

UNIVERSITY OF VAASA
FACULTY OF TECHNOLOGY
DEPARTMENT OF INDUSTRIAL MANAGEMENT

Juha Sauna-aho

PRODUCTION CONTROL
Case: ABB Oy, Motors and Generators Vaasa

Master's Thesis in
Industrial Management

VAASA 2012

TABLE OF CONTENTS	page
SYMBOLS AND ABBREVIATIONS	5
ABTSTRACT	8
TIIVISTELMÄ	9
PREFACE	10
1. INTRODUCTION	11
1.1. Research problem	13
2. BASIC CONCEPTS	14
2.1. Workstations	14
2.2. Bottlenecks	15
2.3. Lead times, cycle times and on-time-delivery	16
2.4. Little's Law	18
2.3.1. Using Little's Law correctly	20
3. VARIABILITY AND BUFFERING	21
3.1. Quantifying variability	22
3.1.1. Process time variability	23
3.1.2. Flow variability	27
3.2. The combined effect of variability and utilization	28
3.3. Buffering	31
3.3.1. Buffer location	32
3.3.2. Reducing buffering as a continuous improvement scheme	33
3.3.3. Ease of management—a powerful reason to use capacity buffers	34

3.4. Pooling	36
3.4.1. Applications of pooling	38
4. PUSH AND PULL SYSTEMS	39
4.1. Definition of push and pull systems	40
4.2. CONWIP	41
4.3. Benefits of pull	42
4.3.1. Pull systems have less congestion	42
4.3.2. Pull systems are easier to control	43
4.3.3. Pull systems facilitate improvement measures	45
4.4. Applying CONWIP	45
4.4.1. Effect of parallel routings in the same CONWIP loop	46
4.4.2. Multi-loop CONWIP	48
4.5. Other production control concepts	49
4.5.1. Drum-Buffer-Rope	49
4.5.2. Simplified Drum-Buffer-Rope	51
4.5.3. Period Batch Control	52
5. CASE: ABB OY, MOTORS AND GENERATORS VAASA	54
5.1. Outline of the order fulfillment process and production	54
5.2. Job releases, DBR and CONWIP	57
5.3. Problems in the current CONWIP loops	60
5.4. Suggestion for an improved CONWIP configuration	61
5.4.1. CONWIP loop in assembly	62
5.4.2. Making shorter loops	63
5.4.3. Reducing parallel routings preceding assembly buffer	64

5.4.4. Releases based on winding locations	65
5.4.5. Intended benefits of the new configurations	66
5.5. Other approaches for improvement in the case company's production	67
5.5.1. Reduce queue time	67
5.5.2. Increase station overlap time and remove unnecessary operations	70
6. FURTHER CONWIP DISCUSSION WITH SIMULATION	72
6.1. Simulated systems and configurations	72
6.1.1. The Tandem system	73
6.1.2. The Purchasing system	73
6.1.3. The Parallel system	74
6.1.4. The Motors system	75
6.2. Simulation results	77
6.3. Conclusions based on the results	81
6.4. Discussion on the simulation study and CONWIP implementation	82
7. CONCLUSIONS	85
REFERENCES	88
APPENDIXES	91
APPENDIX 1. Effects of shorter cycle times and smaller cycle time variability	91
APPENDIX 2. Explanation of the blocks used in simulations	92

SYMBOLS AND ABBREVIATIONS

A	Availability. The portion of time that a workstation is available to produce.
A/B/C/D	A notation method for workstations in queuing systems where A is the arrival rate distribution, B is the processing time distribution, C is the number of parallel machines at a workstation and D is the maximum number of jobs in the system.
CONWIP	Constant work in process. A protocol used to control releases to a system.
CT	Cycle time. The time it takes on average for a job to traverse a routing.
CT_q	Queuing time. The average time a job spends in the queue of a workstation.
CV	Coefficient of variation. A relative measure of variability.
CV_0	Coefficient of variation of natural processing time.
CV_1	Coefficient of variation of an individual population.
CV_a	Coefficient of variation of the arrival rate.
CV_e	Coefficient of variation of processing time.
CV_n	Coefficient of variation of combined individual populations.
CV_s	Coefficient of variation of setup time.

FGI	Finished goods inventory.
m	The number of parallel identical machines.
MRP	Material requirements planning.
MTO	Make to order.
MTS	Make to stock.
MTTF	Mean time to failure.
MTTR	Mean time to repair.
n	Number of identical populations being combined.
N_s	Number of jobs per setup.
QRM	Quick response manufacturing.
T	Time term in the VUT equation. $T = t_e$.
t_0	The average natural processing time.
t_e	The average processing time.
t_s	The average time it takes to perform a setup.
TH	Throughput. The number of jobs produced in a time period.
TOC	Theory of constraints.

TPS	Toyota production system.
u	Utilization.
U	Utilization term in the VUT equation. $U = \frac{u}{(1-u)}$
V	Variability term in the VUT equation. $V = \left(\frac{CV_a^2 + CV_e^2}{2}\right)$.
WIP	Work in process.
μ	Average of a data set.
σ	Standard deviation of a data set.
σ_0	Standard deviation of natural processing time.
σ_e	Standard deviation of processing time.
σ_s	Standard deviation of setup time.

UNIVERSITY OF VAASA**Faculty of technology**

Author:	Juha Sauna-aho	
Topic of the Master's Thesis:	Production control	
Instructor:	Petri Helo	
Degree:	Master of Science in Economics and Business Administration	
Major subject:	Industrial Management	
Year of Entering the University:	2008	
Year of Completing the Master's Thesis:	2012	Pages: 93

ABSTRACT:

This thesis analyzes important concepts in production control from the perspective of a typical manufacturing plant. The scope is further limited to include theory that is especially relevant for the case company. The case company is an electric motor manufacturer ABB Oy, Motors and Generators Vaasa. The purpose of the research is first to develop understanding of theoretical concepts regarding production control. Secondly the case company will be used as an example to show some applications of the concepts discussed. The goal is to find the most effective tools for the development of the case company's production control.

The research is divided into three parts: a theoretical part based on literature on production control, to the analysis of the case company's production control and to a simulation study. The main focus will be given to principles that are directly applicable by the management of a manufacturing plant. The purpose of simulation will be to further increase the understanding of the theory discussed and to show the contrast of some varying production control configurations.

The research problem is: How can theoretical frameworks regarding production control be used for significant improvement in a typical manufacturing plant such as the case company? By discussing and clarifying many of the practical activities and processes in production control with a theoretical framework, the research shows that understanding such a framework can give managers valuable insights and perspectives for the development of processes.

KEYWORDS: Production control, operations management, variability, operations research, queuing theory.

VAASAN YLIOPISTO**Teknillinen tiedekunta**

Tekijä:	Juha Sauna-aho	
Tutkielman nimi:	Tuotannonohjaus	
Ohjaajan nimi:	Petri Helo	
Tutkinto:	Kauppätieteiden maisteri	
Oppiaine:	Tuotantotalous	
Opintojen aloitusvuosi:	2008	
Tutkielman valmistumisvuosi:	2012	Sivumäärä: 93

TIIVISTELMÄ:

Tämä tutkielma analysoi tuotannonohjauksen peruseriaatteita tyypillisen valmistusyrityksen näkökulmasta. Aihetta on lisäksi rajattu siten, että kohdeyrityksen tarpeet tulevat mahdollisimman tehokkaasti huomioitua. Kohdeyrityksenä toimii sähkömoottorivalmistaja ABB Oy, Moottorit ja generaattorit Vaasa. Tutkimuksen tarkoituksena on ensiksi kehittää ymmärrystä tuotannonohjauksen olennaisista teoreettisista käsitteistä. Toiseksi kohdeyritystä käytetään esimerkkinä teoreettisten käsitteiden soveltamisesta. Tavoitteena on löytää mahdollisimman tehokkaat työkalut kohdeyrityksen tuotannonohjauksen kehittämiseen.

Tutkielma jakautuu kolmeen osaan: teoriaosuuteen perustuen kirjallisuuteen tuotannonohjauksen alalta, kohdeyrityksen tuotannonohjauksen analysointiin, sekä simulointitutkimukseen. Pääpaino annetaan käsitteille joita valmistusyrityksen johto voi suoraan soveltaa tuotannonohjauksessa. Simuloinnin tarkoitus on kasvattaa teoreettista ymmärrystä, sekä tutkia läpikäytyjen käsitteiden vaikutusta simulaatiosysteemissä.

Tutkimusongelmana on: miten saavuttaa merkittävää kehitystä teoreettisia viitekehyksiä tuotannonohjauksen alalta hyväksikäyttäen kohdeyrityksen kaltaisessa tyypillisessä tuotantolaitoksessa? Käsittelemällä ja selkeyttämällä tuotannonohjauksen käytännön prosesseja teoreettisella viitekehysellä, tutkielma osoittaa että, teoreettisen näkökulman ymmärtämällä voi saavuttaa arvokkaita menetelmiä prosessien kehittämiseen.

AVAINSANAT: Tuotannonohjaus, toiminnanohjaus, hajonta, operaatioanalyysi, jonoteoria.

PREFACE

This master's thesis was written for ABB Oy, Motors and Generators Vaasa between 1.9.2011–6.6.2012. I would like to express my gratitude to ABB Oy, Motors and Generators Vaasa for giving me the opportunity to analyze a very interesting production environment for the purposes of this thesis.

Also I would like to thank the instructors of this thesis Tero Tammisto and Petri Helo for challenging me to explain my thinking more thoroughly and for pointing out topics that required further description. Finally a big thank you goes to everyone who have, in the process of writing this thesis, discussed its topics with me, and had the patience to listen to me explain my thinking.

In Vaasa on June 6, 2012

Juha Sauna-aho

1. INTRODUCTION

This thesis intends to study, discuss and apply theoretical concepts with significant practical implications regarding production control. What makes production control a challenging subject is that there is an infinite amount of possible production systems each having a different optimal control policy. Thus we cannot directly copy what the “best in the business” are doing. Instead we need to understand why some approach performs well in a particular production configuration and then apply the understanding to the production configuration that we are associated with. Even if we were to copy every single detail of a successful production facility and its control policy, we would still need to understand how it works as the business environment is subject to continuous change. We need to have the expertise to be able to react to and take advantage of the continuous change.

Companies can have many goals. The most common main goal of a company tends to be some variation of the following: to make money, have high ROI%, create quality goods with minimal costs, etcetera. The goal of production control is ultimately to help the company achieve its goal. This type of goal formulation does not help much in practice. However, for production control it is fairly easy to formulate practical sub-goals: low inventory investment, high throughput with low capacity investment, fast cycle times, high quality. The first two relate to keeping costs low and the latter two relate mainly to a high level of customer service.

In addition to the goals presented above—simplicity, lightness and ease of use of the production control system is essential. A simple system enables us to use it effectively and make the appropriate modifications as the business environment develops over time. We will refer to the reluctance and failure to update and modify systems and control policies as **inertia**. Using a method or system that is not understood is a recipe for inefficiency and will promote inertia. Often there can be an important tradeoff to be made between the seemingly (or temporarily) more effective alternative and the simple and robust alternative.

Historically possibly the two most eminent developers of production control have been Henry Ford, inventor of the flow line and Taiichi Ohno the genius behind the Toyota Production System (TPS) (Goldratt 2008: 3). Both of these men implemented their ideas in practice with unmistakable success. This alone proves that there is much to learn from the concepts that they have developed and used. However the systems that they developed were purpose-built for their specific business. Therefore it is necessary to investigate what were the fundamental reasons that their approaches were so effective.

This thesis will rely heavily on the approaches by four academics that I believe currently to represent the highest evolution in understanding production control, including the concepts developed by Ford and Ohno. These academics are Eliyahu Goldratt developer of the Theory of Constraints (TOC), Rajan Suri developer of Quick Response Manufacturing (QRM) and the writers of the book *Factory Physics*: Wallace Hopp and Mark Spearman. Especially the work of Hopp and Spearman is given precedence as they seem to be the best at not oversimplifying issues while still staying relevant in practical terms.

Some of the theoretical perspectives presented will be applied to a case company: ABB Oy, Motors and Generators Vaasa. The concepts discussed are meant to be as relevant and practical as possible to the production control of the case company and the average manufacturing company. By average I mean that the “extremes” of production are not addressed, such as commodity products (sugar, chemicals, oil, etcetera) and one of a kind very low volume production. More specifically the perspective used will be that of a **disconnected flow line**.

Finally a simulation study is performed to give an additional perspective to some of the behavior in a production plant previously discussed. Simulation is also used to evaluate the effect of some of the suggestions made for the case company. Simulation can be considered as a middle ground between a real-life system and pure theoretical ideas and concepts. Therefore it is a very powerful tool for creating further understanding. The most obvious benefit of simulation is, that the cost of a single simulation run is

miniscule compared to testing a new system in actual production. Additionally with simulation we can perform many runs with different configurations in a short time period.

1.1. Research problem

Much of the theoretical work in operations management is ignored by practitioners because much of it is simply not very useful (Hopp & Spearman 2011: 31; Hopp & Spearman 2000: 170). Instead it is common to turn to some oversimplified popularizations of management philosophy, for advice. Therefore the research question is formulated to support the investigation of the most impactful theoretical works while avoiding oversimplification: *How can theoretical frameworks regarding production control be used for significant improvement in a typical manufacturing plant such as the case company?*

2. BASIC CONCEPTS

A manufacturing facility with disconnected flow lines consists of workstations and buffers. These are linked via routings which determine the material flow between different workstations and buffers. Products that share the same routing are often considered to be a part of the same product family. A series of workstations and buffers that form a cohesive whole inside a production plant is often called a production line or an assembly line. In contrast a flow line is one that has a rigid routing and a paced material handling system, for instance an automobile plant with the frames all moving at the same time in even time intervals (Hopp et al. 2011: 10).

2.1. Workstations

There are three parameters that give the overall performance of a workstation. These are **average processing time**, **variability of processing time** and volume. Average processing time is the time it takes on average to process one batch. Variability of the processing time is the spread of the processing time. Volume is the amount of jobs in a batch. Consider a workstation that takes on average a day to process a job but occasionally takes only an hour and occasionally takes a week. Based on this limited information one might conjecture that the workstation is very slow and has a very high spread of processing times. Further suppose the workstation processes 1000 jobs at once. Now even though the workstation is slow it has high volume.

The performance of a workstation depends on two factors. First is the overall capability of the workstation. Second is the input rate of jobs into the workstation. The most effective input rate is of course, whenever the previous job has finished. This can be achieved with inventory or **work-in-process (WIP)** buffers. In effect a WIP buffer establishes a queue in front of the workstation, and thereby enables the workstation to start a new job whenever it has finished the previous job.

In queuing theory the notation $A/B/C/D$ is used to describe a queuing model, for instance a single workstation. Here A describes the distribution of the input rate, B describes the distribution of processing time, C describes the number of parallel identical machines inside the workstation and D is the maximum of the sum of jobs that can fit inside the buffer and workstation at once. For example a workstation described as $M/G/1/100$ has Markovian or exponential input rate, general processing times, one machine and space for 100 jobs inside the workstation and its buffer. General distribution is defined as any distribution possible. (Hopp et al. 2011: 283.)

2.2. Bottlenecks

The busiest workstation of a routing is its bottleneck, that is, the workstation with the highest utilization (Hopp et al. 2011: 315; Hopp 2008: 14). Let us give a definition for utilization and capacity

$$utilization = \frac{input\ rate}{capacity} \quad (1)$$

$$capacity = maximum\ average\ output\ rate \quad (2)$$

(Hopp 2008: 13–14).

The capacity of the bottleneck of a routing determines the capacity of the whole routing (Goldratt & Cox 2004: 145; Hopp & Spearman 2011: 248). The **throughput (TH)** of a routing is

$$TH = bottleneck\ capacity * bottleneck\ utilization \quad (3)$$

We can see from (3), that there are two ways to increase the throughput of a routing. First the capacity of the bottleneck can be increased, which can be done by buying new equipment or assigning more workers on the bottleneck. Second, utilization of the

bottleneck can be increased which is done by increasing buffering of the bottleneck. (Hopp et al. 2011: 340.)

However in practice a situation as simple as implied above is rare. Most practical routings involve multiple products with different processing times. This can cause the bottleneck to “float” depending on the product mix currently being processed. To circumvent this complication it may make sense to create a steady bottleneck. This can be done by ensuring that all other workstations have ample capacity excluding the workstation which is assigned to be the bottleneck (Hopp et al. 2011: 486–487). The obvious choice for the bottleneck is then the workstation for which adding capacity is the most expensive. In fact a workstation where capacity is cheap should never be the bottleneck (Hopp et al. 2011: 663).

