.847 | -0.0011 | -2.61893 |

For Kanban, the repair time and the batch size show a considerable positive effect on the response variable [see Table 7-17]. Considering all the terms including the setup ratio leaves a positive effect on the average cycle time. An increase in the setup ratio results in an increase of the average cycle time. The amount of cards assigned to the entire line is positively influential on the response variable, too. The average cycle time increases with an increasing number of cards allocated to the line. However, the square term of the same variable has a fairly large negative effect on the indicator, neutralizing the positive effect to some extend. The interaction term of the repair duration and the availability have a negative influence on the indicator. This indicates the decrease of cycle time for an increase in availability. The least influence is subjected by the interaction of availability, batch size and setup ratio, which has a low information content in any case.

Table 7-18: The domain and the corresponding effects for the regressor terms including machine failure for CONWIP's regression model.

| Regressor | Domain of Regressor | Average | Coefficient | Effect |
|---|---|---|---|---|
| $ar_s$ | [0.94, 9.9] | 5.307716 | 0.2508 | 1.331175 |
| $ac$ | [0.94, 44) | 16.06885 | 0.0582 | 0.935207 |
| $ab$ | [0.94, 9.9] | 5.307818 | 0.1694 | 0.899144 |
| $t_{repair}$ | [873, 5515] | 2480.833 | 0.0002 | 0.496167 |
| $bc$ | [10, 440] | 80.65 | 0.0011 | 0.088715 |
| $c^2$ | [100, 1936] | 16.65 | -0.0005 | -0.00833 |
| $bcr_s$ | [10, 4400] | 443.575 | -0.0001 | -0.04436 |
| $b^2$ | [1, 100] | 38.5 | -0.0038 | -0.1463 |
| $r_s^2$ | [1, 100] | 38.5 | -0.0065 | -0.25025 |
| $abr_s$ | [0.94, 99] | 29.19294 | -0.0097 | -0.28317 |
| $at_{repair}$ | (820, 5460) | 2380.664 | -0.0002 | -0.47613 |
| $a$ | [0.94, 0.99] | 0.965053 | -8.7099 | -8.40552 |

The most influential parameters for CONWIP are the interaction terms of the availability with the setup ratio, the number of cards, and the batch size [see Table 7-18]. The average cycle time increases with an increasing setup ratio, number of cards assigned, and batch size. The repair duration has a fairly strong positive influence on the average cycle time as well. As with Kanban, an increasing availability decreases the indicator, expressed by the large negative coefficient of the main availability term. The interaction terms including availability are not influential enough to compensate for the strong negative effect of the main availability term.

Table 7-19: The domain and the corresponding effects for the regressor terms including machine failure for MRP's regression model.

| Regressor | Domain of Regressor | Average | Coefficient | Effect |
|---|---|---|---|---|
| $a^2$ | (0.88, 0.98) | 0.964983 | 1383.788 | 1335.332 |
| $t_{intfail}$ | [28200, 178200] | 79050 | 0.0103 | 814.215 |
| $t_{intarr}$ | [48, 927] | 379.0517 | 0.1648 | 62.46771 |
| $b$ | [1, 10] | 5.5 | 7.448 | 40.964 |
| $t_{intarr}^2$ | [2304, 859329] | 167781 | 0.0002 | 33.5562 |
| $ar_s$ | [0.94, 9.9] | 0.964983 | 22.7093 | 21.91409 |
| $bt_{intarr}$ | [48, 9270] | 5.5 | 0.0048 | 0.0264 |
| $b^2$ | [1, 100] | 5.5 | -0.0494 | -0.2717 |
| $br_st_{intarr}$ | [48, 92700] | 14457.09 | -0.001 | -14.4571 |
| $r_s$ | [1, 10] | 5.5 | -10.4123 | -57.2677 |
| $at_{intarr}$ | (45, 918) | 364.9666 | -0.4997 | -182.374 |
| $at_{intfail}$ | [26508, 176418] | 76668.18 | -0.0103 | -789.682 |
| $a$ | [0.94, 0.99] | 0.964983 | -3132.03 | -3022.36 |

As the model for MRP only required square root transformation and not a natural logarithm transformation, the effects are much bigger for MRP's model [see Table 7-19]. As before, the machine availability has a big effect on the average cycle time. The negative influence is stronger than the positive influence of the parameter, taking the main effect, the effect of the square term and the effects of the interaction terms into consideration. MRP is outranked by the pull systems, only when machine failure (availability) is introduced to the models. The regression analysis thus supports the strong influence of this parameter on the performance of the push system. Further, the interfailure time influences the average cycle time positively. The utilization decreases with increasing interfailure time [see 7.3.2], resulting in longer average cycle times. As expected, the interarrival time has a negative effect on the response variable. The average cycle time increases with a decreasing interarrival time.

Looking at the output data immediately reveals the higher average cycle time for the shorter interarrival time. Unexpectedly, the setup ratio shows a negative effect on the indicator. Although, the negative influence is compensated by the interaction term of the setup ratio and the machine availability, the sum of the effects results in an overall negative effect on the average cycle time for the parameter setup ratio. Most importantly, the batch size does not strongly effect the average cycle time. In practice, the batch size is considered an influential factor, which has a linear effect on the performance of a manufacturing system. For MRP, the machine availability seems to influence the performance more significantly, demoting the batch size to a less significant factor.

#### 7.3.11.3 Model Validation

In the previous section, the effect of the regressors on the response variable was compared to the theoretical expectations. All of the expectations with one exception were satisfied. The only unexpected effect was found for the setup ratio in the MRP model [see Table 7-19].

Further, new data was collected to check the models and its predictive ability. Values within the parameter domains [see Table 7-17, Table 7-18, and Table 7-19] produced a good fit of the models. Unfortunately, the above regression models were found to produce unsatisfactory results for predicting average cycle times with parameter levels outside the given intervals. Input values close to the given parameter domains resulted in somewhat representative output data. However, the fit of the predicted values worsened with an increasing distance of the input values from the given domains.

### 7.3.12 Conclusions

The following summarizes the information obtained from constructing regression models for the three control systems:

- The machine availability has a strong negative influence on the average cycle time.
- The number of cards assigned to the line, the interarrival time, the interfailure time and the repair duration, keeping the availability constant, have a strong positive effect on the average cycle time.
- The batch size has only a small positive effect on the average cycle time. The effect was expected to be more significant for this parameter. The setup ratio has a positive effect on the performance of the pull systems, while the regression model for MRP indicates a negative effect of the setup ratio on the average cycle time.

The next chapter covers the variability of the performance indicators and summarizes the calculations done on confidence intervals to ensure accurate computation of estimated values.

# CHAPTER 8
# CONFIDENCE

To ensure a statistically sound interpretation of the observations, all the data was tested on a 95% confidence interval. The half-width [see 4.2.1] of the confidence interval of the point estimate was maintained below the 10% fraction of the mean as a measure of accuracy:

$$h \le \gamma \bar{X}(n),$$

where $\gamma$ is the given parameter, $\gamma$=0.1.

As the most variable configurations were unknown, confidence tests were done before and after the simulations.

The transient behavior of the systems was investigated prior to the experiments. Then, a 95% confidence interval was calculated for the configuration assumed to have the highest variability to guarantee the desired accuracy.

The configuration with the highest variability could only be determined by reasoning prior to a set of replications. The actual configuration was found after completing the experiments by analyzing each replication. If the configuration assumed to have the highest variability was found to actually produce the highest variability for an experiment, no additional calculations were performed. Otherwise, the confidence interval was recalculated.

The coefficient of variation for the cycle time, $\tilde{c}_{cycle}$, was computed for every replication:

$$\tilde{c}_{cycle} = \frac{\sqrt{s^2(t_{cycle})}}{E(t_{cycle})},$$

as the ratio between the sample standard deviation and the expected value of the cycle time. The half-width of the confidence interval of the configuration with the highest coefficient of variation was computed and assured to be below the 10% value of the expected value. This procedure was ensued for all replications done.