Creating a steady bottleneck is an example of unbalancing the production line. In an unbalanced production line the capacity of different workstations is not the same. The underlying reason for an unbalanced line is the facilitation of bottleneck utilization with capacity buffers (Goldratt et al. 2004: 265–266; Hopp et al. 2011: 340). Hopp et al. (2011: 662–663) list three reasons for unbalanced production lines: (1) when a distinct bottleneck is present the production line is easier to manage; (2) it is typically cheaper to maintain excess capacity in some workstations; (3) often adding capacity is possible only in discrete-size increments, for example a new machine. Balanced lines are often maintained due to misguided utilization metrics and the ingrained notion that an efficient production line is a balanced one (Goldratt et al. 2004: 265–266; Hopp et al. 2011: 663).

2.3. Lead times, cycle times and on-time-delivery

Most important factors for satisfying customers in operational terms are a fast delivery time and a high **on-time-delivery (OTD)** (Hopp et al. 2011: 346). We define delivery time as the time in which a company promises to deliver a product from the time that

the customer placed their order. We define OTD as the ratio of times that a company successfully delivers the order within the time promised. Delivery time can be classified as a type of lead time. Lead time is a predetermined time that some process should take—usually a constant time period. Lead times are used for planning and quoting delivery times for customers. The problem with lead times is that real life processes are not constant and lead times often fail to give an accurate estimation of the actual time needed.

Cycle time (CT) is the actual time that some process has taken. This can be, for example the time taken by a single workstation or a whole plant. Usually when discussing CT we actually mean the average CT of many orders which share the same routing. With CT we are mainly concerned with two of its parameters: the average and variability. In the case of the CT of the whole company, these two parameters are the main determinants of the company's OTD and delivery time. The effect of the average cycle time is obvious but the effect of variability requires some further explanation.

Delivery time is determined by adding safety time to the average cycle time of a product family (Hopp et al. 2011: 346). If we were to determine a delivery time equal to the average cycle time, our OTD% would be around 50% which usually is not an acceptable level. The amount of safety time to be added depends on the level of variability of the cycle times and the level of OTD that we want to maintain. Figure 1 shows a comparison between two cycle times with different levels of variability but equal averages.

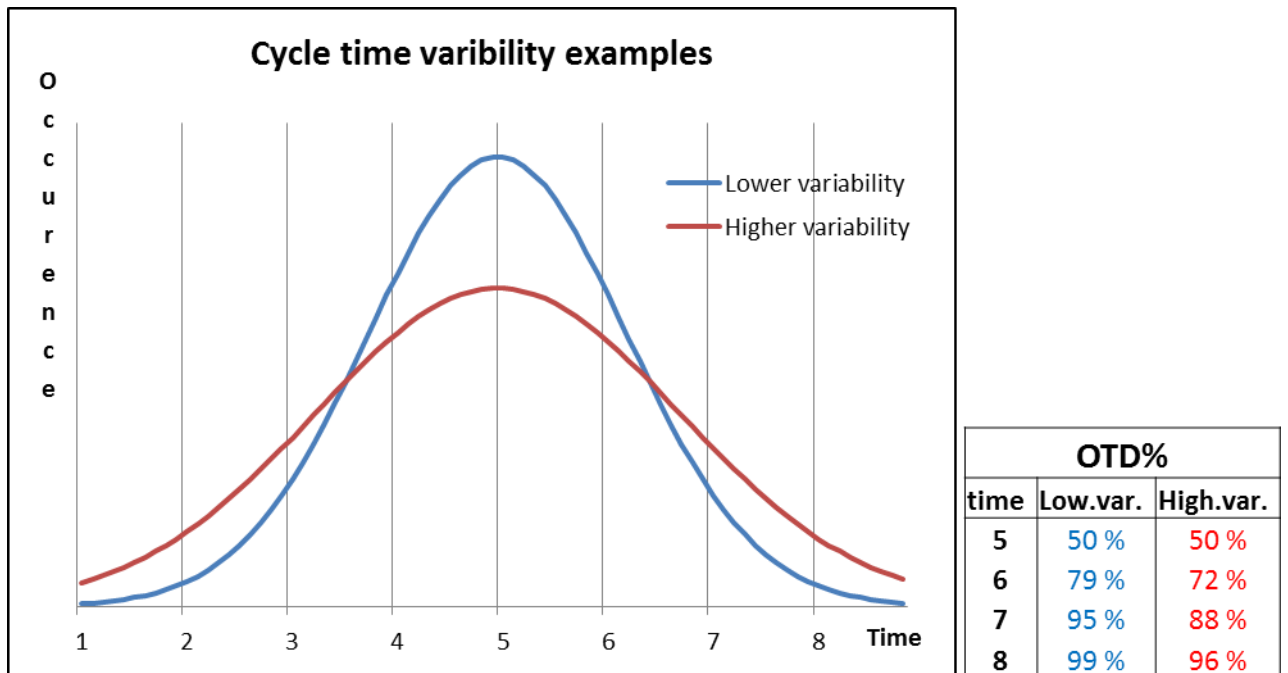


Figure 1. Example distributions of cycle times.

On the right we see the OTD% for the cycle time distributions with different time parameters. The OTD% can be calculated by summing the surface area of the distribution left of the time parameter chosen (assuming that the total surface area is one). From the OTD% table above we can see that if we wish to have an OTD level of 95%, we should set the delivery time to seven with the lower variability case and close to eight with the higher variability case.

Along with OTD and delivery time we should pay some attention to tardiness. Average tardiness is the sum of time that orders are late (Hopp et al. 2011: 517). For example average tardiness is the same level with one order late 10 days as with 10 orders late one day, whereas OTD% is much worse in the latter case. We can see from figure 1 that tardiness is also affected negatively by a higher level of variability in CT.

2.4. Little's Law

Little's Law is an equation from queuing theory which gives the relationship between the average number of items inside a queuing system, the average rate at which items arrive and the average time that an item spends in the system (Little & Graves 2008: 82). Over the past few decades the usefulness of Little's Law has become recognized in manufacturing management, where Little's Law is used to give the relationship between TH, WIP, and CT.

$$WIP = TH * CT \quad (4)$$

(Little et al. 2008: 92.)

What makes Little's Law widely applicable is that it does not require any assumptions regarding, for example arrival and processing time distributions, number of machines or queue disciplines (Little 1961: 387). In production control context Little's Law can be applied to a single workstation, a production line or a whole plant (Hopp et al. 2011: 239).

Whenever two of the terms in (4) are known the third can be quickly calculated. In a production facility it is often the case that TH and WIP are known but CT is not. One way to acquire CT would be to individually record the time each job spends in production and then calculate the average. This is very tedious and laborious so we use Little's Law instead. Hopp (2008: 24) points out that by Little's Law reducing CT and WIP are really two sides of the same coin. If TH stays constant a smaller WIP requires a smaller CT and vice versa. This implies that if we want improvements in CT we should look at where the WIP is piling up.

Suri (1998: 183–185) gives Little's Law two important uses in manufacturing management. First is setting consistent targets, for example a CT target of one day is clearly not feasible with a WIP target of 20 and a TH of 10 jobs per day. Second use is for performance reports, for example we can compute the actual CT of some department and compare the time against predetermined standard lead times.

2.3.1. Using Little's Law correctly

When using Little's Law first one must ensure that the units used are consistent. For example if CT is measured in days then TH must be measured in items per day. On the other hand, the units for TH must correspond to the units for WIP which can be measured, for example in jobs, parts, money or processing time at the bottleneck. Also if, for example we want to know the CT of a particular customer then we need to take into account only the WIP, TH and CT of jobs for that customer (Suri 2010b: 12–13).

As stated before all of the terms in (4) are averages. It is easy to choose misrepresentative values especially for WIP and CT as their values can fluctuate heavily. Consider a single workstation with an average WIP of five in a time period. We will get a very misleading result if we only check the WIP during a time when the WIP was at 15.

A condition where Little's Law should not be used is when there is considerable ramp up or ramp down during the time period being investigated (Suri 2010b: 12). The condition of having no ramp ups or ramp downs ongoing is called steady-state (Hopp et al. 2011: 285). For example in simulation studies it is often necessary to ignore the beginning time period due to the ramp up phase. This way only the steady-state situation is observed. Generally in a production facility input equals output. When this is not the case, that is, there is yield loss, Little's Law does not hold. In practice this is only an issue if the yield loss is considerable. (Suri 2010b: 12.)

3. VARIABILITY AND BUFFERING

As we see from the A/B/C/D notation there are two categories of variability in a queuing system: the arrival rate (A)—see figure 2 for an example of low and high variability arrival rates—and the processing time (B). For a production plant variability causes bursts and lapses in the amount of work. Bursts cause WIP to accumulate inside the plant as the capacity is not enough to handle the increase in work. By Little's Law this also causes an increase in CT. Lapses in the amount of work lead to wasted capacity, which translates to a smaller TH.

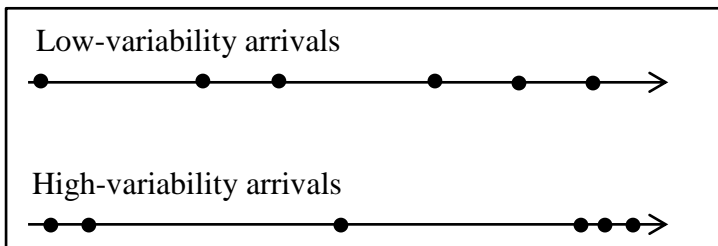


Figure 2. The contrast between low-variability arrivals and high-variability arrivals (Hopp et al. 2011: 279).

A big portion of variability is a strategic choice (Suri 2010: 4). In regard to arrival rate this means the ability to deliver products to customers at exactly the time and quantity they wish to have the products delivered. A company that aims for an excellent service, in terms of delivery time, predisposes itself to a very lumpy demand which translates into high arrival rate variability.

In regard to processing time variability, strategy determines the amount of customization of products that the company decides to perform in order to satisfy customers. High amount of customization causes a very variable product mix which in term causes high processing time variability. Hopp et al. (2011: 307) mention that Henry Ford can be considered to be almost fanatic about minimizing variability. He is frequently quoted of saying that a customer can have any color desired as long as it is

black. In the 1930s and 1940s when General Motors started to introduce greater product variety, Ford Motor Company lost a big portion of its market share to General Motors and came close to bankruptcy.

Suri (2010: 4) divides sources of variability to **strategic variability**—as described above—and **dysfunctional variability**. Dysfunctional variability is caused by errors, ineffective systems and poor organization. One task of production control is to reduce dysfunctional variability by minimizing errors, using and creating effective systems and creating an effective organization within production. Considering that removing variability completely is impossible as it is a fact of life (Hopp et al. 2011: 301; Suri 1998: 159) and that always some amount of strategic variability must be accommodated, another task for production control is to manage the inherent variability in production as effectively as possible. This is done with effective buffering.

3.1. Quantifying variability

In order to quantify and compare different sources of variability the **coefficient of variability (CV)** is used. To calculate the CV we need two parameters: standard deviation (σ), which gives the absolute variability of our data set; and the mean (μ) which is the average of our data set. By dividing σ with μ we get a relative measure of variability

$$CV = \frac{\sigma}{\mu} \quad (5)$$

(Hopp et al. 2011: 268.)

Hopp et al. (2011: 269) classify process times with a CV less than 0.75 as low variability, a CV between 0.75 and 1.33 as moderate variability and a CV above 1.33 as high variability.

3.1.1. Process time variability

Process time variability can be divided into three sources: **natural variability**, **preemptive outages** and **non-preemptive outages**. Natural variability accounts for variability during processing of the job itself. This includes, for example differences in operator speed or differences in the jobs being worked on: some jobs being faster to process, some slower. The other two sources are outages, that is, processing has to be stopped for a while. Preemptive outages are unexpected outages such as breakdowns or unexpected operator unavailability. Non-preemptive outages are outages that we have some control over as to when exactly they occur, such as setups. (Hopp et al. 2011: 271–275.)

We define processing time to be the time that a job causes the workstation to be busy. Suppose a job arrives to a workstation at 8:00. The operator starts to setup her machine to accommodate the job and finishes at 8:10 (non-preemptive outage). Then the processing of the job starts but the machine breaks down before the job is finished at 8:20 (preemptive outage). The operator manages to fix the problem at 8:50. Then starts to process the job again and finishes at 9:00. We get a total of one hour of processing time for this job. Suppose the next job is similar to the previous one and thus needs no setup. The operator starts at 9:00 and finishes at 9:10. We get a processing time of 10 minutes. Based on these two samples the processing time for this workstation seems to be quite variable.

As an example let's assume that we have a machine that never breaks down and that we always have an operator to fill in if the current operator needs to leave on an emergency. In other words we have no preemptive outages. To compute **processing time** (t_e) we need the average natural process time (t_0), average setup time (t_s) and the average amount of jobs processed between setups (N_s). Then assuming that the probability of doing a setup after any part is equal we have

$$t_e = t_0 + \frac{t_s}{N_s} \quad (6)$$

(Hopp et al. 2011: 276.)

To compute **process time variability** (σ_e) we need the natural standard deviation (σ_0) and the standard deviation of the setup time (σ_s).

$$\sigma_e = \sqrt{\sigma_0^2 + \frac{\sigma_s^2}{N_s} + \frac{N_s-1}{N_s^2} t_s} \quad (7)$$

(Hopp et al. 2011: 276.)

Suppose that we have five different product families which each require a setup and are processed on one machine. Further suppose that a product from any product family is equally likely to arrive to the queue of the machine. Then we have a 20% probability that the next job in the queue is the same type as the previous, that is, a 20% probability that no setup is needed. Then the expected amount of jobs between setups is

$$\begin{aligned} N_s &= 1 + \frac{1}{5} + \frac{1}{5^2} + \frac{1}{5^3} + \dots + \frac{1}{5^\infty} \\ N_s &= 1 + \frac{1}{5} N_s \\ N_s &= \frac{5}{4} \end{aligned}$$

As this is a geometric series we can simplify the calculation to

$$1+x^1 + x^2 + x^3 + \dots = \frac{1}{1-x} \quad (8)$$

(Zwillinger 2003: 38).

Where x is the probability that the next job in the queue is the same type. In our example $x = 0.2$.

Suppose we have measured the following values

$$t_0 = 10 \text{ minutes}$$

$$t_s = 5 \text{ minutes}$$

$$\sigma_0 = 0.9 \text{ minutes}$$

$$\sigma_s = 0.6 \text{ minutes}$$

Then for process time and standard deviation we get

$$t_e = 10 + \frac{5}{1.25} = 14$$

$$\sigma_e = \sqrt{0.9^2 + \frac{0.6^2}{1.25} + \frac{1.25 - 1}{1.25^2} * 5} \approx 1,38$$

And for the coefficients of variances: natural process time CV (CV_0), setup time CV (CV_s) and **process time CV (CV_e)** we get

$$CV_0 = \frac{0.9}{10} = 0,09$$

$$CV_s = \frac{0.6}{5} = 0,12$$

$$CV_e = \frac{1,38}{14} \approx 0,098$$

To compute the corresponding values with a preemptive outage we need to know: mean time to failure (MTTF), mean time to repair (MTTR) and the standard deviation of the repair times (σ_r). First we compute availability (A) which is the time that the machine is

not broken

$$A = \frac{MTTF}{MTTF+MTTR} \quad (9)$$

The process time and standard deviation are

$$t_e = \frac{t_0}{A} \quad (10)$$

$$\sigma_e = \sqrt{\left(\frac{\sigma_0}{A}\right)^2 + \frac{(MTTR^2 + \sigma_r^2)(1-A)*t_0}{A*MTTR}} \quad (11)$$

(Hopp et al. 2011: 273–274.)

If we have both preemptive and non-preemptive outages then we need to apply these formulas consecutively (Hopp et al. 2011: 277). We shall now do this with our example. First we replace t_0 and σ_0 in our preemptive formulas with the values for t_e and σ_e which we calculated previously in the non-preemptive case. Suppose we have measured that our machine breaks down on average once per day and it takes on average 30 minutes to fix the machine. With seven working hours per day we have a MTTF of 420 minutes and a MTTR of 30 minutes. Additionally suppose that the repair times are moderately variable with a CV of one which converts to $\sigma_r = 30$.

$$A = \frac{420}{420 + 30} \approx 0.933$$

$$t_e = \frac{14}{0.933} = 15$$

$$\sigma_e = \sqrt{\left(\frac{1.38}{0.933}\right)^2 + \frac{(30^2 + 30^2)(1 - 0.933) * 14}{0.933 * 30}} \approx 7,89$$

$$CV_e = \frac{7,89}{15} \approx 0,53$$

Including the breakdowns has a major effect on the variability of the machine. The CV_e increased from 0.098 to 0.52. We can conclude that the biggest source of variability for this workstation is clearly the breakdowns. In general breakdowns can easily generate massive amounts of variability. Thus it can be effective to attempt to prevent breakdowns with steady maintenance, that is, replace preemptive outages with non-preemptive ones.