## 8.1 Transient Behavior

The transient behavior of the three control systems was primarily investigated to reveal the duration needed to produce stable statistics [see 4.1]. Furthermore, as most of the simulations were run for a non-terminating system, it was worth finding out, whether the statistics should only be initialized after a warm-up period. This would reduce the number of entities sent through the system and reduce the experimentation time. Table 8-1 lists the configurations for the analysis of the transient behavior for Kanban, CONWIP, and MRP.

Table 8-1: The configurations for the analysis of the transient behavior for Kanban, CONWIP, and MRP.

| System | Batch Size | Number of Cards | Interarrival Time |
|---|---|---|---|
| CONWIP | 1 | 10 | - |
| Kanban | 1 | 10 | - |
| MRP | 2 | - | 42 |

At a glance, the graphs [see Figure 8-1, Figure 8-2, and Figure 8-3] indicate that Kanban and CONWIP show very similar patterns, MRP reveals a very interesting behavior. For Kanban and CONWIP the transient phase ends after only 1,000 entities produced as the average cycle time evens out. For MRP it takes about 3,000 entities before the steady-state is approached. One could argue that it takes even longer as the average cycle time line starts to really even out only after more than 10,000 units have been processed.

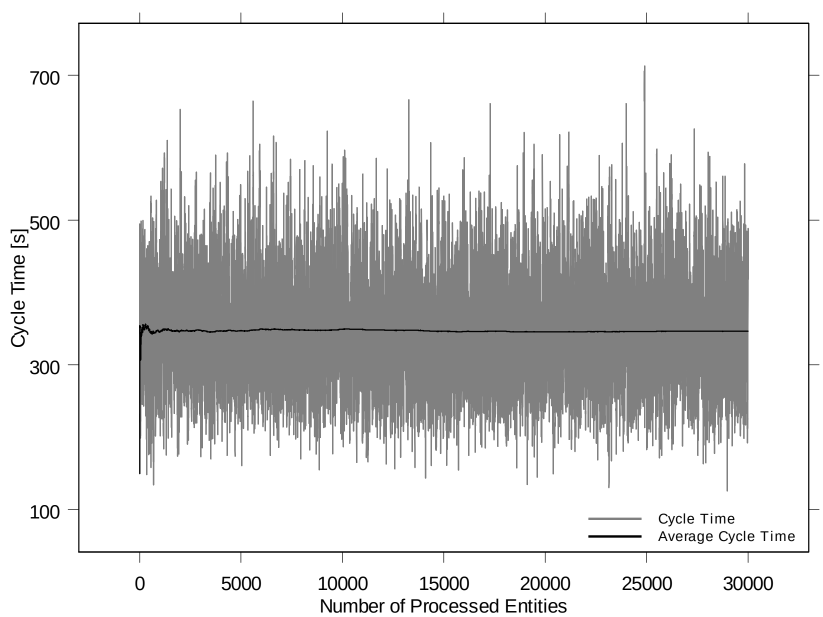


Figure 8-1: Cycle time and average cycle time dependent on the number of processed entities for CONWIP.

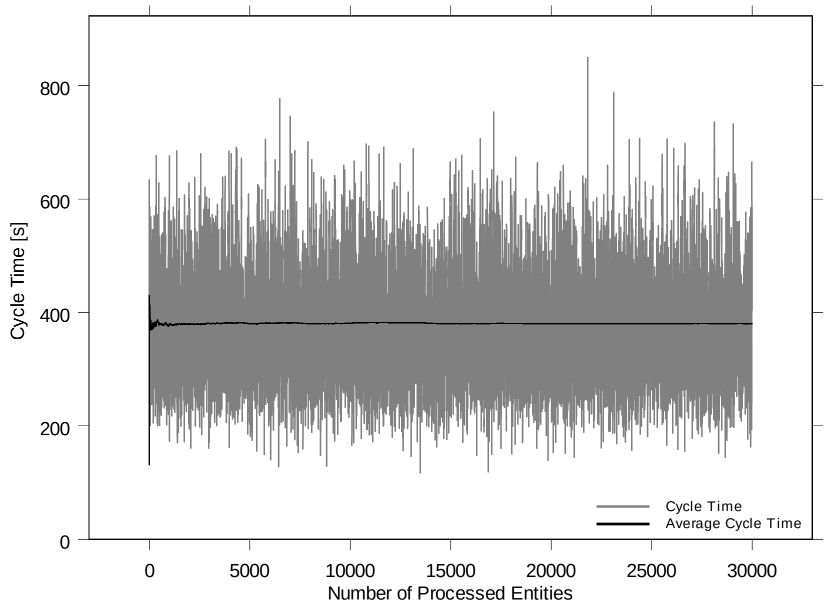


Figure 8-2: Cycle time and average cycle time dependent on the number of processed entities for Kanban.

Kanban has a few increases in the average cycle time for quite a big number of entities, finally reaching the steady state close to 15,000 entities. CONWIP has only a few minor bumps in the average cycle time line. For both these systems, the changes can most probably be attributed to the random generator. However, as the number of produced entities increases, the average no longer shows disturbances by the generated input values.

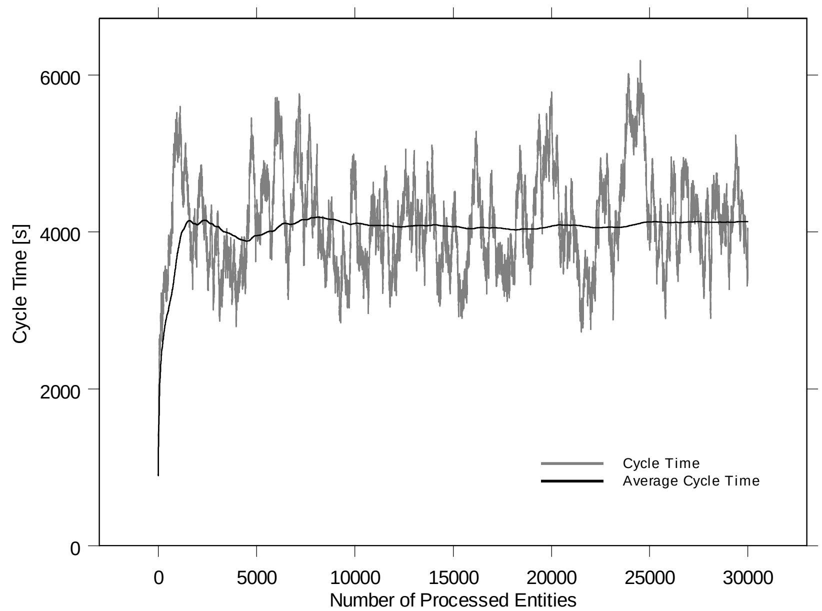


Figure 8-3: Cycle time and average cycle time dependent on the number of processed entities for MRP.

MRP exhibits a very different pattern. The cycle times increase and decrease almost cyclically. As the times differ so very much in length, the average cycle time is influenced even for a high amount of entities produced. This cyclic behavior can most likely be blamed on two factors: the random number generator and the build up of work in process [see 7.2]. As the random number generator calculates longer process times, the constant interarrival time keeps pushing the same amount of entities into the system. However, the entities ahead of them block their passage and trap them in the system, raising their cycle time. In comparison to the pull systems, the amount of entities prevented from passing through with the same average time is not limited. This results in quite a big number of entities with long cycle times, representing a large weight in the calculated average. This is true for the short cycle times as well.

Only when this weight decreases relative to the number of processed entities, the average cycle time will remain constant.

As the analysis revealed short transient times [see 4.1] for all the systems, 10,000 entities produced were thought to be sufficient to provide an accurate estimation of the statistical indicators of performance. The computations done on the confidence interval disclosed this assumption as truthful. These calculations are shown in the following sections.

However, one important issue remains: the warm-up period. Several points lead to the dismissal of taking warm-up into account:

- The configurations change considerably resulting in different steady-state times, the warm-up periods would have to be estimated for a large number of replications or chosen very long,
- The duration of one replication for producing 10,000 entities was about two minutes on average, not making time an issue, and
- Calculations of the confidence interval with a warm-up period didn’t show a great improvement of the confidence interval without one.

## 8.2 Batchsize

### 8.2.1 Prior to Simulations

The standard deviation is dependent on the work in process in the system and increases with increasing WIP. Thus, WIP should be maximum to obtain a maximal standard deviation. As the WIP increases, the average cycle time increases, resulting in a decrease of the coefficient of variation. Assuming the average cycle time increases faster than the standard deviation, the smallest WIP, and consequently, the smallest batch size and the least amount of cards, should be chosen. Table 8-2 shows the configuration chosen and its output.

Table 8-2: Configuration for Kanban and CONWIP to determine confidence interval prior to simulation and the corresponding utilization and coefficient of variation as the output.

| System | Batch Size | Number of Cards | Utilization | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|
| Kanban | 1 | 10 | 0.4297 | 0.2007 |
| CONWIP | 1 | 10 | 0.5276 | 0.2279 |

CONWIP's utilization was taken to calculate the interarrival time for MRP [see 5.2.1]:

$$\left.\begin{array}{l}\left.\begin{array}{l} Th_{theory} = \dfrac{1}{t_{process}} \\ \bar{u} = \dfrac{Th_{average}}{Th_{theory}} \end{array}\right\} \bar{u} = Th_{average}\, t_{process} \Leftrightarrow Th_{average} = \dfrac{\bar{u}}{t_{process}} \\ Th_{average} \approx \dfrac{1}{t_{intarr}} \Leftrightarrow t_{intarr} \approx \dfrac{1}{Th_{average}} \end{array}\right\} t_{intarr} = \frac{t_{process}}{\bar{u}},$$

$$t_{intarr} = \frac{20}{\bar{u}_{CONWIP}} = \frac{20}{0.5276} = 37.9075.$$

This resulted in the configuration and output given in Table 8-3.

Table 8-3: Configuration for MRP to determine confidence interval prior to simulation and the corresponding utilization and coefficient of variation as the output.

| System | Batch Size | Interarrival Time | Utilization | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|
| MRP | 1 | 38 | 0.5280 | 0.2384 |

To calculate the confidence interval on the expected value of the average cycle time, the cycle time for every entity leaving the system was saved in a file. This data was analyzed according to the Method of Batch Means [see 4.2.2 Analysis for Non-Terminating Simulations, The Method of Batch Means]. Here, batch refers to data accumulations, allowing an unbiased statistical analysis of their means. It should not be confused with the system parameter influencing the performance of a control system. A correlation analysis then provided the lag length for greatest independence of data between two batches [see Figure 8-4].

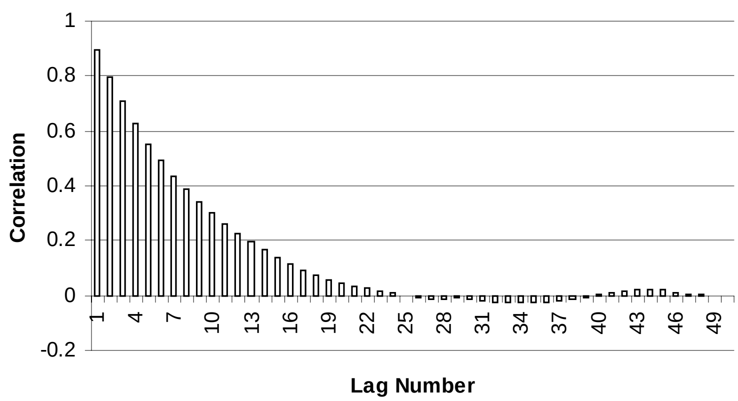


Figure 8-4: Correlogram for MRP indicating the correlation dependent on the lag number.

The amount of batches resulting is given in Table 8-4 as number of observations. It can clearly be seen that for Kanban a greater correlation of the data than for CONWIP existed, resulting in larger batches and a smaller number of observations. MRP showed the highest correlation of the data. Unfortunately, the correlation did not decrease with the assignment of different random number strings to the individual machines.

Table 8-4: Output for confidence interval calculations for CONWIP, Kanban, and MRP.

| Identifier | Average | Standard Deviation | 0.95 C.I. Half Width | Minimum Value | Maximum Value | Number of Observations |
|---|---|---|---|---|---|---|
| CONWIP | 381 | 71.3 | 3.43 | 217 | 706 | 1666 |
| Kanban | 349 | 43.5 | 3.41 | 228 | 532 | 625 |
| MRP | 357 | 41.5 | 5.79 | 277 | 492 | 200 |

The half-widths for the different systems are well below the 10% accuracy requirement. In the case of Kanban and CONWIP, they are even below a 1% accuracy level. Thus, producing 10,000 entities would result in a high confidence. To ensure that this was true for all the replications, the runs with the highest coefficient of variation were determined from the output data.

### 8.2.2 Succeeding Simulations

The output data indicated the configurations in Table 8-5 to have the highest coefficients of variation.

Table 8-5: Configuration for CONWIP, Kanban, and MRP resulting in the highest coefficient of variation of all the simulations run.

| System | Batch Size | Number of Cards | Interarrival Time | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|
| CONWIP | 1 | 10 | - | 0.2279 |
| Kanban | 1 | 11 | - | 0.2144 |
| MRP | 1 | - | 38 | 0.2384 |

The initial configuration chosen for CONWIP and MRP were confirmed to have the highest coefficient of variation. For Kanban a slightly higher coefficient was found for eleven instead of ten cards being assigned. This deviation was classified as not substantial. No additional calculations were done.

Table 8-6 exhibits the configurations for the minimal coefficients of variation. Comparing the values given in Table 8-5 and Table 8-6 reveals higher coefficients in general for the push system, MRP. This can most probably be attributed to the behavior described earlier in this chapter [see 8.1, Figure 8-3].

Table 8-6: Configuration for CONWIP, Kanban, and MRP resulting in the lowest coefficient of variation of all the simulations run.

| System | Batch Size | Number of Cards | Interarrival Time | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|
| CONWIP | 9 | 20 | - | 0.0751 |
| Kanban | 10 | 19 | - | 0.0756 |
| MRP | 10 | - | 252 | 0.0803 |

Table 8-6 unveils a minimal coefficient of variation for a batch size of 9 for CONWIP instead of a batch size of 10. This is due to the fact that simulations were only run for 17 cards with a batch size of 10. Every additional card would have resulted in a lower coefficient of variation. Comparing Table 8-5 and Table 8-6 reveals a decrease of the coefficient of variation for an increase in WIP for the pull systems. MRP shows the same behavior for an increase in WIP due to an increase in batch size. However, the response of the coefficient of variation to an increase in WIP due to a decrease in interarrival time changes for different batch sizes.