3.1.2. Flow variability

In a production line, departures from one workstation become arrivals to another workstation. Thus the arrival variability of a workstation is equal to the preceding workstations departure variability. This variability in the transfer of jobs between workstations is called flow variability (Hopp et al. 2011: 279). Flow variability shows us how variability propagates downstream in a production line. Suri (1998: 181–182) calls this propagation of variability: “the ripple effect of variability”.

The **departure variability** (CV_d) from a workstation depends both on the variability of arrivals to that station and on the process time variability. Which variability contributes more depends on the utilization of the workstation. If a workstation has utilization close to 100% then departure variability is close to equal to the process time variability of the workstation. On the other hand a very low level of utilization leads to departure variability close to the arrival variability of that workstation (Hopp et al. 2011: 280). Hopp et al. (2011: 280) suggest a formula to estimate departure variability

$$CV_d = \sqrt{1 + (1 - u^2) * (CV_a^2 - 1) + \frac{u^2}{\sqrt{m}} * (CV_e^2 - 1)} \quad (12)$$

where u is the utilization of the workstation and m is the number of machines.

Suppose that in our example machine we have a capacity of one job per 15 minutes and that we see from history data that the machine has processed on average 16 jobs per day. With 420 minutes of working time per day we get an input rate of 0.0381 jobs per minute. Now utilization is

$$u = \frac{0.0381}{\frac{1}{15}} \approx 0.57$$

We have a 57% utilization for the machine.

In practice the inter-arrival times of jobs between workstations are rarely measured or known but the scheduled start day or the demand on production is often available (Hopp et al. 2011: 281). Starting with variability of the start date, that is, the variability of arrivals to the first workstation, and then computing the process time variability of individual workstations and then using the formula for CV_d , we can investigate the flow of production throughout the plant. Variability reduction possibilities in the beginning of the line should be given priority as variability early in a line propagates downstream and is therefore more disruptive (Hopp et al. 2011: 318).

3.2. The combined effect of variability and utilization

The levels of variability and utilization have important consequences in a manufacturing plant. In order to gain better intuition of these consequences we can turn to an equation from queuing theory called the **VUT equation**. The equation holds exactly for the M/G/1/∞ queue but is a good approximation for the G/G/1 queue and for a typical manufacturing system in general. Cases when the equation is not accurate are when utilization is larger than 0.95 or smaller than 0.1, or when the CVs are much greater than one. The VUT equation gives the queuing time (CT_q) of a workstation. (Hopp et al. 2011: 288–289.)

$$CT_q = V * U * T \quad (14)$$

The equation consist of: a variability term (V), a utilization term (U) and a time term (T).

$$V = \left(\frac{CV_a^2 + CV_e^2}{2} \right) \quad (15)$$

$$U = \left(\frac{u}{1-u} \right) \quad (16)$$

$$T = t_e \quad (17)$$

Cycle time of a workstation computed using the VUT equation is

$$CT = V * U * T + T \quad (18)$$

(Hopp et al. 2011: 288–289.)

Let us plot three examples with waiting time on the y-axis and utilization on the x-axis, one with both arrival and processing CV of 1.5 a second with CVs of 1 and a third with CVs of 0.5. Effective process time is 10 minutes. See figure 3.

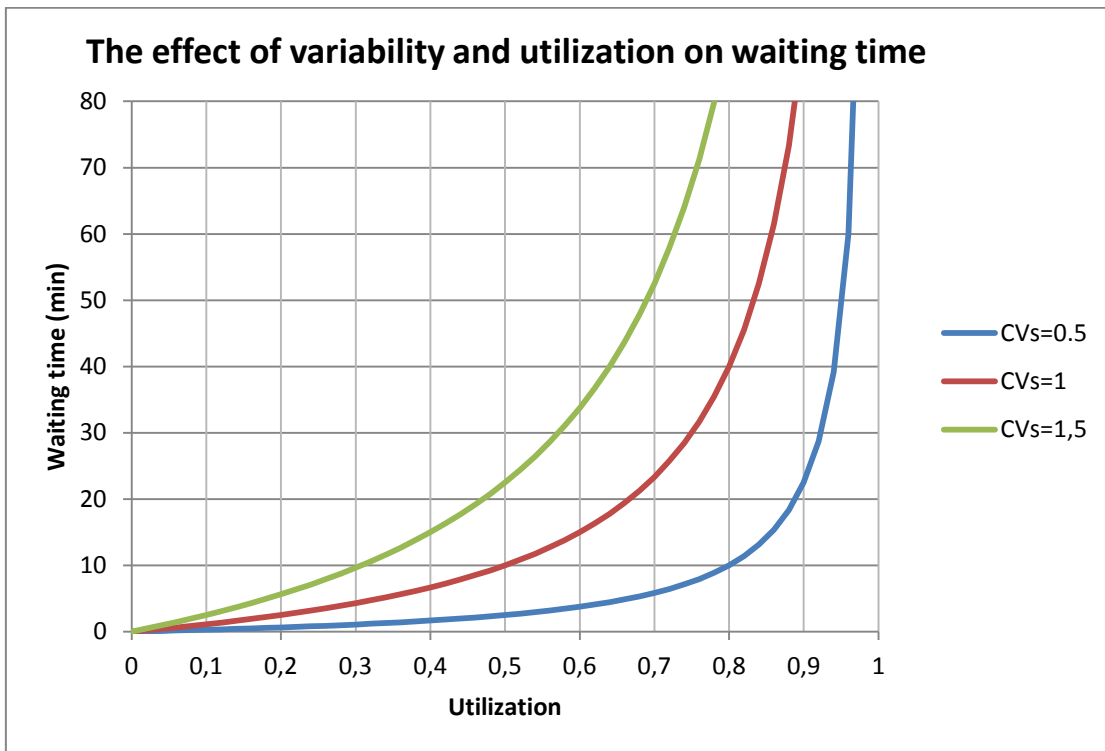


Figure 3. The effect of utilization and variability on queuing time (Hopp et al. 2011: 317; Hopp 2008: 32; Suri 1998: 168).

The first observation we can make from figure 3 is that higher variability causes higher queuing time. We see that waiting time exceeds 10 minutes at: 30% utilization for the 1.5 CVs case, at 50% utilization for the 1 CVs case and at 80% for the 0.5 CVs case. The second observation is that as utilization increases linearly, queuing time increases non-linearly. In fact with utilization of a 100% queuing time is infinity. In order to relate this theoretical concept with the real world it is important to realize that the theoretical model assumes an infinite time period and that no changes are made to the system during that time period. Obviously if the utilization of a workstation is close to 100% for one day the queuing time will not “explode”. But if we set a goal of utilization close to 100% for some workstation for months or a year then the consequences might be detrimental in terms of cycle time and WIP.

Based on the VUT equation we get a mathematical argument to reduce variability as much as possible and to plan for a utilization level less than 100%. The issue of

determining utilization levels is most important in practice when managing expensive machinery, where we have high motivation not to waste capacity. Suri (1998: 162–165) advocates that critical resources be planned to operate at 70–80% utilization. Ultimately the optimal utilization level depends on: (1) the price of capacity—no reason to have high utilization on stations with cheap capacity, (2) the amount of variability, (3) how long cycle times (and high WIP) are we willing to tolerate.

3.3. Buffering

All process time variability and arrival rate variability are buffered with some combination of time, inventory and capacity. The best possible mix of buffering depends on the strategy of the company and the nature of the production line being buffered (Hopp 2008: 81). The mix can be affected either intentionally or it can be the indirect consequence of management decisions. Figure 4 illustrates the choice of buffering mix.

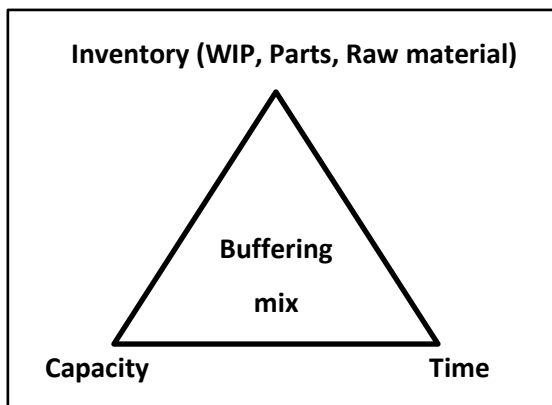


Figure 4. Illustration of the choice in buffering mix.

Most commonly buffering is considered to be the queuing of jobs in front of a workstation or some excess parts in inventory “just in case”. Building an inventory

queue is not always feasible. This is the case whenever producing products to stock is relatively expensive or impossible. Impossible cases would include tailored products where the specifications from the customer are needed before production can start, and services. For example if a machine breaks down in a plant, it has to either wait (buffering with time) or the repair crew must have ample capacity (buffering with capacity) to deal rapidly with the demand placed on the repair crew.

As discussed before bottlenecks in an unbalanced line are buffered with capacity. In fact if there is no capacity buffering in a production line, then all of the workstations in the production line have equal capacity and they are all bottlenecks. A common use of time buffers is in order quoting in a make-to-order (MTO) production plant. If the plant has too many orders to accommodate with a feasible delivery time, then a longer delivery time is simply quoted for the incoming orders. This way the demand on the plant (arrival variability of orders) is evened out along a longer time period.

The size of buffers can be reduced with flexibility. In terms of inventory, flexibility can be introduced by using generic parts that can be used for multiple products or by combining stocks of the same item in different locations. For example consider two similar parts used in assembly. To ensure that we don't run out of parts, some buffering stock is maintained. If engineering were to manage to replace those parts with one new part, the total buffering stock could be reduced. Flexible capacity can be introduced by training workforce in many different workstations or using multipurpose machinery (Hopp et al. 2011: 313–314). The core concept in making buffers flexible is called variability pooling (Hopp 2008: 149).

3.3.1. Buffer location

Increasing the utilization of bottlenecks can be thought of as the main purpose of buffering. The two ways that bottleneck utilization can suffer are: (1) the bottleneck is starved, that is, there are no jobs for it to process and (2) the bottleneck is blocked, that

is, there is not enough space succeeding the bottleneck to produce jobs. Generally the best place to add buffering in a production line is before or after the bottleneck, depending on which is more likely: blocking or starving. (Hopp 2008: 86.)

However from figure 3 we can see proof of why adding buffering to the bottleneck might not always be the best choice. Adding inventory buffering is equivalent to adding waiting time. Figure 3 shows that as utilization gets higher, more added waiting time is needed to get the same increase in utilization. In other words buffering has diminishing returns. The diminishing return of buffering also applies to buffering with capacity (without the added waiting time). Therefore sometimes adding buffering for non-bottleneck stations is more useful than adding buffering for bottlenecks. (Hopp 2008: 87–89.)

3.3.2. Reducing buffering as a continuous improvement scheme

Variability is harmful to production because it causes buffering (Hopp 2008: 89). Buffering is harmful to conducting business because it's expensive. The interaction between variability and buffers, and the fact that buffers have a diminishing return, imply that when variability is reduced buffering should be reduced approximately at the same pace. This gives us a basis for a simple continuous improvement framework, see figure 5.

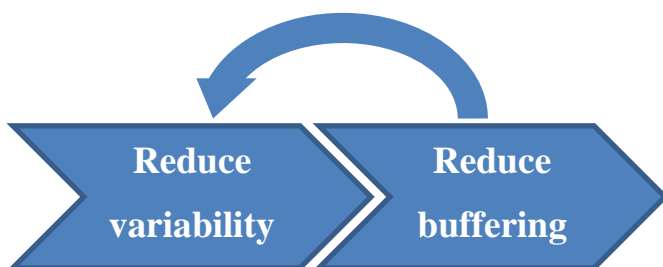


Figure 5. A continuous improvement framework based on variability reduction (Hopp 2008: 91).

Hopp (2008: 91) recommends that in order to facilitate variability reduction, inventory buffers should be replaced with capacity buffers (wherever possible). This will enhance visibility in the system which will enable us to identify the sources of variability more efficiently and help to eliminate them. This view is akin to the “WIP as water level and problems as rocks in the bottom of a lake” analogue (more on this in the discussion on pull systems). Finally when variability is reduced and there is some excess capacity buffering, the best way to eliminate the excess capacity is of course to increase TH and sales (Goldratt 2008: 20). In a typical situation—from the point of view of production—increasing sales is simply done by improving on-time-delivery and shortening delivery time, which should be possible to do if there is excess capacity.

3.3.3. Ease of management—a powerful reason to use capacity buffers

Large WIP buffers do more than just increase inventory investment and CT. They steal time from management for sorting out priorities and “traffic jams” (Goldratt 2008: 15). In fact it is fair to say that an excessive amount of expediting caused by traffic jams steal time from everybody—from the shop floor worker to the CEO. Long time buffers have a similar effect as forecasting becomes more and more inaccurate with longer time spans. Also with a long time buffer customers are more likely to cancel or revise their orders. In figure 6 Goldratt (2008: 15) illustrates the effect of buffer size to the attention required from management. The time buffer in figure 6 refers to the time given for production to finish jobs and thus by Little’s Law is synonymous to WIP buffers.

Goldratt (2008: 15–17) explains how conventional companies—located in the right hand side of figure 6—release orders to production too early, which causes high WIP, which in term causes along CT and “traffic jams”. This leads to missed due dates. To improve the due date performance, the conventional company decides to release orders even earlier which just causes more problems. Along with a long CT these companies can be identified with a poor on-time-delivery performance, and the prioritization system used in the company. The formal prioritization system isn’t used or it doesn’t

exist and the prioritization in practice is something on the lines with: ““hot”, “red-hot”, “drop everything—do it now””.

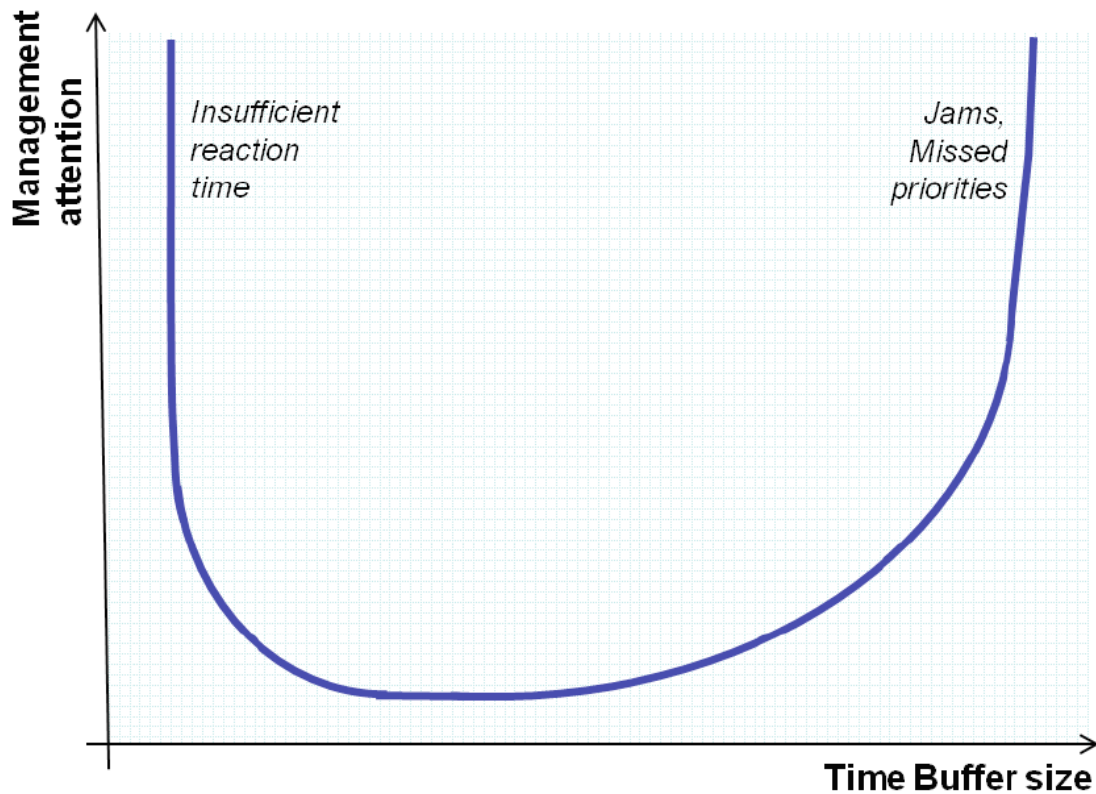


Figure 6. The effect of WIP buffers size on management attention (Goldratt 2008: 15).