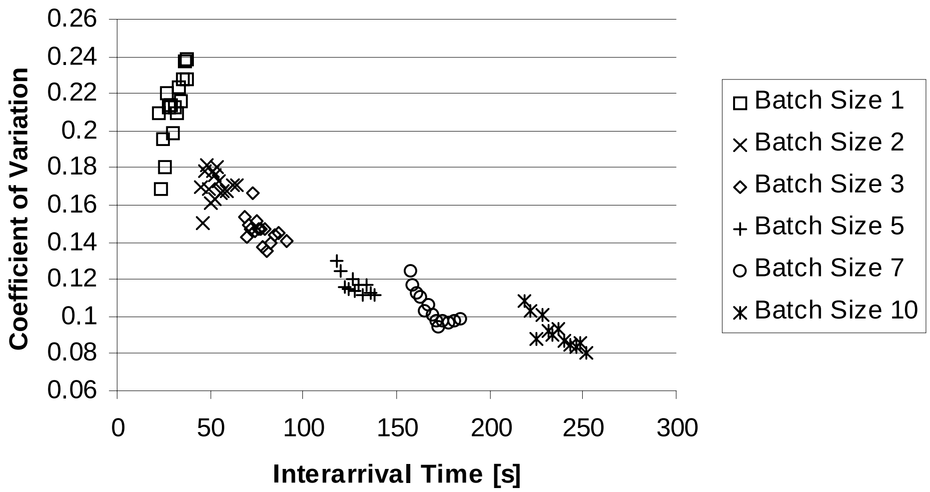


Figure 8-5: The coefficient of variation dependent on the interarrival time for MRP.

Figure 8-5 shows an increase of the coefficient of variation for an increase in interarrival time, resulting in a decrease in WIP, for a batch size of one. As the batch size increases, the decrease in the coefficient of variation for an increase in interarrival time, resulting in a decrease in WIP, becomes more evident. Thus, for MRP the standard deviation of cycle time increases as fast as the average cycle time with increasing WIP, for a given batch size. For the pull systems, the average cycle time increases faster than the standard deviation of cycle time, resulting in a decrease of the coefficient of variation for an increase of the WIP, for a given batch size.

## 8.3 Setup

### 8.3.1 Prior to Simulations

Assuming that the systems including setup time would behave as the systems without setup, similar initial configurations were chosen. For Kanban 10 cards were assigned, as the increase of the coefficient of variation for 11 cards was considered a unique phenomenon. As the coefficients tended to decrease for increasing throughput, the longest setup time was chosen, resulting in a low throughput [see Table 8-7].

Table 8-7: Configuration for Kanban, CONWIP, and MRP including setup time to determine confidence interval prior to simulation and the corresponding throughput and coefficient of variation as the output.

| System | Batch Size | Setup Time | Number of Cards | Interarrival Time | Throughput | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|---|---|
| Kanban | 1 | 200 | 10 | - | 0.1503 | 0.1936 |
| CONWIP | 1 | 200 | 10 | - | 0.1503 | 0.2192 |
| MRP | 1 | 200 | - | 399 | 0.1503 | 0.2151 |

Table 8-8 exhibits the results of the calculations done. Again, the half-widths lie well below the 10% value of the average, indicating a very accurate calculation of the expected value. The major difference between this and the previous results are the number of observations. While the number has decreased for the two pull systems, it has increased for MRP, allowing more accurate calculations [compare with Table 8-4]. Generally speaking, the correlation of the output data has increased by the introduction of setup.

Table 8-8: Output for confidence interval calculations including the setup time for CONWIP, Kanban, and MRP.

| Identifier | Average | Standard Deviation | 0.95 C.I. Half Width | Minimum Value | Maximum Value | Number of Observations |
|---|---|---|---|---|---|---|
| CONWIP | 4.02e+003 | 347 | 34.1 | 3.18e+003 | 4.92e+003 | 400 |
| Kanban | 3.65e+003 | 408 | 35.8 | 2.7e+003 | 5.22e+003 | 500 |
| MRP | 3.76e+003 | 520 | 56 | 2.65e+003 | 5.64e+003 | 333 |

Although the half-widths were found to easily satisfy the 10% criterion, an analysis of the output data was performed.

### 8.3.2 Succeeding Simulations

The coefficients of variation were analyzed after the completion of 4,626 replications. Table 8-9 lists the three configurations with the highest coefficients.

Table 8-9: Configuration for Kanban, CONWIP, and MRP including setup time to determine confidence interval succeeding the simulations and the corresponding throughput and coefficient of variation as the output.

| System | Batch Size | Setup Time | Number of Cards | Interarrival Time | Throughput | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|---|---|
| Kanban | 1 | 200 | 12 | - | 0.1366 | 0.2027 |
| CONWIP | 1 | 200 | 10 | - | 0.1492 | 0.2181 |
| MRP | 1 | 200 | - | 271 | 0.2208 | 0.2403 |

Comparing the configurations found with the highest coefficients of variation to the configurations determined prior to the replications [see Table 8-7], no major deviations could be discerned. The batch size and setup time were equal. For CONWIP a small change in the number of cards assigned occurred. MRP showed an unexpected increase in variability for a decrease in interarrival time. Figure 8-6 illustrates the unusual behavior of the configuration. Although the utilization decreases steadily with an increasing interarrival time, the variability of the specific interarrival time shows an extraordinary jump. However, disregarding the exceptions, the expected increase of variability with a decreasing utilization for batch size one can be discerned in the graph.

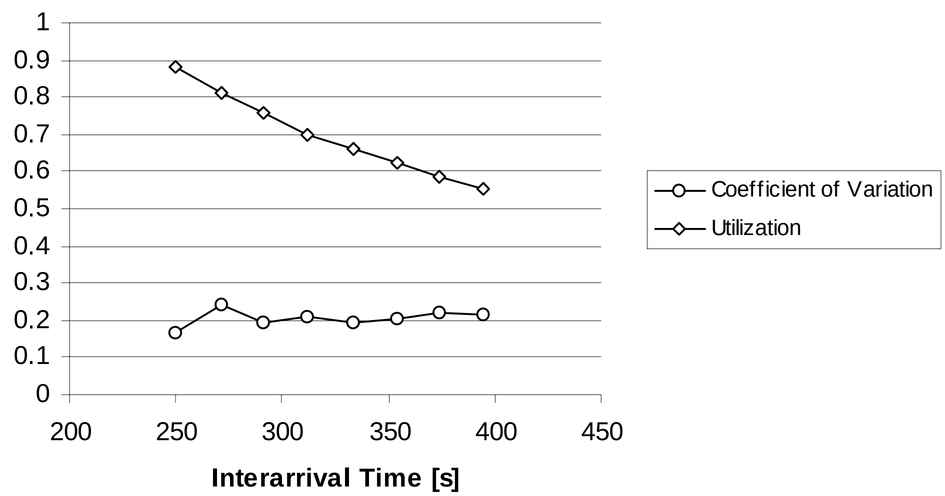


Figure 8-6: The coefficient of variation and the utilization dependent on the interarrival time for MRP with batch size one and setup time 200.

The extraordinarily high variation for the interarrival time of 271 can most probably be ascribed to the random generator, as the utilization does not show an unusual behavior. A jump in the utilization would have been most unlikely for a constant interarrival time resulting in a predetermined throughput and utilization.

Table 8-10: The coefficients of variation prior to the simulations and succeeding the simulations and their difference including setup for Kanban, CONWIP, and MRP.

| System | $\tilde{c}_{cycle}^{prior}$ | $\tilde{c}_{cycle}^{succeeding}$ | $\Delta\tilde{c}_{cycle}$ |
|---|---|---|---|
| Kanban | 0.1936 | 0.2027 | 0.0091 |
| CONWIP | 0.2192 | 0.2181 | -0.0011 |
| MRP | 0.2151 | 0.2403 | 0.0252 |

The confidence intervals were not recalculated for the differing configurations, as the differences in the coefficients of variation for the cycle times, $\Delta\tilde{c}_{cycle}$ , were considered to be negligible [see Table 8-10]. The half-widths calculated prior to the bulk of the simulations were sufficiently small to allow small deviations.

## 8.4 Failure

With a new source of variability added to the models, the machine failure, the amount of entities processed to ensure an accurate calculation of the indicators had to be reevaluated. The small half-widths determined for setup indicated some tolerance for increasing variability. However, especially the variability for MRP was expected to increase tremendously, as a large amount of entities would be trapped in the manufacturing line during repair.

Parallel to CHAPTER 7 incorporating the discussions on the influence of machine failure on the performance of the manufacturing system, the dynamic response is dealt with first, followed by the response to machine failure in steady-state.

### 8.4.1 Dynamics of Failure

The confidence on data for the two areas of investigation, the time spent in the system [see 7.2.3] and the time to recover after failure [see 7.2.4], is given below.

#### 8.4.1.1 Time Spent in the System

To enable a comparison of the three manufacturing systems observing their dynamic behavior to failure, 20 replications were completed. The calculated half-width of the 95% confidence intervals of their mean was found to exceed the 10% limit. Additional 10 replications were run and the t-test was repeated for the passing point 11 [see 7.2.3]. The coefficients of variation for the data collected at this passing point were very close to the highest for Kanban and CONWIP and the highest for MRP. The output of the computations is given in Table 8-11.

Table 8-11: The output for t-tests done for the time after failure at passing point 11 for Kanban, CONWIP, and MRP.

| Null Hypothesis | | | | | | |
|---|---|---|---|---|---|---|
| True mean of differences is equal to 0. | | | | | | |
| Output | | | | | | |
| System | t | df | p-value | 95% Confidence Interval | 95% C.I. Half-width | Mean estimate |
| Kanban | 29.1104 | 30 | 0 | (4763.814; 5482.668) | 359.427 | 5123.241 |
| CONWIP | 28.8857 | 30 | 0 | (4986.223; 5744.937) | 379.357 | 5365.580 |
| MRP | 27.8355 | 30 | 0 | (4899.202; 5675.026) | 387.912 | 5287.114 |

The half-widths clearly meet the 10% requirement, ensuring statistically confident observations.

#### 8.4.1.2 Recovery Time

To ensure an accurate estimation of the mean times, only 20 replications were needed for the pull systems and double the amount was needed for MRP [see Table 8-12]. This immediately indicates the greater variability of the push system, which can primarily be attributed to the high level of work in process accumulating in the system during failure.

Table 8-12: The results for the calculation of the confidence intervals for the time after failure and the moving average of the cycle times for Kanban, CONWIP, and MRP.

| System | Indicator | t | Df | p-value | 95% C.I. | Half-Width | Mean |
|---|---|---|---|---|---|---|---|
| Kanban | $t_{now}$ | 24.221 | 19 | 0 | (9064.196; 10778.908) | 857.356 | 9921.552 |
| Kanban | $\bar{t}_{cycle}^{exps}$ | 142.1707 | 19 | 0 | (5294.726; 5452.952) | 79.113 | 5373.839 |
| CONWIP | $t_{now}$ | 31.4352 | 19 | 0 | (10436.21; 11925.08) | 744.435 | 11180.64 |
| CONWIP | $\bar{t}_{cycle}^{exps}$ | 81.5805 | 19 | 0 | (5063.826; 5330.502) | 133.338 | 5197.164 |
| MRP | $t_{now}$ | 23.4124 | 39 | 0 | (19825.32; 23574.83) | 1874.755 | 21700.08 |
| MRP | $\bar{t}_{cycle}^{exps}$ | 451.5923 | 39 | 0 | (5296.451; 5344.110) | 23.8295 | 5320.28 |

The half-widths for the confidence intervals calculated for the time after failure, $t_{now}$, meet the 10% criterion, while those calculated for the exponentially smoothed cycle times, $\bar{t}_{cycle}^{exps}$, indicate a very accurate estimation of the mean [see Table 8-12].

### 8.4.2 Machine Failure in Steady-State

As done previously, the configuration with the highest variability had to be determined to enable accurate calculations of the expected values for the indicators. In a first step, the influence of the interfailure time and the repair duration on the performance of the manufacturing control system was investigated. Then the levels for the relevant parameters could be set with the information gained by constructing regression models.

**8.4.2.1 Influence of Interfailure  Time and Repair Duration**

The influence of increasing interfailure time and repair duration for a constant availability can be seen in Table 8-13 for a low utilization level and in Table 8-14 for a high utilization level. The regression was performed for eleven combinations of interfailure time and repair duration and varying batch size from size one to size ten [see Table 7-5].

Table 8-13: The response of the average utilization to different combinations of interfailure time and repair duration and varying batch size for a small number of cards [see Table 6-2]assigned to a line controlled by CONWIP.

| Formula | | | | |
|---|---|---|---|---|
| Ul ~ Bsize + Bsize^2 + Combination | | | | |
| Coefficients | | | | |
| Identifier | Value | Std. Error | t-value | Pr(>\|t\|) |
| (Intercept) | 0.2799 | 0.0048 | 58.4823 | 0.0000 |
| Bsize | 0.0439 | 0.0018 | 24.6760 | 0.0000 |
| I(Bsize^2) | -0.0021 | 0.0002 | -13.2893 | 0.0000 |
| Combination | -0.0061 | 0.0004 | -16.9164 | 0.0000 |
| Residual Standard Error | | | | |
| 0.01202 on 106 degrees of freedom | | | | |
| Multiple R-Squared | | | | |
| 0.9679 | | | | |
| F-Statistic | | | | |
| 1066 on 3 and 106 degrees of freedom, the p-value is 0 | | | | |

Table 8-14: The response of the average utilization to different combinations of interfailure time and repair duration and varying batch size for a large number of cards [see Table 6-2] assigned to a line controlled by CONWIP.

| Formula | | | | |
|---|---|---|---|---|
| Uh ~ Bsize + Combination | | | | |
| Coefficients | | | | |
| Identifier | Value | Std. Error | t-value | Pr(>\|t\|) |
| (Intercept) | 0.5451 | 0.0055 | 99.4371 | 0.0000 |
| Bsize | 0.0111 | 0.0007 | 16.6929 | 0.0000 |
| Combination | -0.0094 | 0.0006 | -15.6008 | 0.0000 |
| Residual Standard Error | | | | |
| 0.02 on 107 degrees of freedom | | | | |
| Multiple R-Squared | | | | |
| 0.8299 | | | | |
| F-Statistic | | | | |
| 261 on 2 and 107 degrees of freedom, the p-value is 0 | | | | |

In both cases, for the low and high average utilization level, the combination has a significant effect on a high confidence level on the average utilization. The negative gradient of the combination indicates, that the average utilization decreases with an increasing interfailure time and repair duration. However, the small coefficient for the combination in comparison to the coefficient for the batch size indicates a minute effect on the average utilization and thus on the performance of the manufacturing line [see Figure 7-10].

Comparing the low and high utilization levels, the sensitivity of the average utilization to an increasing interfailure time and repair duration increases from the low level to the high level. The decrease of the coefficient of determination indicates an increase of variance with increasing utilization. The model provided is not able to describe the interdependence of the regressors with the response sufficiently.

After constructing the models and observing a significant effect (on a 95% confidence level) of the increase in interfailure time and repair duration on the utilization, the levels for the relevant parameters could be set.

#### 8.4.2.2 Prior to Simulations

With the aim to determine the configuration with the highest variability before the experiments were carried out, the negative correlation between utilization and variability lead to the following interdependence:

The lower the utilization, the higher the variability.

However, not all of the variability can be derived from the utilization level. The amount of material trapped in the system during repair plays a significant role on a system's performance as well. As the number of entities in the system increases during machine failure and repair, the number of entities with a longer cycle time increases and consequently, the average cycle time increases. Once the entities held in the line upstream from the failed machine pass out of the system, the average cycle time decreases rapidly. This increases variability on the average cycle time. Thus, there exists a trade-off between low utilization, which results from a low WIP level, and the high WIP level during the repair time. The setup time manipulates both factors. As the setup time increases, the utilization decreases. On the other hand, an increase in setup time decreases the reactivity of the system to failure as it takes longer for each entity to pass through the system. The material trapped upstream the failed machine is not built up as fast during failure. Furthermore, with increasing setup time, keeping the batch size constant, the average time spent in the system increases as well, making a fairly short repair duration less effectual.