A rational way to approach adding capacity buffering would be:

1. Make a list of all workstations and their utilization levels and the price of adding capacity.
2. Start with the workstation which has the lowest price of adding capacity.
3. If the utilization level is high invest in more capacity.
4. Go to the next workstation

Likely the most challenging part of the above list is determining the utilization levels. An intuitive way to do this is to check the queues that a workstation has historically

accumulated. If there are long queues most of the time (and the workstation is not subject to continuous blocking) then the level of utilization is obviously high.

3.4. Pooling

Variability is most damaging when the extremes occur. For example, a month where demand is much smaller than anticipated is more damaging than a few months of moderately low demand. During a month with inordinately high demand, most of the profit cannot be capitalized on, due to the inability to produce enough. With a few months of moderately high demand it is much easier to capitalize on the increase in demand. To significantly reduce frequency and level of the extremes we can “lump” sources of variability together. This causes the variability to even out, that is, the CV of the combined source of variability is smaller than the average CV of the individual sources. (Hopp 2008: 149–150.)

Consider two sources of variability both with a 2% probability of an “extreme” event. Half of these events are “highs” and half are “lows”. Now if these sources are combined into one, the probability of an extreme high (or low) reduces to $0.01 * 0.01 = 0.0001$. If one source has an extreme low and the other has an extreme high then the combined source is just experiencing the average event, which is what we are best prepared for. Suppose that the two original populations are normally distributed with a mean of 20 and a standard deviation of five. The contrast between the combined population and the individual populations is shown in figure 7. The x-axis shows the value of an event and the y-axis shows the probability of the event.

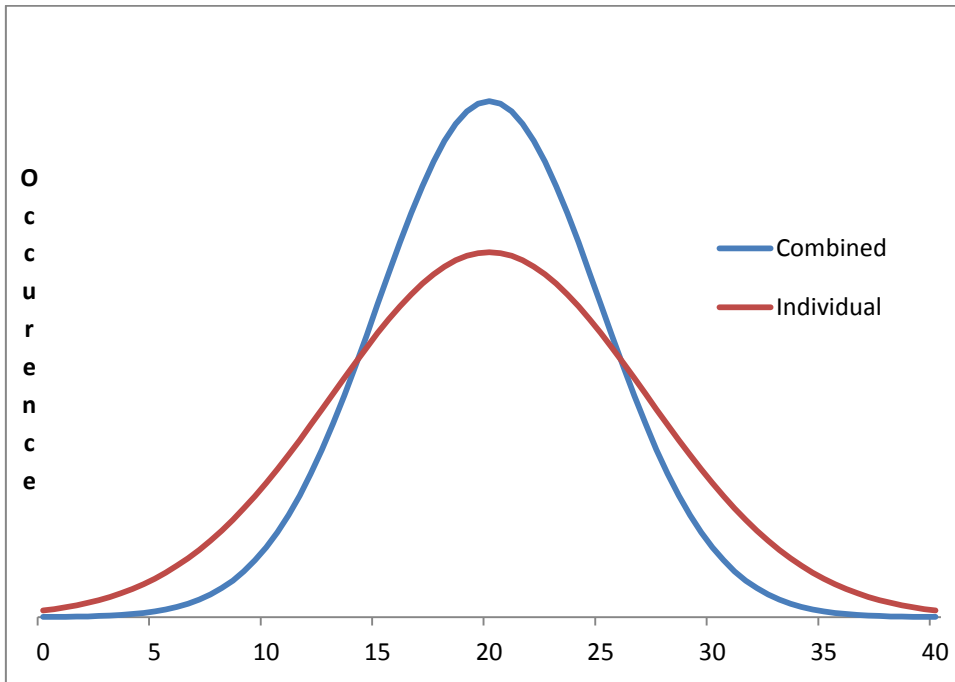


Figure 7. The effect of combining two individual identical sources of variability (Hopp 2008: 151)

Hopp (2008: 150) shows that when combining independent identical distributions the CV of the combined population CV_n is

$$CV_n = \frac{1}{\sqrt{n}} CV_1 \quad (19)$$

where CV_1 is the CV of the individual distributions and n is the amount of populations being combined. In our example we have

$$CV_1 = \frac{5}{20} = 0.25$$

$$CV_n = \frac{1}{\sqrt{2}} * 0.25 \approx 0.112$$

In practice we might not know the standard deviations of distributions and rarely know the shape of the distribution. This will make calculating the pooling effect accurately

difficult. Luckily we don't need to calculate the effect in order to benefit from it. Therefore significant sources of variability should be identified and the possibility of combining these sources investigated.

3.4.1. Applications of pooling

Hopp (2008: 154–160) lists some generic applications of pooling: centralization, standardization, postponement, work-sharing and chaining. The first two applications apply to inventory buffers. Centralization refers to combining inventories of the same parts. For example when two workstations in the same plant have a stock of the same part, combining the stock into one reduces the probability of running out of that part. Standardization refers to combining different parts into one by designing a part that can be used to replace the old ones.

Postponement is often the case when moving from make-to-stock (MTS) production to MTO production. This translates to substituting inventory buffers with time buffers. For example instead of having a large finished goods inventory (FGI), we wait for the customer to tell us their specific order before manufacturing is started. Here we are combining the variable demand of many individual products from the FGI into one demand on manufacturing.

The next two applications are pooling related to capacity buffers. Work-sharing simply refers to a flexible workforce. Chaining is pooling applied to machinery, production lines or even whole plants. For example a series of assembly lines that are capable of assembling some portion of another assembly lines products are “chained” together. This way when one assembly line has problems another line can fill in.

4. PUSH AND PULL SYSTEMS

The terms push and pull refer to the way that jobs are authorized to move through production. The analogue is that jobs are either pushed or pulled through the plant. The terms originate from Taiichi Ohno and other practitioners of the TPS (Hopp & Spearman 2004: 140). Even though the terms push and pull have been popular in the management vocabulary since the 1980's there has been a lot of confusion of their exact definition (Bonney, Zhang, Head, Tien & Barson 1999: 53; Hopp et al. 2004: 133). Hopp et al. (2004: 140) point out that to begin with the terms push and pull were used in a vague manner. In essence there has been no widely recognized definition for push and pull.

However there is consensus on the archetypes of push and pull. These are Material Requirements Planning (MRP) for push and kanban for pull (Hopp et al. 2004: 136, 140; Burbidge 1996: 153, 155). With MRP a schedule is created based on expected demand and the expected capacity of production. Then based on the schedule, jobs are released, or "pushed" into production (Hopp et al. 2011: 369).

With a kanban system a maximum number of jobs per buffer stock are defined. Under no circumstances are we allowed to exceed the determined amount of jobs per stock. When a job from a buffer stock is removed a void is created in the buffer for the preceding workstation. The void signals or "pulls" a job from the preceding workstation to the buffer stock (Hopp 2008: 96–97). Bonvik, Dallery and Gershvin (2000: 2845) give a concise definition of kanban: "In its simplest form, kanban control reduces to each machine in the system having a finite output buffer, which the machine attempts to keep full". Figure 8 shows an example of a five station kanban system.

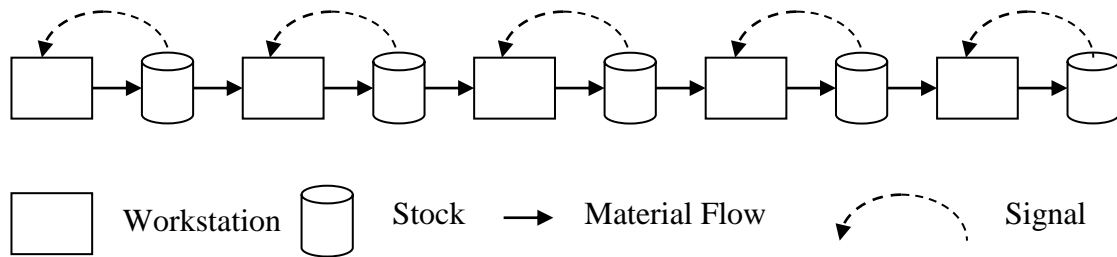


Figure 8. A kanban system. (Hopp 2008: 106)

4.1. Definition of push and pull systems

Hopp et al. (2004: 142) note that with the limited buffer sizes of kanban, an upper limit or a cap is created for the WIP of the system. Further they argue that in fact the WIP cap is the essence of pull. In other words a pull system is one where the status (WIP-level) of a system determines further releases of jobs in to the system. Conversely this implies that push systems are ones where the WIP-level is not limited.

In their article on **constant work-in-process (CONWIP)**, Hopp, Spearman and Woodruff (1990: 879) define push and pull: “For, our purposes, push systems will be those where production jobs are scheduled. Pull systems, on the other hand, are those where the start of one job is triggered by the completion of another.” Burbidge (1996: 153) gives effectively the same definition with a slightly different perspective:

These two classes divide ordering systems into those which issue orders for completion by specific due-dates based on estimated lead times (push systems), and those which seek to maintain a selected inventory level by immediately replacing any issues from stock (pull systems).

If we consider the definitions above critically it becomes clear that in practice pure pull or push doesn't exist. There will always be a set of circumstances where the assumptions of pure push or pull will be violated. In fact all practical systems are hybrids of push and pull (Hopp et al. 2004: 143). Consider a push system where

capacity has been overestimated. Release rate exceeds throughput. In a pure push system WIP would grow indefinitely. In the real world, at some point the management notices the excess WIP and schedules overtime, cancels jobs or slows down the rate of releases. By doing one or more of these arrangements the management introduces features of pull in to the system. In fact if we were to try to implement a pure pull or push system, the result in the long run would surely be bankruptcy.

Even though all practical systems function as hybrids of push and pull, it is still feasible to divide the basic operation mode of a system as push or pull. For example a production line where a WIP cap is set explicitly and the accordance to that cap is enforced in most situations, can be called a pull system. On the other hand, a production line where a WIP cap is not set explicitly or the cap is ignored can be called a push system. Let us use the terms push and pull with this relaxed approach, but also take advantage of the concepts of pure push and pull as defined by Hopp et al. (1990: 879).

4.2. CONWIP

If we accept the definition of push and pull by Hopp et al. and Burbidge then the simplest form of pull is the protocol called CONWIP (Hopp et al. 2011: 363). The motivation behind CONWIP is to introduce a pull method without the disadvantages of kanban. Although kanban is regarded as essential to the success of TPS, it requires the definition and maintenance of multiple parameters, as all the buffer sizes included in the kanban system need to be set exclusively. Further, on a production line with varying product mix the optimal buffer sizes may change rapidly depending on the product mix. All this amounts to kanban being inflexible and therefore suiting best for repetitive manufacturing (Hopp et al. 2011: 373-375). With CONWIP WIP will naturally accumulate in front of the busiest workstation which is where we want it (Hopp et al. 2011: 376).

With CONWIP we select a production line or a set of consecutive workstation and set a

WIP cap for those workstations and their buffers. This area is then called a CONWIP loop. Now as a job finishes at the end of the line a signal triggers a new job to be released at the start of the line. The workstations inside the CONWIP loop operate in push mode, but the system operates as pull (Hopp 2008: 102). Figure 9 shows an example of a five station CONWIP system.

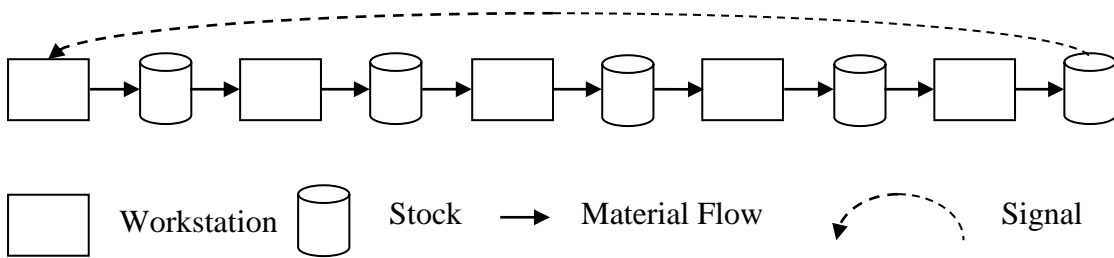


Figure 9. A CONWIP system. (Hopp 2008: 106)

It should be noted that in practice if there is no viable orders to release to the CONWIP loop then naturally releases are held of even if a signal is received to release more jobs. Then as suitable orders become available they are released immediately until the WIP cap is full.

4.3. Benefits of pull

In sorting out the benefits of pull a CONWIP system is considered, as it is the simplest form of pull. For a push system we consider a simple schedule of releases that are based on the estimated capacity of a production line.

4.3.1. Pull systems have less congestion

Spearman and Zazanis (1988: 524–525) show that pull systems have a smaller mean

cycle time. This is largely due to the ability of pull systems to work ahead, whereas a push system is required to stick to the schedule. They also conjecture that a pull system has smaller cycle time variance, which is caused by a negative correlation in the amount of WIP at different workstations. With push system there is no correlation at all. In other words with a pull system a high amount of WIP at one workstation implies a low amount of WIP in other workstations. Some of the benefits of these results are:

- Cycle times are easier to predict.
- Better OTD.
- Shorter frozen zone, that is, more time to introduce engineering changes to the products.
- Smaller WIP and finished goods inventory, and thereby, less inventory investment, less exposure to damage and a smaller requirement for storage space.

4.3.2. Pull systems are easier to control

An important corollary to the definitions of push and pull used is that, a pull system controls WIP and observes throughput while a push system controls throughput and observes WIP. A pull system controls WIP by setting a WIP cap, while a push system controls throughput by setting the rate of releases. Controlling WIP is inherently easier as it can be observed directly. Whereas controlling throughput requires capacity to be estimated. This is difficult as it requires the estimation of a multitude of factors, such as worker absenteeism, machine breakdowns and rework. (Hopp et al. 2011: 369).

The most important benefit of pull systems, as stated by Hopp et al. (2011: 372), is the robustness in the WIP cap, compared to the robustness of the input rate. In other words the results of setting the WIP cap sub-optimally are much less detrimental than setting the input rate sub-optimally. Gayru and Kleijnen (2001) emphasize the importance of robustness in production control systems: “a solution that is optimal for a given

scenario, is not practically relevant if that solution breaks down as soon as the environment changes.” Gayru et al. (2001: 452).

Roderick, Hogg and Phillips (1992) simulate different order release strategies under various shop conditions. They strongly recommend that CONWIP be considered by manufacturing enterprises and praise CONWIP especially for its robustness: “...there appears to be little doubt as to its robustness as an order release strategy.” (Roderick et al. 1992: 625-626). Spearman et al. (1988: 526-527) construct profit functions for CONWIP and a push system to illustrate the effect of robustness, see figure 10.

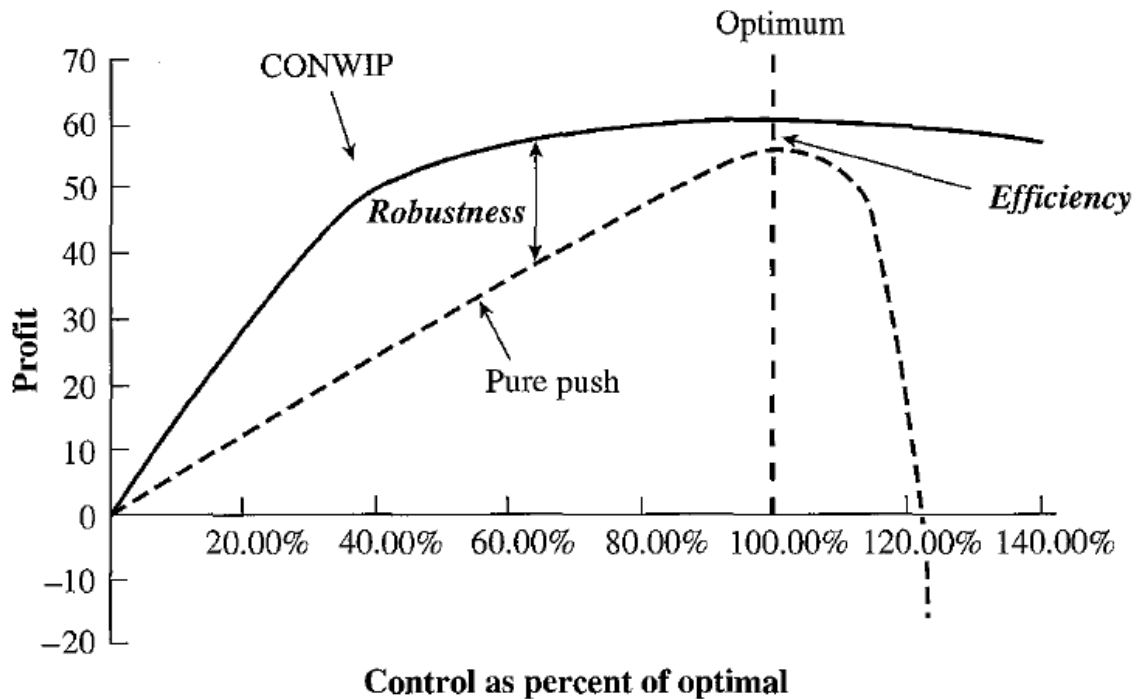


Figure 10. Illustration of the robustness of pull (CONWIP) versus push (Hopp et al. 2000: 358).