Keeping the trade-off in mind and judging by the results obtained from previous experiments, the parameters were set at the following levels:

- Small number of cards assigned to the line,
- Small batch size,
- small setup time,
- Low availability.

The influence of the interfailure time and the repair duration was not taken into account. Table 8-15 lists the resulting configurations.

Table 8-15: The configurations for Kanban, CONWIP, and MRP including machine failure prior to simulations.

| System | Batch Size | Setup Time | Interfailure Time | Repair Duration | Number of Cards | Interarrival Time | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|---|---|---|
| Kanban | 1 | 20 | 28200 | 1800 | 13 | - | 1.199 |
| CONWIP | 1 | 20 | 28200 | 1800 | 10 | - | 1.261 |
| MRP | 1 | 20 | 28200 | 1800 | - | 71 | 1.082 |

Calculations done for 10,000 entities processed, revealed large half-widths. Consequently, the number of entities passing through the manufacturing line was increased until the half-width easily met the 10% criterion. The number of entities processed was increased by additional units to buffer unexpected variability. Table 8-16 shows the resulting amount of entities processed to ensure adequate accuracy for the calculation of the expected values.

Table 8-16: The amount of entities processed to ensure good estimation of indicators including machine failure.

| Control System | Amount of Entities Processed |
|---|---|
| Kanban | 15,000 |
| CONWIP | 20,000 |
| MRP | 100,000 |

The computations resulted in the numbers illustrated in Table 8-17.

Table 8-17: Output for confidence interval calculations including machine failure for Kanban, CONWIP, and MRP.

| Identifier | Average | Standard Deviation | 0.95 C.I. Half Width | Minimum Value | Maximum Value | Number of Observations |
|---|---|---|---|---|---|---|
| Kanban | 1.06e+003 | 839 | 52 | 476 | 8.77e+003 | 1000 |
| CONWIP | 1.08e+003 | 820 | 56.9 | 523 | 6.14e+003 | 800 |
| MRP | 1.05e+004 | 2.5e+003 | 292 | 6.06e+003 | 1.87e+004 | 285 |

### 8.4.2.3 Succeeding Simulations

Fortunately, the number of entities processed was chosen fairly high. The difference between the coefficients of variation for the configurations chosen prior to the bulk of simulations [see Table 8-15] and the highest coefficients for all configurations simulated were tremendous. Table 8-18 illustrates the configurations with the highest coefficient of variation for the average cycle time.

Table 8-18: The configurations for Kanban, CONWIP, and MRP including machine failure succeeding the simulations.

| System | Batch Size | Setup Time | Interfailure Time | Repair Duration | Number of Cards | Interarrival Time | $\tilde{c}_{cycle}$ |
|---|---|---|---|---|---|---|---|
| Kanban | 1 | 20 | 86400 | 5515 | 14 | - | 1.929 |
| CONWIP | 1 | 20 | 86400 | 5515 | 10 | - | 2.365 |
| MRP | 1 | 20 | 86400 | 3600 | - | 99 | 1.344 |

Table 8-19 shows the outputs for the calculations on the confidence interval. Unfortunately, the 10% criterion was not met for CONWIP. However, there were only two configurations out of 2,400, for which the coefficient of variation of the average cycle time exceeded a tolerable value. Thus, the output data could be regarded as a good estimation of the true indicators. The fairly small half-width for MRP indicates, that it was unnecessary to produce as many as 100,000 entities, which resulted in long simulation times. A smaller number would have sufficed.

Table 8-19: Output for confidence interval calculations including machine failure for Kanban, CONWIP, and MRP succeeding the simulations.

| Identifier | Average | Standard Deviation | 0.95 C.I. Half Width | Minimum Value | Maximum Value | Number of Observations |
|---|---|---|---|---|---|---|
| Kanban | 1.12e+003 | 1.47e+003 | 105 | 506 | 1.68e+004 | 750 |
| CONWIP | 1.2e+003 | 1.47e+003 | 130 | 573 | 1.34e+004 | 500 |
| MRP | 2.82e+003 | 1.88e+003 | 185 | 599 | 1.48e+004 | 400 |

The above investigation reveals a strong increase in variability for increasing failure time and repair duration. Looking at the half-width, the output data becomes almost double as variable for the same machine availability [compare Table 8-17 and Table 8-19].

The last chapter gives an overview of the conclusions made throughout prior discussions.

# CHAPTER 9
# CONCLUSIONS

## 9.1 Summary

In this thesis the performances of Kanban, CONWIP, and MRP were evaluated for a ten identical machine tandem line with respect to parameters including batch size, setup time, and machine failure. The utilization (throughput) was kept constant for all control systems. The parameters were introduced to the models one at a time, thereby increasing the realism and the variability of the manufacturing line. Thus, the performances of the three control mechanisms were explored on three levels of complexity. Initially, only the influence of batch size on the performances of the control systems was investigated. Then, the setup time was taken into consideration in addition to the batch size. Last, the machine failure was introduced to the models to augment the realism of the models resulting in a higher practical applicability. On each level, the performances were evaluated for steady-state, assuming the manufacturing line would run indefinitely. In addition, the response of the performance to machine failure was observed dynamically while keeping batch size and setup time constant.

Conclusions for the models including batch size variations only [see 5.2.3]:

- There is a linear dependence between the number of cards assigned and the average cycle time. The batch size influences the average cycle time non-linearly while the batch size and the number of cards interact. Thus, the batch size has a higher impact on the average cycle time than the number of cards assigned to the line.
- For an increasing batch size, the increase in average cycle time increases for every additional card assigned to the system.
- For a fixed number of cards allocated to the line, the average cycle time increases with increasing batch size.
- For a batch size greater than one, the average cycle time is always greater than zero. For a batch size of one, the average cycle time approaches zero as the WIP level approaches zero.
- For a given WIP level, an optimal configuration can be found for Kanban, CONWIP, and MRP resulting in the following ranking of their performance:
  1. MRP,
  2. CONWIP, and
  3. Kanban.
- For a given WIP level a lower batch size always results in a smaller average cycle time than a higher batch size, which seems to hold across all manufacturing control systems.

Conclusions for the models including both batch size and setup time variations [see 6.2.2 and 6.3.2]:

- The influence of the setup time on the average cycle time increases with decreasing batch size. Thus, for a big batch size an increase in setup time is not as detrimental for the performance of a system as for a small batch size.
- Batch size five separates the batch sizes into those with a higher sensitivity to change and those with a lower one. As a medium batch size, it encompasses the advantage of a small batch size and the good attributes of a big batch size. A small batch size results in a lower average cycle time, a big batch size has a lower reactivity to a change in setup time.
- Observing the average cycle time dependent on the batch size and the setup ratio, MRP outperforms the two pull systems. In contradiction with the previous findings , Kanban shows a better performance than CONWIP. Therefore, the order is:

  1. MRP,
  2. Kanban, and
  3. CONWIP.

Conclusions for the models looking at the dynamic response to machine failure [see 7.2.5]:

- Time spent in the system by the first entity trapped upstream the failed machine, the control systems can be ranked in the following manner:

  1. Kanban,

2. MRP, and
3. CONWIP.

- However, this ranking can not be stated with high confidence, as the differences in the time spent in the system are not highly significant.
- A closer look at the time needed for recovery after failure reveals the following order:

1. Kanban,
2. CONWIP, and
3. MRP.

Conclusions for the models combining batch size, setup time, and machine failure variations [see 7.3.3]:

- As the interfailure time and the repair duration increase keeping the availability constant, the average utilization decreases. Thus, it seems to be of advantage to implement preventive maintenance done frequently with a shorter duration. Waiting for the failure to occur resulting in bigger damage and longer repair times can reduce the performance of a system.
- Looking at the average cycle time as an indicator of performance, Kanban outperforms the other systems. The difference between Kanban and CONWIP is not as obvious as that between the pull system and the push systems. Therefore, the order is:

1. Kanban,
2. CONWIP, and
3. MRP.

- Table 9-1 lists the optimal configurations for the minimal average cycle time for Kanban, CONWIP, and MRP.

Table 9-1: The optimal configurations for the minimal average cycle time for Kanban, CONWIP, and MRP.

| System | $r_s$ | $b$ | $c$ / $t_{intarr}$ | $t_{intfail}$ | $t_{repdur}$ | $a$ | $\bar{t}_{cycle}$ |
|---|---|---|---|---|---|---|---|
| Kanban | 1 | 1 | 60 | 178200 | 1800 | 0.99 | 1888.963 |
| CONWIP | 1 | 1 | 44 | 86400 | 873 | 0.99 | 2101.542 |
| MRP | 1 | 1 | 48 | 86400 | 873 | 0.99 | 2457.908 |

- The same ranking results for the maximum cycle time as an indicator.
- Table 9-2 shows the optimal configurations for the minimal maximum cycle time for Kanban, CONWIP, and MRP.

Table 9-2: The optimal configurations for the minimal average cycle time for Kanban, CONWIP, and MRP.

| System | $r_s$ | $b$ | $c$ / $t_{intarr}$ | $t_{intfail}$ | $t_{repdur}$ | $a$ | $\bar{t}_{cycle}$ |
|---|---|---|---|---|---|---|---|
| Kanban | 1 | 2 | 50 | 86400 | 873 | 0.99 | 5865.879 |
| CONWIP | 1 | 2 | 37 | 86400 | 873 | 0.99 | 5939.28 |
| MRP | 1 | 4 | 120 | 86400 | 873 | 0.99 | 8928.158 |

- The difference in standard deviation of cycle time between the three control systems is considerable, resulting in the following ranking of the mechanisms:
  1. Kanban,
  2. CONWIP, and
  3. MRP.

- Table 9-3 illustrates the optimal configurations for the minimal standard deviation of cycle time for Kanban, CONWIP, and MRP.

Table 9-3: The optimal configurations for the minimal average cycle time for Kanban, CONWIP, and MRP.

| System | $r_s$ | $b$ | $c$ / $t_{intarr}$ | $t_{intfail}$ | $t_{repdur}$ | $a$ | $\bar{t}_{cycle}$ |
|---|---|---|---|---|---|---|---|
| Kanban | 1 | 4 | 25 | 86400 | 873 | 0.99 | 512.7491 |
| CONWIP | 1 | 2 | 37 | 86400 | 873 | 0.99 | 461.947 |
| MRP | 2 | 6 | 196 | 86400 | 873 | 0.99 | 1155.772 |

- The following summarizes the information obtained from constructing regression models for the three control systems with the average cycle time as the response variable:
  - The machine availability has a strong negative influence on the average cycle time.
  - The number of cards assigned to the line, the interfailure time and the repair duration have a strong positive effect on the average cycle time.
  - The setup ratio, the batch size and the interarrival time have only a small effect on the average cycle time.

Summarizing the above observations, MRP does not perform inferior to pull systems for a manufacturing line with reliable machines, when the release rate is selected judiciously. In fact, MRP tops the list on several experimental settings. Only in experiments with unreliable machines, the MRP system shows inferiority. For the more realistic setting, Kanban performs best, closely followed by CONWIP. MRP is placed third with a significant difference in performance.

## 9.2 Future Work

Although the realism was augmented throughout this study, the models could be improved further by introducing the following parameters:

- Transportation,
- Varying process times,
- different product types,
- move batches unequal to process batches, and
- irregular demand and supply.

Further, the parameter levels could be increased. Factor analysis instead of regression analysis could be applied to study the effect of the different parameters on the performance of the system, as the regression models were found to insufficiently predict performance outside the given intervals.

GLOSSARY

| | |
|---|---|
| $a$ | Machine availability |
| $b$, $Bsize$ | Batch size |
| $\beta$ | Regression model coefficients |
| $c$ | Number of cards assigned to line |
| $\tilde{c}$ | Coefficient of variation |
| $\tilde{c}_{cycle}$ | Coefficient of variation for cycle time |
| $e$ | Excess amount of data |
| E(X) | Expectation of the random variable X |
| $\varepsilon$ | Regression model random error |
| $h$ | Half-width of confidence interval |
| $I$ | Percentage increase in throughput |
| $M$ | Exponential distribution |
| $M_{beginning}$ | Resultant moment around beginning of line for card allocation rules |
| $M_{median}$ | Resultant moment around median of line for card allocation rules |
| $R^2$ | Multiple coefficient of determination |
| $r_s$ | Setup ratio |
| $s$ | Sample standard deviation |
| $s^2$ | Sample variance |
| $SSE$ | Sum of square errors |
| $Avgct$, $\bar{t}_{cycle}$ | Average cycle time |
| $\bar{t}_{cycle}^{exps}$ | Exponential smoothing of cycle times |
| $t_{delay}$ | Delay time |
| $Th$ | Throughput |
| $Th_{average}$ | Average throughput |
| $Th_{theory}$ | Theoretical throughput |
| $t_{intarr}$ | Interarrival time |
| $t_{intfail}$ | Interfailure time |
| $t_{process}$ | Process time |
| $t_{process}^{batch}$ | Batch process time |

| | |
|---|---|
| $t_{recover}$ | Recover time |
| $t_{repair}$ | Repair duration |
| $t_{setup}$ | Setup time |
| $u$ | Utilization |
| $\bar{u}$ | Average utilization |
| $WIP$ | Work in process |
| $WIP_{max}$ | Maximal work in process |
| $\bar{X}$ | Sample mean |

# REFERENCES


AFY98
Afyonoglu, G. Performance Assessment of MRP, Kanban, and CONWIP Manufacturing Systems with Setups using Emulated Flexible Manufacturing Laboratory
Master's Thesis at the Department of Industrial & Systems Engineering
Gainesville, FL: University of Florida, 1998

BAI98
Bai, S. X. Digital Simulation Techniques
Class Notes
Department of Industrial & Systems Engineering, University of Florida
Gainesville: University Copy Center 1998

BAK95
Baker, K. R. Lot streaming in the two-machine flow shop with setup times
Annals of Operations Research, 57, pp. 1-11, 1995

BER91
Berkley, B. J. Tandem queues and kanban-controlled lines.
International Journal of Production Research, 29(10), pp.2057-2081, 1991

BER92
Berkley, B. J. A review of the kanbn production control research literature
Production and Operations Management, 1(4), pp.393-411, 1992

BON97
Bonvik, A. M.; Couch, C. E.; Gershwin, S. B. A comparison of production-line control mechanisms
International Journal of Production Research, 35(3), pp. 789-804, 1997

CHA83
Chaudhry, M. L.; Templeton, J. G. C. A First Course in Bulk Queues
New York: John Wiley & Sons, 1983

CHU92
Chu, C. H., Shih, W. L. Simulation studies in JIT production
International Journal of Production Research, 30, pp. 2573-2585, 1992

CHY98
Chaouiya, C.; Liberopoulos, G.; Dallery, Y. The Extended Kanban System for Production Control of Assembly Systems
Laboratoire D'Informatique de Paris 6, LIP 6 1998/024

COX58
Cox, D.R. Planning of Experiments
New York: John Wiley & Sons, 1958

DAL95
Dallery, Y.; Liberopoulos, G. A New Kanban-Type Pull Control Mechanism for Multi-Stage Manufacturing Systems
Proceedings of the 3rd European Control Conference, 4 (2), pp. 3543-3548, Rome, 1995