On y-axis we have the profit and on the x-axis we have the level of control parameter as a percentage of the optimal level. We should first note that there is a gap in the profit of optimal push and CONWIP levels. This is a result of the ability of CONWIP to work ahead. If this ability were to be denied, the gap would disappear. As we can see setting the input rate (push system) 130% from the optimal yields a negative profit. This is

caused by the WIP level exploding as utilization approaches 100%. On the other hand, setting the WIP cap 130% from the optimal is still very close to the optimal profit level.

4.3.3. Pull systems facilitate improvement measures

By setting a WIP cap the tolerance for inefficient operations is lowered. High WIP level has the effect of hiding quality problems, long setups, etcetera. With high WIP we have the opportunity of ignoring problem cases and just grab the next job to work on. This phenomenon is widely described with the analogue of a lake with rocks of various heights on the bottom. The level of the water in the lake represents WIP level and the rocks represent various problems in production. As WIP level is lowered we see the problems clearly and are forced to deal with them. Otherwise we will crash into the rocks, so to speak. Lower amount of WIP also implies shorter queues, which leads to a shorter time interval between the creation and detection of a defect. (Hopp et al. 2004: 137–138.)

4.4. Applying CONWIP

Hopp et al. (2011: 490) list three conditions that need to be fulfilled for a CONWIP system to work well. First condition is that part routings need to be set appropriately into individual CONWIP loops. In other words we construct parallel CONWIP loops when necessary. Differences of routings inside a CONWIP loop will translate into variability which causes all the pitfalls associated with variability. On the other hand constructing a CONWIP loop for every discernible routing inside a plant will make for a very complicated and high maintenance system.

Second condition is that a CONWIP loop should not be too long. A long CONWIP loop requires a large WIP cap which in term causes the loop to begin to behave as a push system. With a long loop and high WIP the WIP can accumulate into sections which

cause the WIP to be unavailable for the rest of the loop. These “WIP bubbles” defeat the purpose of CONWIP by disrupting the flow of materials. A second reason why a CONWIP loop should not be too long is that a CONWIP loop becomes difficult to manage if it spans over more than one managerial field. Hopp (2008: 106) also implies that communication inside a single CONWIP loop is important. Therefore cutting a CONWIP loop might be appropriate where communication might get compromised, for instance between workstations with long distances separating them.

The third condition is that there must be a reasonable measure of WIP. WIP can be measured in many ways: jobs, parts, and money, weight, length and work hours at the bottleneck. The best choice is the one that gives the most consistent measure of load on to the system and the one that is the simplest to use.

4.4.1. Effect of parallel routings in the same CONWIP loop

Let us hypothesize on the effects of parallel routings in the same CONWIP loop. In figure 11 we have an example of a CONWIP loop that operates effectively. Imagine that the loop has a WIP cap of 12. Suddenly the second workstation starts to collect most of the WIP. Let's say it has 8 WIP and the first workstation has 4. This can be caused by, for example machine breakdown, operator unavailability or a product mix that is challenging for the second workstation. Now we are losing capacity in the third workstation as it has zero WIP. What do we do? Ignore the WIP cap and release more WIP into the loop? Certainly not, as it would give us absolutely zero benefit. The queue would only increase in front of the second workstation and this would lead to all of the problems of excess WIP, enlarged frozen area and so on. We can conclude that the WIP cap is doing its job.

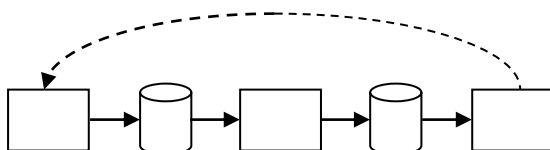


Figure 11. An example of an efficient CONWIP loop.

Now in figure 12 we have a CONWIP loop that operates ineffectively. The workstations in the middle are parallel workstations with inflexible capacity, meaning that we cannot move jobs of workforce between them. Now what happens when there is a problem in one of the middle workstations? It collects most of the WIP. Now we are losing capacity from three workstations. What do we do? Ignore the WIP cap and release more WIP into the loop? Yes! We release more jobs to the CONWIP loop and are able to utilize the capacity of all the workstations. The only problem is that now we have a push system and with that, all the disadvantages associated with push systems previously mentioned.

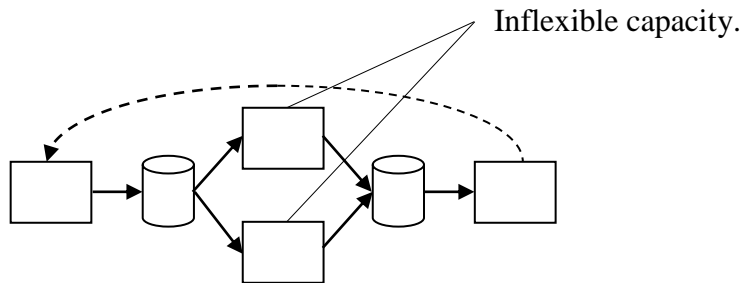


Figure 12. An example of an ineffective CONWIP loop.

How could we improve the ineffective CONWIP loop? We can construct separate CONWIP loops for the two routings. This way if one routing is operating poorly due to problems in one of the middle workstations, the other routing is still operating efficiently.

With multiple CONWIP loops it might not be feasible to include the last workstation inside the loop. To clarify this point let us consider a similar system to that in figure 12 except that now we have 10 parallel workstations in the middle and each has their own CONWIP loop. The last workstation has to have the capacity and the WIP buffer to be able to process the jobs from all 10 routings. Now what negative consequences might we have for including the last workstation inside to CONWIP loop? Well, one CONWIP loop might get overrun by the other CONWIP loops at the last workstation. By this I mean that a temporary increase in the output of the other CONWIP loops might cause the WIP of one loop to wait for an excessive amount of time and thus starving the beginning of the loop. On the other hand when the other loops would

eventually slow down; there would be more than enough capacity to process the jobs of just one loop. In essence the capacity and therefore the WIP of the last workstation, relative to one CONWIP loop is likely to fluctuate violently.

4.4.2. Multi-loop CONWIP

By cutting up a production facility into multiple CONWIP loops we create a multi-loop CONWIP system, see figure 13.

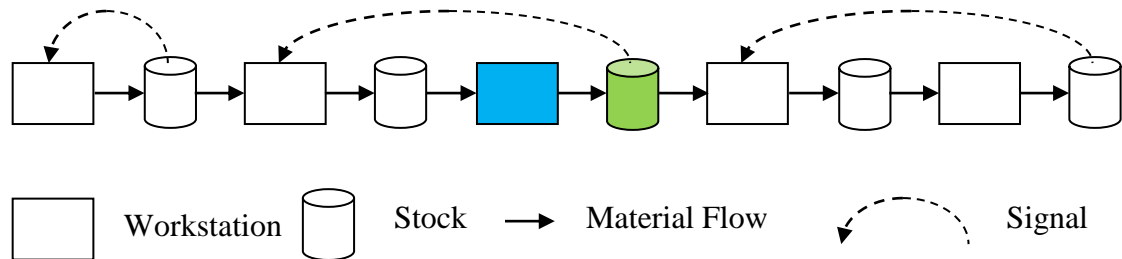


Figure 13. A multi-loop CONWIP system. (Hopp 2008: 106.)

If we compare a kanban system and a CONWIP system we notice that a multi-loop CONWIP system is just a hybrid of a kanban and CONWIP. In fact kanban can be considered as just a special case of CONWIP. On the other hand the developers of the TPS (and kanban) might have considered CONWIP as just a special case of kanban. In any case the two protocols are very closely related. (Hopp et al. 2011: 496, 502.)

There are two ways to construct a multi-loop CONWIP system. One way is to start the next loop right where the previous loop ended. The other is to exclude a part of the production line from any CONWIP loop. In figure 13 we see that the signals start from the buffer preceding the first workstation in the next loop. What happens in the middle loop if we start the signal from the blue workstation instead of the green buffer? The green buffer ceases to be a part of any CONWIP loop and therefore WIP is allowed to accumulate freely in the buffer. This is very useful if the bottleneck of the system is

located in the middle loop. (Hopp et al. 2011: 496–497.)

If the bottleneck is located in the middle loop and the signals work as in figure 13, then situations will arise where the bottleneck gets starved due to the last loop temporarily functioning slowly. On the other hand if the green buffer is not included in the middle loop then its WIP might temporarily rise high. Eventually the last loop will consume the WIP in the green buffer, because the bottleneck is in the middle loop and therefore the last loop has a higher capacity than the middle one. (Hopp et al. 2011: 496–497.)

4.5. Other production control concepts

There are many different concepts for production control such as CONWIP, kanban and MRP. For comparison three additional concepts are described: Drum-Buffer-Rope (DBR), Simplified Drum-Buffer-Rope (S-DBR) and Period Batch Control (PBC). Ultimately the best concept depends on the type production facility in question and the strategy of the company (Benders & Riezebos 2002: 502, 505).

4.5.1. Drum-Buffer-Rope

DBR is a production planning technique by Eliyahu Goldratt first applied in 1984 in the novel “The Goal” and formally described in 1986 in the book “The Race” (Goldratt et al. 2004; Goldratt & Fox 1986: 96–117). The purpose of DBR is to enable improved decision making and scheduling on the shop floor. DBR focuses on exploiting the bottlenecks of a system thus simplifying the problem of how to schedule a system (Schragenheim & Ronen 1990: 18).

In DBR the drum corresponds to a bottleneck workstation. Based on the capacity of the bottleneck (also called the drumbeat) a schedule is devised. The buffer protects the drum from variability (also known as: statistical fluctuations, disruptions or Murphy).

The buffers planned size is determined explicitly. In DBR the size of the buffer is determined in time, for example a three day buffer. This can be easily converted into a number of jobs, for example a drum with the capacity of two jobs per day: a three day buffer converts to six jobs. (Goldratt et al. 1986: 96–117.)

In DBR language a rope is tied from the drum to some workstation preceding the drum. This is the workstation in which jobs are released based on the drumbeat. The rope is a mechanism that forces the release rate to the system to be at the rate of the drumbeat. The rope is implemented with a detailed schedule (Schrageheim et al. 1990: 18). Non-bottleneck workstations, however, are not bound by a definite schedule. This gives them flexibility to work on available jobs. The important point is not to release more jobs if the rope does not allow it, even if the first non-bottleneck workstations are idle (Schrageheim, Cox & Ronen 1994: 1873). Figure 14 illustrates a simple DBR system.

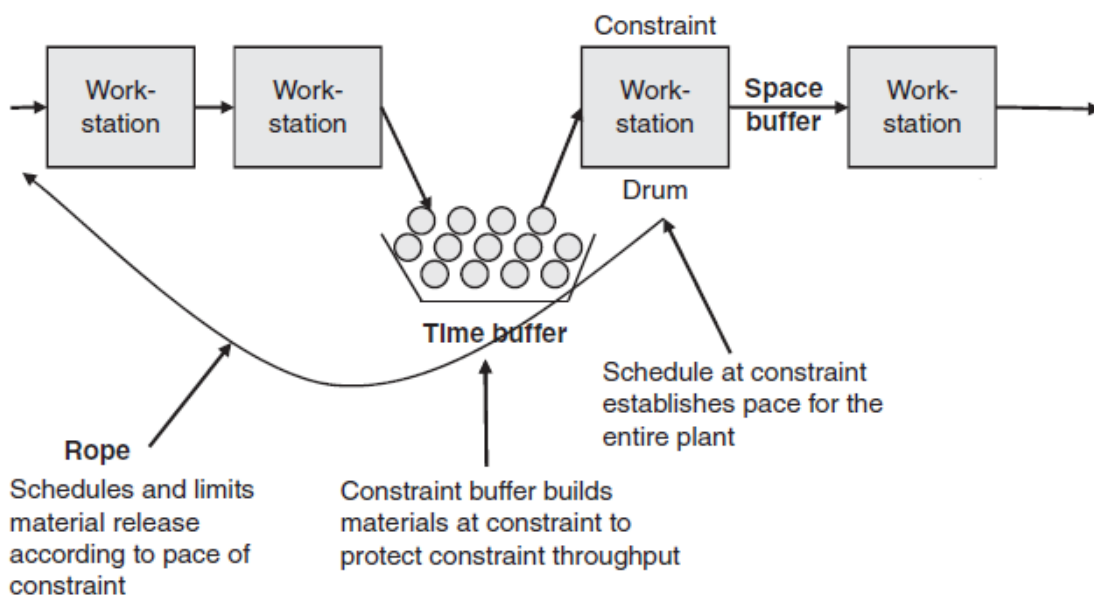


Figure 14. A simple DBR system. (Cox III, Goldratt & Schleier Jr. 2010: 186).

As DBR controls the input rate of a production line, not the WIP level, it is closer to a push system than a pull system. However, it is obvious that the schedule requires to be maintained in regular time intervals. Goldratt et al. (1986) does not give explicit

instructions on how often the schedule should be set, but as an example uses the time frame of one week (Goldratt et al. 1986: 118). Schragenheim et al. (1990: 18) use a time frame of two weeks in their simulation study. As the time buffer preceding the bottleneck is given special attention it is clear that the input rate, which we decide for the next time frame, depends also on the buildup of inventory in front of the bottleneck. In other words the level of WIP is controlled, albeit in an indirect manner. The rigor of WIP control depends on the frequency that the schedule is set. In fact, it is possible for DBR to resemble a CONWIP system with bottleneck as the last workstation (Hopp et al. 1990: 888).

A problem with DBR, when compared to CONWIP, is the sensitivity to errors in determining which workstation is the bottleneck (Hopp et al. 1990: 888). It is easy to imagine that if the bottleneck is not determined correctly, the WIP level will explode, which will lead to all the problems associated high WIP. This is further exacerbated in production lines that have a “floating bottleneck”, that is, the bottleneck of the line changes depending on the product mix (Hopp et al. 2011: 375, 564). Goldratt et al. (1986: 106) recognize the problem of a floating bottleneck, but do not give a method for DBR to manage it. Another problem in DBR is the same that all push systems have: the output rate must be estimated which is sensitive to errors as seen on figure 10.

4.5.2. Simplified Drum-Buffer-Rope

S-DBR is a modification of DBR intended for production plants that have their production rate limited by market demand (or sales) instead of production capacity. Consequently the drum of S-DBR is the market demand. When an order comes in, a quick check on the load of production is performed and if it is determined that the load on production is not too high, the order is released immediately for processing. When production is too heavily loaded, short term measures to reduce loading—such as overtime of additional shifts—are required in order to maintain a high level of OTD. The only buffer explicitly maintained is the shipping buffer as it precedes the drum.

(Schragenheim & Dettmer 2000: 4–7.)

S-DBR is a simple approach to subordinate production to the market demand when large amounts of excess capacity are available. The weak point however seems to be the limited applicability of this approach. Most manufacturing plants can easily take advantage of their excess capacity simply by reducing delivery times. In most markets shorter delivery times increase demand. Both, increased demand and shorter required cycle times, cause production to become a greater limiting factor. For many companies the only situation where market is clearly the main limiting factor is during a recession.

4.5.3. Period Batch Control

PBC is a classic production control concept most famously applied for the Spitfire aircraft manufacturing during 1940–1941. In a PBC system production and order releases are divided into periods of the same length. Each phase (for example a set of workstations) must attempt to complete the work allocated to that period. This means in general that after each period the amount of work released is the same amount that was finished during the previous period. Therefore the throughput time will become the length of a period multiplied by the number of phases. PBC aims at short throughput times and effective scheduling. (Benders et al. 2002: 498–500.)

The use of standard periods leads to a very simple and transparent part ordering, job release and scheduling policy. Every shop floor worker and purchasing personnel knows clearly what must get done before the next period begins. When some workstations are lagging behind, workers from other stations can offer assistance so that the work allotted to the current period gets done in time. This type of policy serves to reduce WIP buffering and causes late jobs to be automatically prioritized. (Benders et al. 2002: 500.)

PBC requires that the production system that it is being applied to can be divided into a

suitable number of periods. A system that cannot be divided to more than one phase is not suitable at all. A system with a very high number of phases is also not suitable for PBC (Benders et al. 2002: 503). Benders et al. (2002: 503–504) elaborate on a case company—a furniture manufacturer—that uses PBC. The case company has a period length of one week and has divided their processes into six phases.