DUE92
Duenyas, I. ; Hopp, W. J. CONWIP Assembly with Deterministic Processing and Random Outages
IIE Transactions, 24, pp. 97-109, 1992

DUE93
Duenyas, I.; Hopp, W. J.; Spearman, M. L. Characterizing the Output Process of a CONWIP Line with Determnistic Processing and Random Outages
Management Science, 39, pp. 975-988, 1993

GOL92
Gold, H. Performance Modeling of Batch Service Systems with Push and Pull manufacturing Mangement Policies
PhD thesis
Würzburg: Universitaet Würzburg, 1992

GST96
Gstettner, S.; Kuhn, H. Analysis of production control systems Kanban and CONWIP
International Journal of Production Research, 34(11), pp. 3253-3273, 1996

GUP97
Gupta, S. M.; Al-Turki, Y. A. Y. An algorithm to dynamically adjust the number of Kanbans in stochastic processing times and variable demand environment.
Production Planning & Control, 8(2), pp. 133-141, 1997

GUP98
Gupta, S. M.; Al-Turki, Y. A. Y. The effect of sudden material handling system breakdown on the performance of a JIT system
International Journal of Production Research, 36, (7), pp. 1935-1960, 1998

HAL83
Hall, R. W. Zero Inventories
Homewood, Illinois: Dow Jones-Irwin, 1983

HOP91
Hopp, W. J.; Spearman, M. L. Throughput of a constant work in process manufacturing line subject to failures
International Journal of Production Research, 29, (3), pp. 635-655, 1991

HOP96
Hopp, Wallace J.; Spearman, Mark L. Factory Physics
Foundations of Manufacturing Management
Boston: Irwin, 1996

HUA97
Huang, M.; Wang, D. Simulation study of CONWIP for a cold rolling plant
International Journal of Production Economics, 54, pp. 257-266, 1997

HUM98
Hum, S.; Lee C. JIT Scheduling Rules: a Simulation Evaluation
Omega, International Journal of Management Science, 26(3), pp. 381-395, 1998

JEN83
Jensen, A.; Chenoweth, H. H. Applied Engineering Mechanics
Fourth edition
New York: McGraw-Hill Book Company 1983

KEL98
Kelton, W. David; Sadowski, Randall P.; Sadowski, Deborah A. Simulation with Arena
Boston: WCB/McGraw-Hill, 1998

KIB88
Kimball, B. General principles of inventory control
Journal of Manufacturing and Operations Management, 1 (1), pp. 119-130, 1988

KIM98
Kim, Y.; Lee, D.; Kim, J. A Simulation Study on Lot Release Control, Mask Scheduling, and Batch Scheduling in Semiconductor Wafer Fabrication Facilities
Journal of Manufacturing Systems, 17 (2), 1998

KLE75
Kleinrock, L. Queuing Systems
Volume I: Theory
New York: John Wiley & Sons, 1975

KLN89
Klein, J. A. The Human Costs of Manufacturing Reform
Harvard Business Review, March-April, pp. 60-66, 1989

LAT81
Latham, D. A. “Are you among MRP’s Walking Wounded?”
Production and Inventory Management, Third Quarter, 1981

LIB95
Liberopoulos, G.; Dallery, Y. On the Optimization of a Single-Stage Generalized Kanban Control System in Manufacturing
INRIA/IEEE Conference on Emerging Technologies and Factory Automation
Paris, October 1995

LIT61
Little, J. D. C. A Proof of the Queuing Formula $L = \lambda W$
Operations Research, 9, pp. 383-387, 1961

MAS96
Di Mascolo, M.; Frein, Y.; Dallery, Y. An Analytical Method for Performance evaluation of Kanban Controlled Production Systems
Operations Research, 44 (1), 1996

MCC94
McClave, James T.; Dietrich, Frank H. Statistics
New York: Macmillan College Publishing Company 1994

MIJ97
Mijon, A. J. Emulated Flexible Manufacturing Laboratory Applications and Implementation
Master's Thesis at the Department of Industrial & Systems Engineering
Gainesville, FL: University of Florida, 1997

MOD83
Monden, Y. Toyota Production System: Practical Approach to Production Management
Norcross, GA: Industrial Engineering and Management Press, 1983

MON91
Montgomery, D. C. Design and Analysis of Experiments
Third Edition
New York: John Wiley & Sons 1991

NET90
Neter, John; Wasserman, William; Kutner, Michael H. Applied Linear Statistical Models
Regression, Analysis of Variance, and Experimental Design
Burr Ridge, Illinois: Irwin, 1990

OHN88
Ohno, T.; Mito, S. Just-in-Time for Today and Tomorrow
Massachusetts: Productivity Press 1988
Translation of Naze hitsuyeo na mono o hitsuyeo na bun dake hitsuyeo na toki ni teikyeo shinai no ka
Tokyo: Diamond 1986

PAT93
Patterson, M. C. Analysis of setup time at constraint resources
International Journal of Production Research, 31 (4), pp. 845-849, 1993

PEG95
Pegden, C. D.; Shannon, R. E.; Sadowski, R. P. Introduction to Simulation Using SIMAN
Second Edition
New York: McGraw-Hill, Inc. 1995

SCG95
Schoening, A. K.; Kahnt, M. Performance Modeling of Pull Manufacturing Systems with Batch Servers
INRIA/IEEE Symposium on Emerging Technologies and Factory Automation, 3, pp. 175-183, Paris, October 1995

SCH82
Schonberger, R. J. Japanese Manufacturing Techniques
Nine Lessons in Simplicity
New York: The Free Press 1982

SPE89
Spearman, M. L.; Hopp, W. J.; Woodruff, D. L. A Hierarchical Control Architecture for Constant Work-in-Process (CONWIP)
Journal of Manufacturing and Operations Management, 2, pp. 147-171, 1989

SPE90
Spearman, M. L.; Woodruff, D. L.; Hopp, W. J. CONWIP: A Pull Alternative to Kanban
International Journal of Production and Research, 28(5), pp. 879-894, 1990

SPE92
Spearman, M. L.; Zazanis, M. A. Push and Pull Production Systems: Issues and Comparisons
Opertions Research, 40(3), pp. 521-532, 1992

SUG77
Sugimori, Y.; Kusunoki, K.; Cho, F; Uchikawa, S. Toyota production system and kanban system materialization of just-in-time and respect-for-human systems
International Journal of Production Research, 15 (6), pp. 553-564, 1977

TAN97
Tan, B. Variance of the throughput of an *N*-station production line with no intermediate buffers and time dependent failures
European Journal of Operations Research, 101, pp. 560-576, 1997

TIT94
Titone, R. C. Integrating MRP II and JIT to achieve world-class status
Hospital Materiel Management Quarterly, 15(4), PP. 52-66, 1994

WAN96
Wang, D.; Chen, X.; Li, Y. Experimental push/pull production planning and control system
Production Planning & Control, 7(3), pp. 236-241, 1996

YAV95
Yavuz, I. H.; Satir, A. Kanban-based operational planning and control: simulation modeling
Production Planning & Control, 6 (4), pp. 331-344, 1995

# BIOGRAPHICAL SKETCH

Thomas Alfons Hochreiter was born in Windhoek, Republic of Namibia, to Alfons and Hedwig Luise on the 20th of October 1973. He completed his primary and secondary education at the Deutsche Höhere Privatschule, a German private school, in Windhoek in November 1992. Thomas continued his education at the University of Karlsruhe in Germany where he obtained his Vordiplom, the German equivalent to the Bachelor's degree, in Engineering Management in October 1995. He commenced his graduate studies in Karlsruhe before he came on a two semester exchange to the University of Florida. After completing his German education in August 1998, he resumed his tertiary education at the University of Florida to obtain the American Master's degree.