PBC is best suited for production facilities systems that are organized in teams that are responsible for multiple work phases (also known as cellular manufacturing). For PBC to function optimally, the teams involved should take approximately the same time to finish their part of the manufacturing process (Burbidge 1996: 156).

5. CASE: ABB OY, MOTORS AND GENERATORS VAASA

ABB Oy, Motors and Generators Vaasa manufactures electric motors for industrial purposes. In weight the motors vary approximately from 20kg to 5000kg and in watts 0.25 - 1000 kW. In 2011 the case company produced approximately 35000 motors. The products produced are mostly MTO, that is, the jobs in production are based on customer orders. The main components in electric motors are: stator, rotor and the frame. An essential part of the stator is the winding. The work phase where winding is done is also called winding and it is one of the most demanding work phases of motor manufacturing in terms of processing time and skill required from the workers. See figure 15.

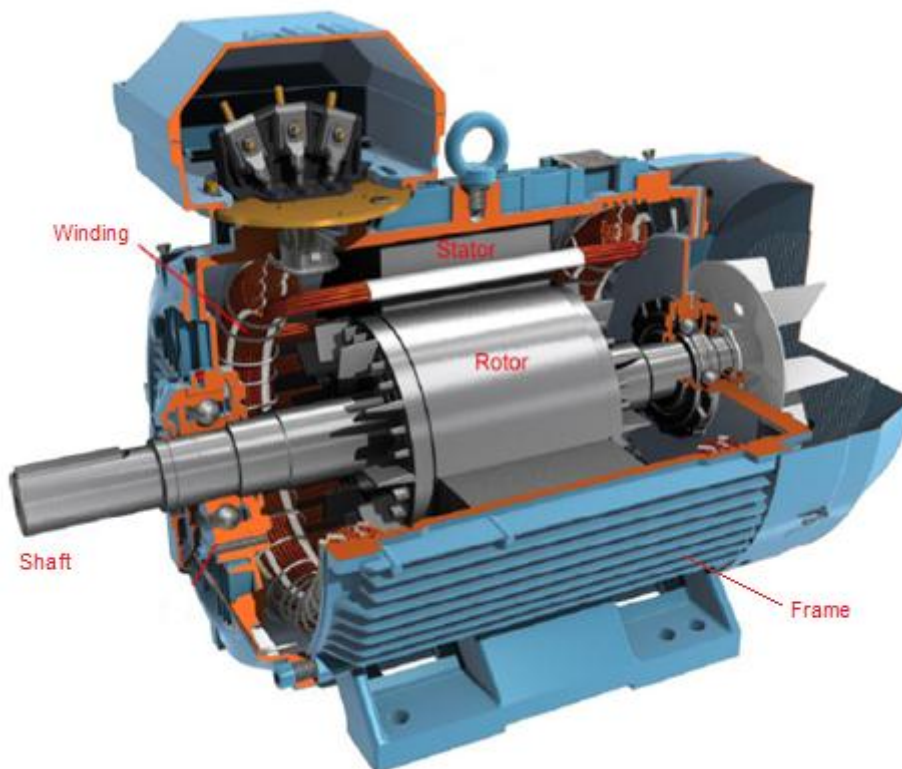


Figure 15. The main components of an electric motor (ABB Oy, Motors and Generators 2012).

5.1. Outline of the order fulfillment process and production

The case company's production consists of component manufacturing and assembly operations, which are organizationally separated. The component manufacturing part of the plant is called the "component factory" and the assembly part of the plant is called the "assembly factory". The assembly of the case company consists of seven assembly lines which are located physically in different locations and assemble different frame sizes. The lines are called AL10, AL15, AL30, AL35, AL40, AL50 and AL55. Frame sizes are measured based on the shaft height of the motor in millimeters. See table 1.

Table 1. Frame size allocation in the assembly of ABB Oy, Motors and Generators Vaasa.

Assembly line	Frame size
AL10	80, 90, 112, 100, 132
AL15	160, 180, 200, 225, 250
AL30/40	280, 315
AL35/50	355, 400
AL55	450

The order fulfillment process is divided into five "gates". We will focus on the three of these: Delivery control, Assembly and Dispatch. These are the gates that are involved in production. See figure 16 for the outline of the production.

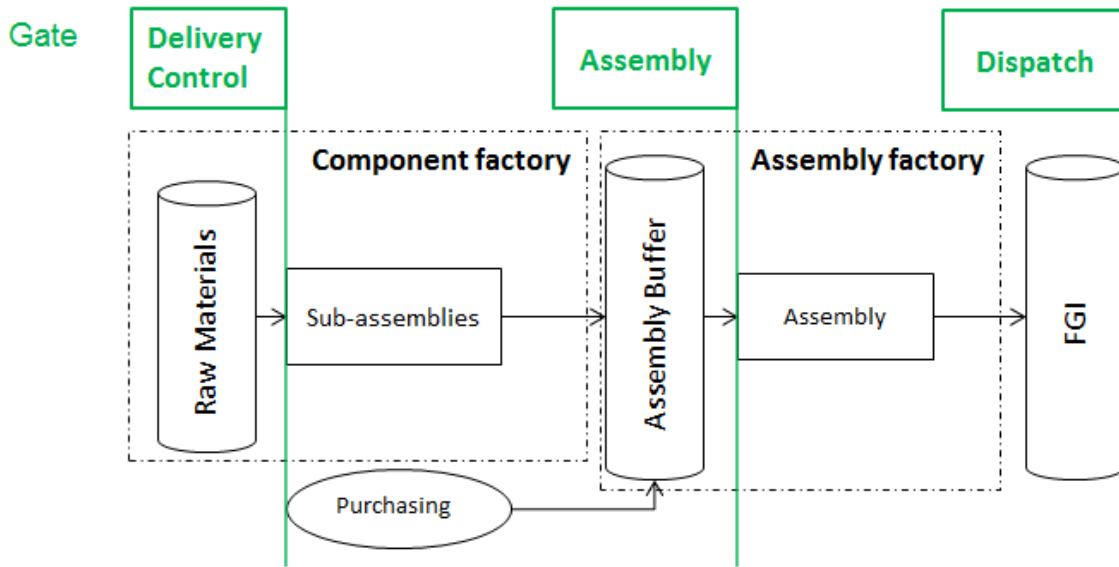


Figure 16. Outline of the case company's production.

For a job to pass a gate, a set of requirements—defined by the gate in question—have to be met. Passing the delivery control gate is synonymous to releasing a job to production. The delivery control gate requires that, for instance a job must have its application engineering done and the planned release date must not be too far into the future. When a job is released, the start of component manufacturing is authorized and purchased parts are ordered. There are three different sub-assemblies in component manufacturing: frame machining, rotor manufacturing and stator manufacturing.

The start of assembly is authorized when all of the components that are manufactured in sub-assemblies and all of the purchased components are available. Finally when the customer order is ready to be shipped in a way agreed with the customer, the product is shipped from the FGI by dispatching. See figure 17 for the process chart of the production.

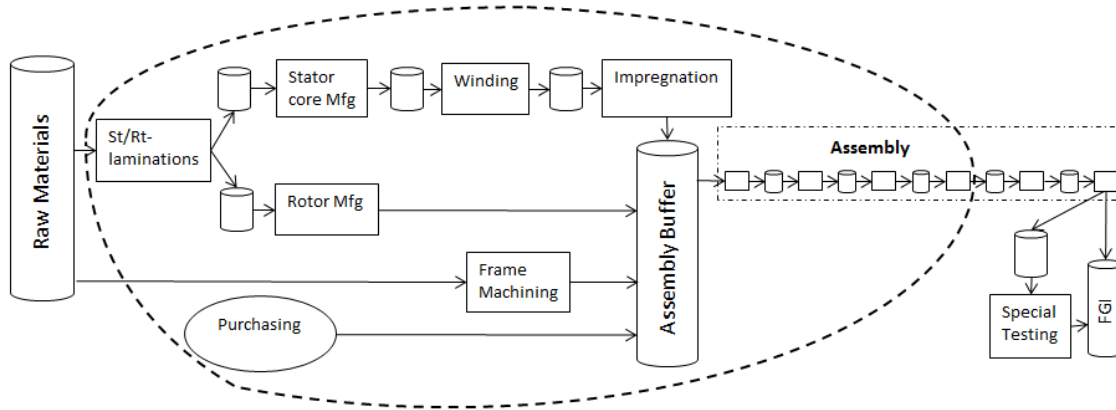


Figure 17. A process chart of the case company's production. The area inside the dotted line is included in a CONWIP loop.

Note that for simplicity the above chart shows only one of seven assembly lines. Similarly in reality there are six winding locations, some of which are located geographically far away from the case company's plant. Five of these winding locations are located in the premises of various sub-contractors. Additionally the types of windings done at each winding locations varies to some extent. The details of the assembly workstations are left out as they are not important for our investigation. Suffice to know that the assembly operations are physically close to each other and the people working in one assembly line are familiar with each other and are often able to work in more than one workstation inside the assembly line. Therefore an assembly line can be considered as a cohesive whole.

5.2. Job releases, DBR and CONWIP

According to the case company's production control guidelines the main principle used in production control is the theory of constraints (TOC)—particularly drum-buffer-rope (DBR), which is a part of TOC. Also the concept of CONWIP is used. A comprehensive enterprise resource planning system (ERP) is used to control the day-to-day parameters. The ERP system is used to release orders to production and it maintains

most of the scheduling done during the order fulfillment process.

The execution of the delivery control gate is performed by the department called delivery control. The input of orders into delivery control is determined by factors such as customer demand, schedule of confirmed customer orders and processes before delivery control, such as application engineering. The jobs are released into production based on a CONWIP loop and the schedule of confirmed customer orders. Each assembly line has its own CONWIP loop. This means that there are in total seven CONWIP loops. The WIP counted for a CONWIP loop consists of the WIP inside an assembly line up to the fourth workstation and the WIP in the component factory that is destined for the assembly line. See figure 17.

With a CONWIP loop in use for all the assembly lines, we are left to ponder where exactly is DBR used? A description of DBR in the production control guidelines of the case company shows an alternative way of implementing the “rope” with a CONWIP loop. This idea is not mentioned in the TOC or CONWIP literature; see for example (Cox III et al. 2010; Goldratt 1986; Goldratt 1990; Hopp et al. 2011; Hopp et al. 1990; Schragenheim et al. 1990). In the literature the rope is consistently described as a detailed schedule which determines when jobs are to be released (Cox III et al. 2010: 189; Goldratt 1986: 100–104; Goldratt 1990: 222–226; Hopp et al. 1990: 888; Schragenheim et al. 1990: 18; Schragenheim et al. 1994: 1872). Further, if we try to implement the rope with CONWIP that would just be the equivalent of a plain CONWIP system. It seems like the case company should clarify these concepts in order to avoid confusion and misunderstanding.

If we have the authorization by the CONWIP loop to release more jobs to production, the release of jobs up to two weeks ahead of schedule is allowed. There are some important reasons to limit job releases not to include ones that are too much ahead of schedule. For example, the customer may want to change their order, which is much more costly to do when the production of the job has already begun. Another reason is to limit the FGI, since if we start a job early, then the job will tend to finish early, which in many cases means that the job will wait an excessive amount of time in FGI. Third

reason is that by starting a job early, we might be stealing capacity from a job that is already late (a “hot job”).

By limiting releases to ones that are not too much ahead of schedule we ensure that we do not have a pure pull system. But again, as mentioned before pure pull or push has no place in a practical system. However, work is released regularly even though the WIP cap is full. This happens when the WIP cap is already full and the schedule indicates that the time of release for a job is today or sooner. Additionally, often even when the WIP cap is full and there are no scheduled releases for today or sooner, jobs are still released. In essence we have a push system.

With an appropriately set CONWIP configuration the WIP cap should not be exceeded. In fact the performance of the system suffers from releasing more jobs when the WIP cap is full. Consider a workstation that already has a long queue. Releasing more jobs to the end of the queue will only inflate cycle time and introduce all the other negative effects associated with high WIP, with no added benefit. Additionally often the WIP level regularly stays well below the WIP cap not reaching the cap for months. This leads us to suspect that the WIP cap is set to a very high level. Effectively, most of the time WIP is not limited, that is, we have a push system.

The fact that the WIP cap is not followed and that it is very high, leads us to suspect that there is room for improvement. There are a few possible reasons that the WIP cap is not adhered to. First, the system was never meant to be strictly a CONWIP system, that is, there were never an intention to strictly maintain a WIP level at or below the CONWIP limit. Second, the concept and benefits of a WIP cap are not understood. Third, the CONWIP loop is not set in a suitable way. For example, it includes workstations that should not be included or it includes too many workstations inside the same loop. In a dysfunctional CONWIP system we might very well benefit from releasing jobs disregarding the WIP cap of the CONWIP loop. Whatever the reason, CONWIP still seems to be a suitable method for the case company and therefore let us investigate how the CONWIP loops are set up in the case company.

5.3. Problems in the current CONWIP loops

First we observe that the CONWIP loops extend over four organizationally different entities. These are the component factory, purchasing, winding operations done by sub-contractors and the assembly factory. This makes the loop inflexible and difficult to manage as the needs of one department (for instance the assembly factory) cannot be addressed without affecting other departments. Delivery control releases jobs based loosely on the CONWIP loop, but after that they do not have any natural interaction with the status of production. It seems like there is no natural owner for the CONWIP loop. Hopp (2008: 106) mentions that it generally makes sense to cut a CONWIP loop between workstations where the workstations are under separate management or the workstations are physically distant from each other.

Second the CONWIP loop is very long. One way to evaluate the length of the loop is to see if it extends over many organizational areas and as we noted above this is the case. Another way is to check the physical distances inside the CONWIP loop. And indeed with many product families the main components travel multiple times between different plants while still inside the same CONWIP loop. For instance stator manufacturing may start from plant 1, be shipped to a winding sub-contractor, then shipped to plant 2 for impregnation, then shipped back to plant 1 for assembly; all this while still in the same CONWIP loop. As previously mentioned a long CONWIP loop leads to behavior similar to a push system.

Third the release of jobs is divided based on assembly lines. This can cause problems in component manufacturing as there is no separation of workstations by assembly line in component manufacturing. Now we have situations where one assembly line is behind on schedule and another one is ahead of schedule. This means that for one assembly line jobs are released ahead of schedule and for another behind schedule. Now situations will occur where the early releases of one assembly line steal the capacity of the component factory from the assembly line that is behind on schedule.

Fourth we observe that there are parallel inflexible workstations inside the same CONWIP loop. This means that there are multiple parallel routings in the same CONWIP loop. The winding operations are parallel as one CONWIP loop contains more than one winding location. They are inflexible as it is costly and laborious to shift WIP from one winding location to another. Further the capacity is inflexible, that is, we cannot move the workers from one winding location to another.

The load on winding operations is addressed by manually balancing the load during job releases. The downside of this solution is that it complicates the job release process and makes it more laborious. Also with daily manual balancing it is easy to make mistakes resulting in sub-optimal loading. Even though attention is given to balance the load on winding operations, it does not mean that the releases of jobs would be limited when there is high load on winding. It only means that there is an effort to have the same load on different winding locations. Eventually if the load on a single winding location gets too excessive, delivery control is informed—typically by e-mail—not to release any more jobs for that location for a period of time. This is exactly how push systems work in practice. The problem is of course that by this time the damage has already been done. In essence the production control system in use does not give sufficient support for limiting and balancing the load on winding locations.

Another set of parallel workstations is located before the assembly buffer. Here in effect we have four parallel workstations all under the same CONWIP loop: stator manufacturing, rotor manufacturing, frame machining and purchasing. All of these also have inflexible capacity and we cannot move work from one workstation to another. Here we should note that if assembly (or some part of it) and component factory are under the same loop then all of the sub-assemblies are automatically in the same loop, as assembly cannot be started before all the components are available.

5.4. Suggestion for an improved CONWIP configuration

The following alternative CONWIP configurations are a step-by-step improvement suggestion for the performance of the case company's production by offering solutions to the problems presented above. Visually the biggest difference in the suggestions and the current configuration is that there are separate loops managing the component factory and the assembly factory. The loop for component factory should be considered the more important as it defines job releases into production. After a job is released it becomes a physical entity, which is much more difficult and costly to manage than a job that is still only an order number in the ERP system. The CONWIP loop for assembly will be the same for all the suggestions. For the assembly CONWIP loop see figure 18.

5.4.1. CONWIP loop in assembly

The purpose of the assembly CONWIP loop is to limit WIP inside an assembly line, which will decrease cycle time inside assembly. This will be beneficial when customers want to make changes to their orders. It is more likely that we have not yet begun the assembly of the job when we get the information of the change, which makes the change easier to execute and we have not wasted capacity on assembling a job that is not in accordance with the customer's wishes. Second, we have the ability to process "hot jobs" faster when they arrive to the assembly buffer. This is due to the fact that we would be maintaining smaller queues inside the assembly and so the queuing time of the hot job is reduced.

Third, in the beginning the overall cycle time is not affected as queuing time is merely transferred from the assembly operations to the assembly buffer. But with time, as the WIP cap of assembly is lowered, the assembly buffer grows larger from the WIP that used to be tied up in the assembly operations. This means that we can reduce the overall assembly buffer size an appropriate amount which in term speeds up the cycle time of the whole company.

A factor that goes hand in hand with a CONWIP loop covering assembly is the

flexibility of the workforce. A situation is likely to arrive where the worker at the first workstation is unable to start a new job due to the WIP cap being full. When this happens the worker has the option of finding out where all the WIP is and go help out in that workstation. This kind of activity is completely feasible inside the assembly operations and it should be encouraged. In effect the CONWIP loop forces workers and foremen to react faster to accumulating WIP inside assembly. Now when jobs accumulate in front of a struggling workstation, immediately more resources will be assigned to that workstation to remove the accumulated WIP bubble.

The assembly buffer is not a part of any CONWIP loop. Therefore it must be monitored separately. If the buffer grows too large, capacity of assembly must be increased or releases to production must be reduced. The easiest way to increase capacity is simply to schedule overtime. The way to limit releases would be to define a smaller WIP cap for the first CONWIP loop. This would have the added benefit of speeding up cycle time in component factory.

5.4.2. Making shorter loops

A logical way to reduce the length of the current CONWIP loops is simply to separate the assembly factory and component factory into their own loops. See figure 18. In principle this configuration is still very similar to the current configuration as production releases would still be executed based on the load on assembly lines. In order to implement the first loop in figure 18 no other actions are required than having a job removed from the WIP count of the CONWIP loop, when all the parts of the job are available in the assembly buffer.

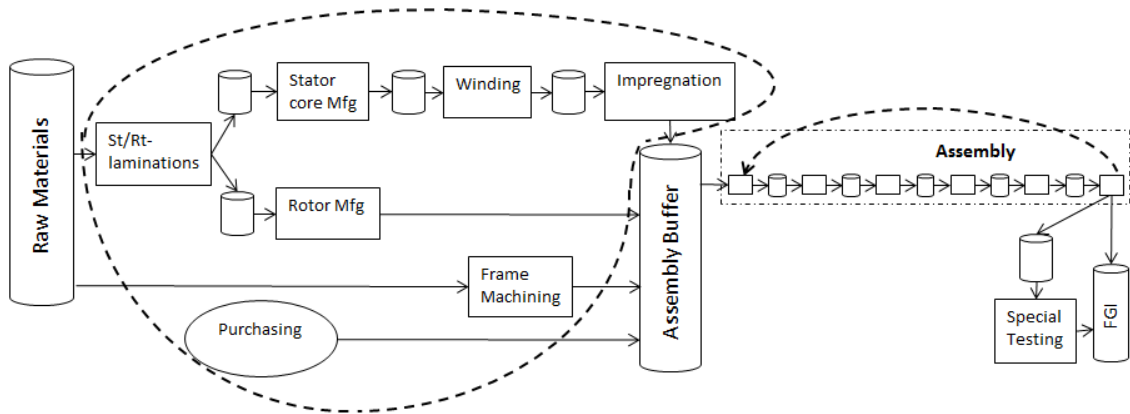


Figure 18. Process chart of the case company's production with separated loops for component factory and assembly factory.

This system addresses only two of the problems previously listed: the length of the loop and the multiple organizational areas inside one CONWIP loop (assembly is separated). Therefore further modification is required.

5.4.3. Reducing parallel routings preceding assembly buffer

The next phase is very similar to the previous system as the releases are still based on assembly line load. The difference is that rotor manufacturing, frame machining and purchasing are now subordinated to stator manufacturing. This means that the pace of stator manufacturing determines the pace for all of the components. See figure 19.

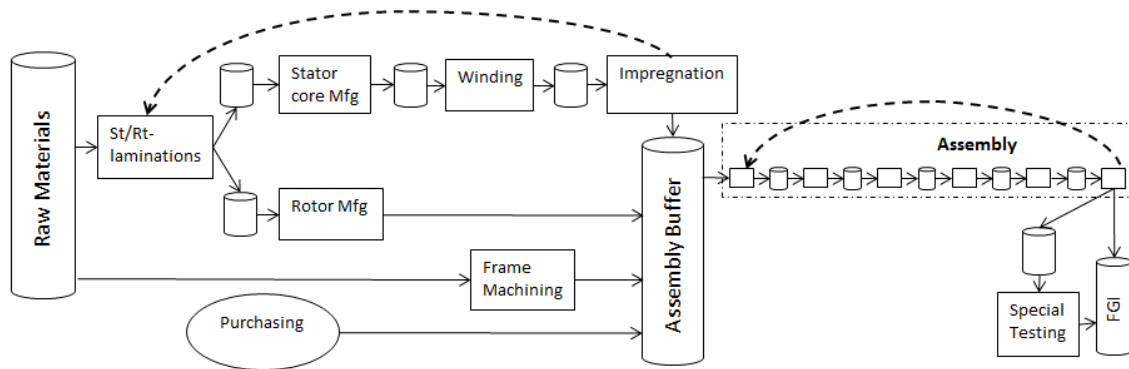


Figure 19. Process chart of the case company's production with a CONWIP configuration where releases are based on the status in stator manufacturing.

Now we are addressing three of the problems previously listed: parallel routings in the same CONWIP loop (rotor manufacturing, frame machining and purchasing removed), the length of the loop and the number of organizational fields in one loop (purchasing and assembly separated). It does not however address the inflexible workstations for winding and the release of jobs based on assembly lines.

5.4.4. Releases based on winding locations

The following modification will address all of the problems previously listed. Shortly put the suggestion is to release jobs based on the load on winding locations, instead of load on assembly lines. The challenge in this option is that its implementation requires the most effort, considering that currently the ERP system is configured to release based on assembly lines instead of winding locations. Also out of the suggestions this would be the biggest change relative to the current system. Therefore it might cause the most opposition due to inertia. On the other hand implementation should be facilitated by the fact that the load on winding locations is already balanced in the current system. So in other words the suggestion is to forget the current CONWIP loop and elevate the mechanism of load balancing for winding locations. By elevation I mean that we not only balance the load, but also limit the load with the CONWIP protocol. See figure 20.

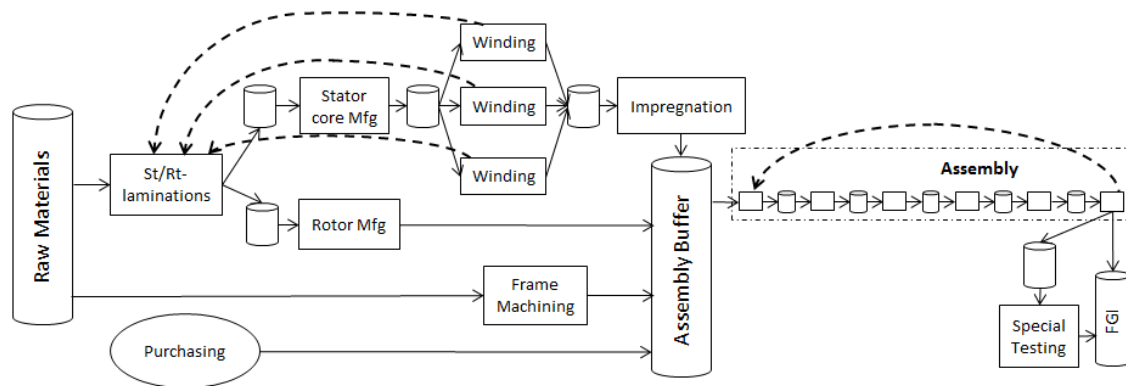


Figure 20. Process chart of the case company's production with winding based releases. Here we have additional winding locations illustrated to clearly demonstrate the suggestion.

To clarify the ramifications of this system, let us recall that there are six different winding locations and seven different assembly lines. This means that we have a separate CONWIP loop for all of them. Impregnation is not included in the first CONWIP loops as it needs to accommodate input from all of the winding locations and therefore has a fluctuating capacity relative to one winding location. This means that the buffer of impregnations needs to be monitored separately by the management responsible of impregnation.

The cycle time requirement of rotor manufacturing, frame machining and purchasing is not considered to be as long as in stator manufacturing. Therefore the start of work in these operations is subordinated to the start of stator manufacturing. In essence we assume that rotors, frames and purchased parts are generally ready before stators. This means, for example that the average WIP in rotor manufacturing must be smaller than the sum of the WIP in the first CONWIP loop and impregnation.

5.4.5. Intended benefits of the new configurations

The most important benefit that we should achieve with the modified CONWIP

configuration is to have shorter and more stable queues for all of the workstations inside production, without sacrificing throughput. This will cause shorter cycle time for production and for the whole company. More stable queues will decrease cycle time variability, which will give us a better OTD. Another benefit will be a shorter frozen zone, that is, we have more time to implement changes in orders before production has begun, thus having less rework. Third, management should become easier: determining an appropriate WIP cap, using overtime and prioritization. See appendix 1 for details on the benefits of shorter cycle times and smaller cycle time variability.

5.5. Other approaches for improvement in the case company's production

Suri (2010: 1) states: "Everyone knows that time is money, but time is actually a lot more money than most managers realize!" He elaborates on this idea with the thesis that reducing lead times is the most effective approach for improving processes in manufacturing (Suri 1998; Suri 2010). The figure in appendix 1 gives us some clue why this might be—reducing delivery time (one form of lead time) does indeed have many positive effects. What makes reducing lead times possible are shorter average cycle times and lower cycle time variability. For an effective focus for improvement measures let us concentrate on reducing lead times for the following examples.

Some of the most effective ways to reduce lead times in manufacturing are:

1. Reduce queue time
2. Increase station overlap time
3. Remove unnecessary operations

5.5.1. Reduce queue time

Queue time can be reduced by increasing capacity buffering and reducing variability. A

guideline for increasing capacity buffering was discussed in chapter 3.3.3. Examples of opportune variability reduction targets are not difficult to find in the case company. The two types of variability that can be reduced are arrival variability and processing time variability. An example of reducing process time variability in the case company would be by increasing work-sharing, that is, a flexible workforce. This will alleviate the slow processing rate of workstations with temporary shortage in workforce, thus reducing the workstations process time variability.

A simple and effective way to reduce arrival variability is to make an effort to minimize variability in job releases. Long term release quantities are determined mainly by demand and capacity. Therefore we will focus on the release variability per day and per week. See figure 21.

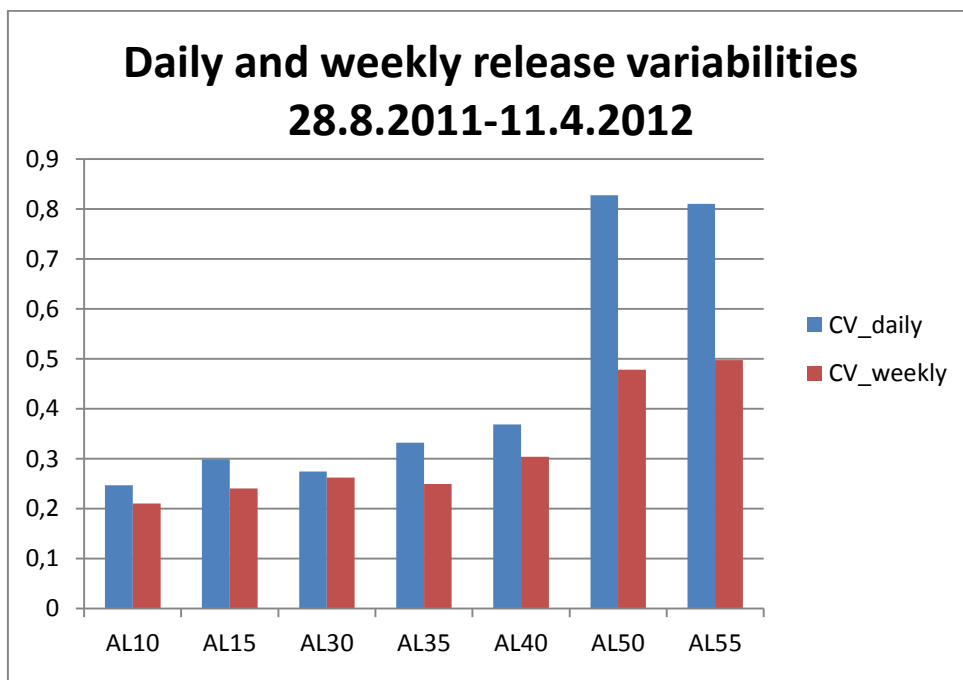


Figure 21. Variability in the daily and weekly job releases per assembly line.

We notice that assembly lines assembling bigger frame sizes have higher variabilities. The simplest explanation for this seems to be that the assembly lines assembling larger

motors have a much lower volume. For AL55 the average weekly volume is approximately 10 and for AL30 it is 170. With a low volume a difference of one job from the optimal can cause significant increase in variability. What we cannot determine from figure 21, is whether the variabilities are high or low. Regardless there is always room for improvement. One approach would be to give the personnel executing job releases the goal of reducing the values in figure 21. Job releases to production should be considered as a powerful tool to minimize the arrival variability in production. This is important because while the orders are still in a purely electronic form, the management of variability is much cheaper.

Another simple and effective method of reducing release variability can be found in AL55. The average daily production speed of AL55 is relatively very low: approximately two motors per day. This makes job releases very sensitive as most of the time three releases per day is too much and one job too little. The opportunity for improvement comes from the lot sizes. The maximum lot size in AL55 is two (although this is sometimes exceeded). In this context lot sizes refer to the number of motors under one order number. When an order is released all of its motors are released at the same time.

Often there is an order with a lot size of one queued up first and an order with a lot size of two queued up second. Effectively this means that we are forced to release either one or three jobs, even when two would be the optimal amount. Obviously this creates unnecessary arrival variability for the first workstation of production, which—as previously discussed—propagates to the rest of the plant. The solution is of course to reduce the maximum lot size to one. Another solution would be to modify the ERP system to allow orders to be partially released, for example only one motor from an order of five.

The method of releases could also be improved by having the ERP system execute them at the optimal time. Currently job releases to production are done manually once per day. With the various pull systems previously discussed, jobs are released immediately as the WIP cap drops below its maximum. With releases executed once per day this is

obviously not the case. A more effective solution would be for the ERP system to release jobs automatically. Instead of releasing jobs once per day, delivery control could give a set of orders permission to be released by the ERP system. This could be done once per day by delivery control. Now the ERP system would automatically release jobs when the WIP cap falls below its maximum level. We should note that this type faster replacement of jobs in the CONWIP loop becomes more important with a shorter CONWIP loop.

5.5.2. Increase station overlap time and remove unnecessary operations

Occasionally in manufacturing plants there are processes executed in tandem without a good reason. In other words a workstation is waiting for some processes to be completed that are not required for that station to begin processing. Possible reasons can be to facilitate poorly set up systems, a sub-optimal layout design or simply due to inertia. For instance in the case company the first two workstations of assembly do not require the rotor component. Nonetheless the current system is set up so that starting the assembly phase is not possible before the rotor is available. The first operation in assembly only requires the stator and the frame. Therefore a better solution would be to move this operation to the component factory in combination with impregnation. This would have major benefits: pooling of resources; faster detection of quality problems and smaller transportation and storage need.

The existence of unnecessary operations is caused by the same reasons as the processes unnecessarily executed in tandem. The largest portion of unnecessary operations is generally from transportations. In the case company there is vast amount of transportations between workstations that could be eliminated with a more effective layout planning and insourcing. Routings for jobs starting from stator production to the FGI commonly involve as many as four transportations by trucks between plants and subcontractors. The time that the motors have to spend in transportation and waiting for trucks, have a massive effect on cycle time. Another type of transportation, not

necessarily dependent on the layout and sourcing arrangement, are lifts from and to storage spaces. A worthwhile goal in production would be to minimize the amount of lifts per pallet or job.

6. FURTHER CONWIP DISCUSSION WITH SIMULATION

In order to gain further insight on the behavior of the CONWIP protocol a simulation study is performed. The purpose of the study is to give an additional perspective and examples from which to consider the best way to set a CONWIP configuration. The study will cover a few simple systems and therefore the result should be considered as examples rather than general truths. Four systems will be simulated. The first three systems model specific cases where different CONWIP setups may or may not produce varied performance. The fourth system is an attempt to simulate some basic factors involved in the case company's production, regarding CONWIP loop performance. The purpose of this simulation is to give some indication of the direct effect on cycle time, that the suggestions in chapter 5.4.2. – 5.4.3., would have. The simulation is conducted with ExtendSim 8.0.1.

6.1. Simulated systems and configurations

All processing times used are exponentially distributed. Exponential distribution has the benefit of giving strictly positive values, which is also the case in real processing times. Additionally it always has a CV of one, which is an appropriate level of variability for our modeling. Third it is very simple to use—minimizing the risk of overcomplicating the study. All the systems will include a bottleneck station which has a maximum capacity of 0.667 jobs per minute. Most of the other workstations will have maximum capacity of one job per minute.

For the first three systems, job releases are strictly based on the CONWIP loop in use. In these systems workstations will generally be able to process two jobs simultaneously, for technical reasons. This is useful as opposed to one job at a time as it will increase the feasible CONWIP limit. The CONWIP limit is a discrete number and with a system that has a higher feasible CONWIP limit, the effect of changing the CONWIP limit can be investigated more accurately. The systems and configurations are illustrated visually

as they are in ExtendSim. For an explanation of the blocks seen in the illustrations, see appendix 2.

6.1.1. The Tandem system

The first system simulated—call it “Tandem”—is one of three tandem workstations with the middle workstation as the bottleneck. Two configurations are considered: a CONWIP loop that consists of the whole system and a CONWIP loop that ends on the bottleneck leaving out the last workstation. See figure 22 and table 3.

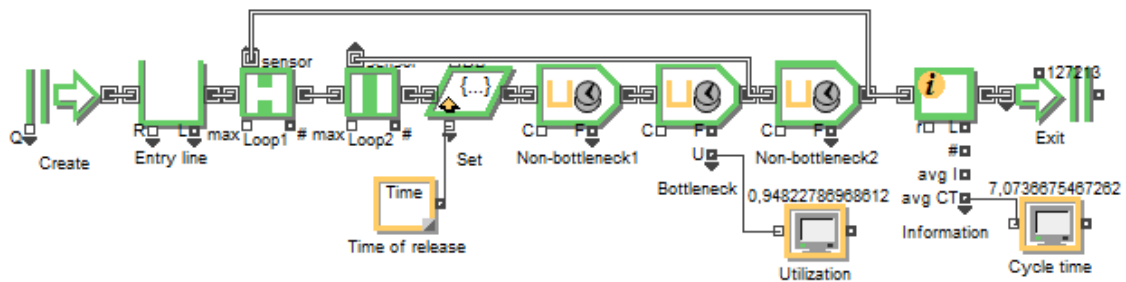


Figure 22. The Tandem system.

Table 3. Workstation parameters for the Tandem system.

Workstation	Processing time (min)	Number of jobs processed simultaneously	Included in CONWIP loops
Bottleneck	3	2	1, 2
Non-bottleneck1	2	2	1, 2
Non-bottleneck2	2	2	1

6.1.2. The Purchasing system

The second system—call it “Purchasing”—consists of three workstations where the first non-bottleneck station and the bottleneck station are set up similarly than in the first system. The third workstation (called purchasing) is set parallel to the other two stations. This workstation has a three minute average processing time and it can work on 100 jobs at once. The purpose of this setup is to simulate a parallel workstation where the rate of production is significantly improved when there are multiple jobs to work on simultaneously. An example would be purchasing of small parts where the impact of our orders on the supplier’s capacity is negligible. In other words the more parts we order the more parts we get without a significant increase in lead time. See figure 23 and table 4.

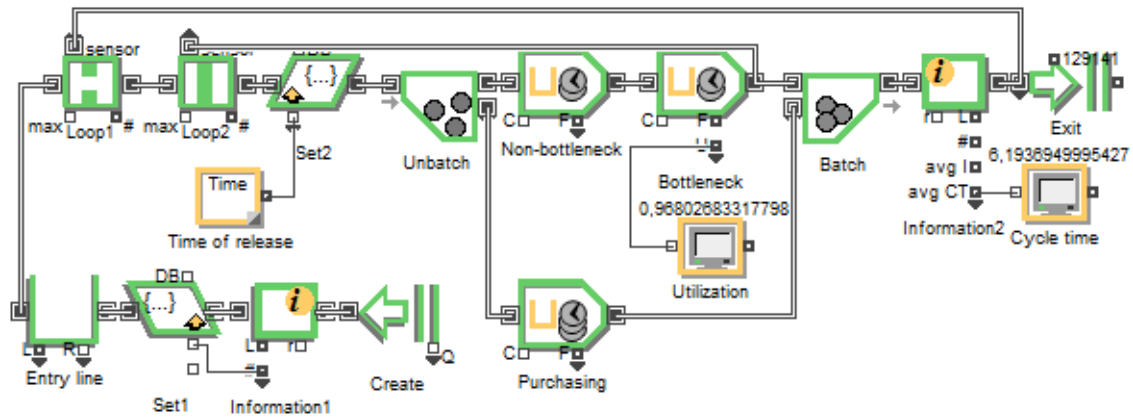


Figure 23. The Purchasing system.

Table 4. Workstation parameters for the Purchasing system.

Workstation	Processing time (min)	Number of jobs processed simultaneously	Included in CONWIP loops
Bottleneck	3	2	1, 2
Non-bottleneck	2	2	1, 2
Purchasing	3	100	1

6.1.3. The Parallel system

The third system—call it “Parallel”—consists of four workstations with two parallel routings each with two workstations. One of the two parallel routings includes the bottleneck. Three CONWIP configurations are simulated: a loop over the whole system; a loop consisting only the routing with the bottleneck; two loops one for each of the routings. The purpose of this model is to simulate a situation where releases are made for two component fabrications of the same end product simultaneously. This differs from the “purchasing” case in that the parallel routing will only be able to work on the same amount of jobs simultaneously as the bottleneck routing. See figure 24 and table 5.

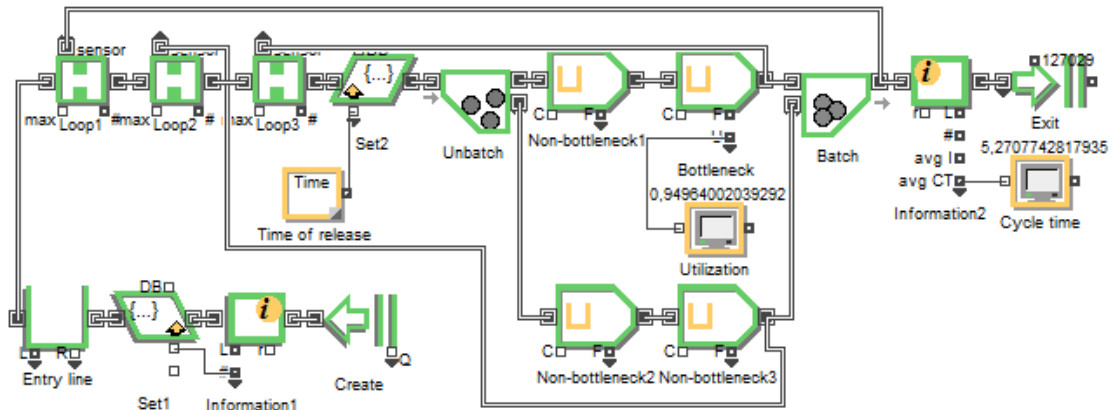


Figure 24. The Parallel system.

Table 5. Workstation parameters for the Parallel system.

Workstation	Processing time (min)	Number of jobs processed simultaneously	Included in CONWIP loops
Bottleneck	3	2	1, 3
Non-bottleneck1	2	2	1, 3
Non-bottleneck2	2	2	1, 2
Non-bottleneck3	2	2	1, 2

6.1.4. The Motors system

The fourth system—call it “Motors”—is constructed to model some of the essential behavior of the case company’s production, regarding CONWIP loop configurations. The scope is limited to a single assembly line. The most significant factors not included are the multiple winding locations and all the secondary effects of a shorter cycle time and smaller cycle time variability illustrated in appendix 1. The secondary effects include factors such as less time wasted by managers on expediting, the effect of which is difficult to include accurately in a simulation. The behavior of the winding operations is complicated, for instance by the manual balancing of the winding locations and by different assembly lines sharing different winding locations. In this system job releases are based on a CONWIP loop and a variability creating workstation. The amount of jobs in simultaneous processing is adjusted to be at a similar ratio than in the case company’s production. See figure 25 and table 6.

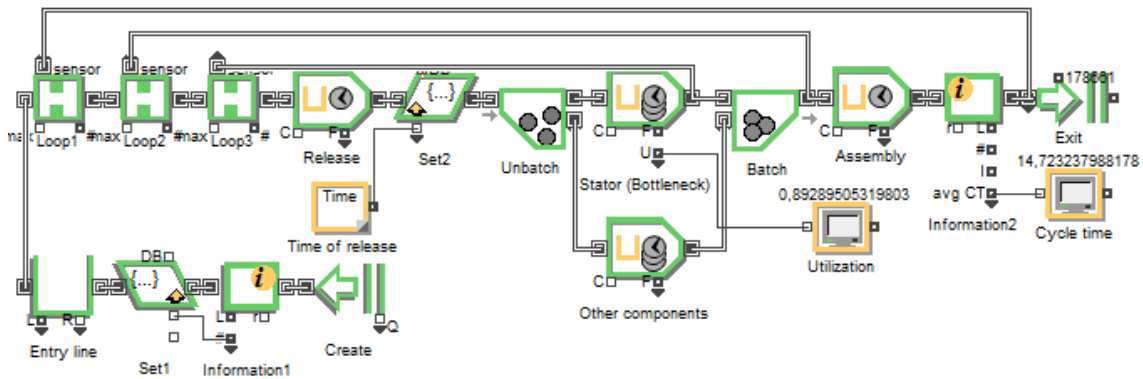


Figure 25. The Motors system.

Table 6. Workstation parameters for the Motors system.

Workstation	Processing time (min)	Number of jobs processed simultaneously	Included in CONWIP loops
Release	1	1	1, 2, 3
Bottleneck	7,5	5	1, 2, 3
Other components	7,5	10	1, 2
Assembly	1	1	1

6.2. Simulation results

The best configuration is the one with the highest TH and smallest CT and WIP. We will measure performance by plotting the relationship of CT and bottleneck utilization. This is done by recording the performance of 5–6 CONWIP parameters for each CONWIP configuration and interpolating between the recorded data points. Results are mainly presented graphically in an approximate manner because the magnitude of the differences between configurations is dependent on the system simulated. There are an infinite amount of possible systems to simulate and therefore—considering the scope of our study—stating exact numerical values on performance serves little purpose. Instead the purpose is to illustrate some general behavior of CONWIP systems with examples. The Motors system will be an exception—for it some numbers will be presented for the purpose of estimating the effects of adjusting the CONWIP loop currently in use in the case company.

Bottleneck utilization is used for illustrating the output instead of TH as it corresponds directly to the TH of the system and it gives more information on the status of the system. By Little's Law we know that CT and WIP give the same information when TH is known. In illustrating the results CT has the advantage over WIP as the effect of time in the system is easier to evaluate than the effect of WIP in the system.

Results for the Tandem system are shown in figure 26. We see two lines one for each CONWIP configuration. The “Whole system” line represents a configuration where jobs are released based on the status of the whole system. The corresponding CONWIP loop in figure 22 is loop1. The “Up to bottleneck” line represents a configuration where jobs are released into the system based on the status of the first two workstations (which include the bottleneck). The corresponding CONWIP loop in figure 22 is loop2. From figure 26 we see that releasing jobs based on “Up to bottleneck” configuration clearly outperforms the alternative, that is, with the same utilization level we have a lower CT.

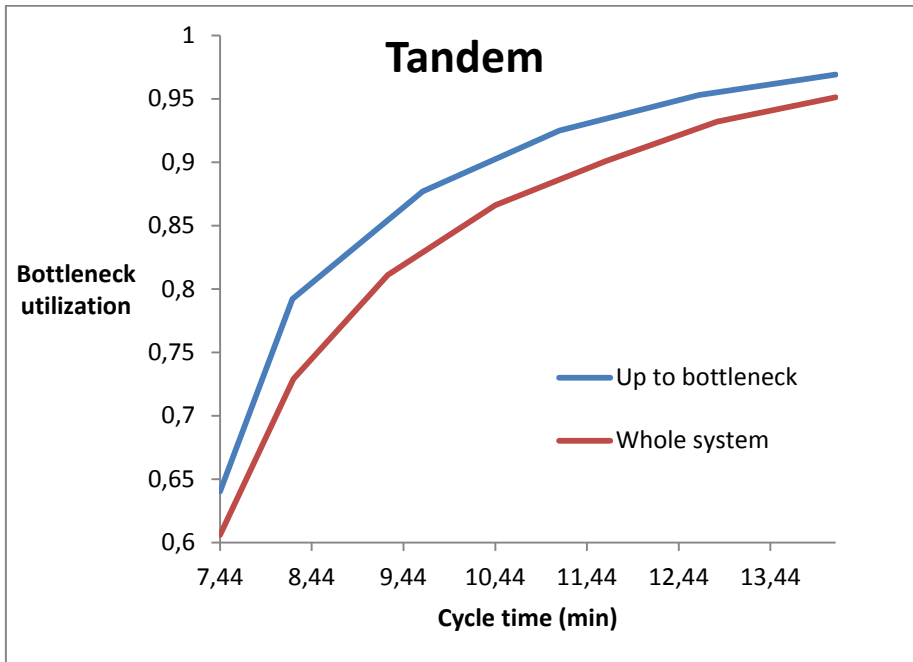


Figure 26. Performance of the CONWIP configurations of the Tandem system.

Results for the Purchasing system are shown in figure 27. As before, we have two lines representing the different CONWIP configurations. The “Whole system” line shows the performance of using loop1 (figure 23) to release jobs into the system. The “Up to bottleneck” line shows the performance of using loop2 (figure 23). We see similar results as before—the configuration “up to bottleneck” outperforms the alternative.

A difference in this system compared with the Tandem system is that the lines in figure 27 are moving closer together as utilization increases. This can be explained intuitively by noting that as utilization increases the cycle time of the bottleneck routing increases, caused by an increased amount of queuing. At the same time the cycle time of the purchasing routing stays the same. Therefore the significance of the purchasing workstation diminishes with a higher utilization.

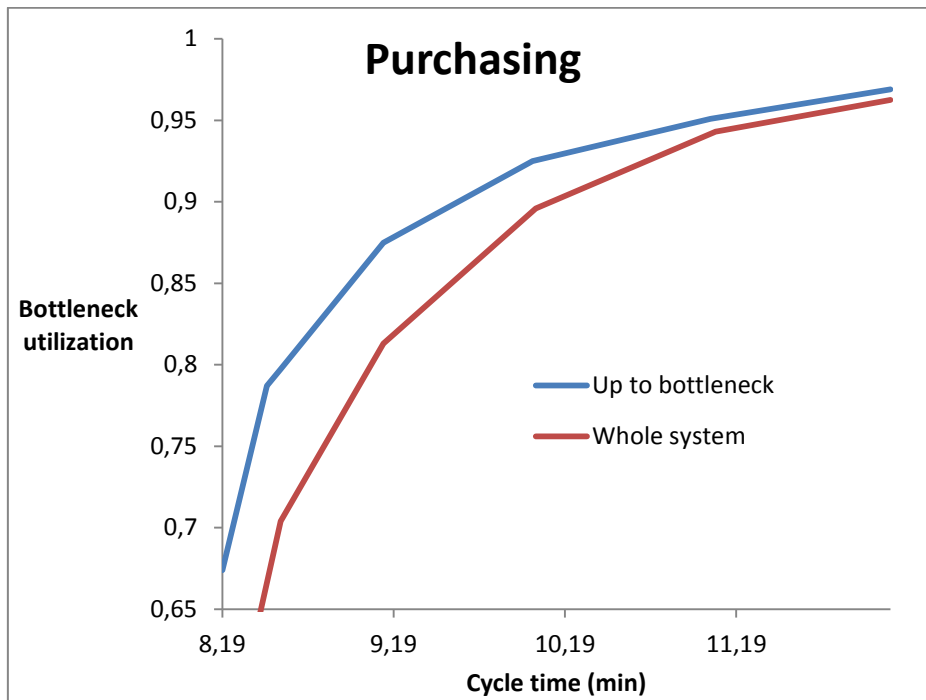


Figure 27. Performance of the CONWIP configurations of the Purchasing system.

Results for the Parallel system are shown in figure 28. In this case we have three CONWIP configurations. The first two are similar in principle than in the other systems. The third configuration consists of two loops. In order for a job to be released the status of both of the loops must be able to accommodate more jobs. We see that the performance of the different configurations seems to be identical.

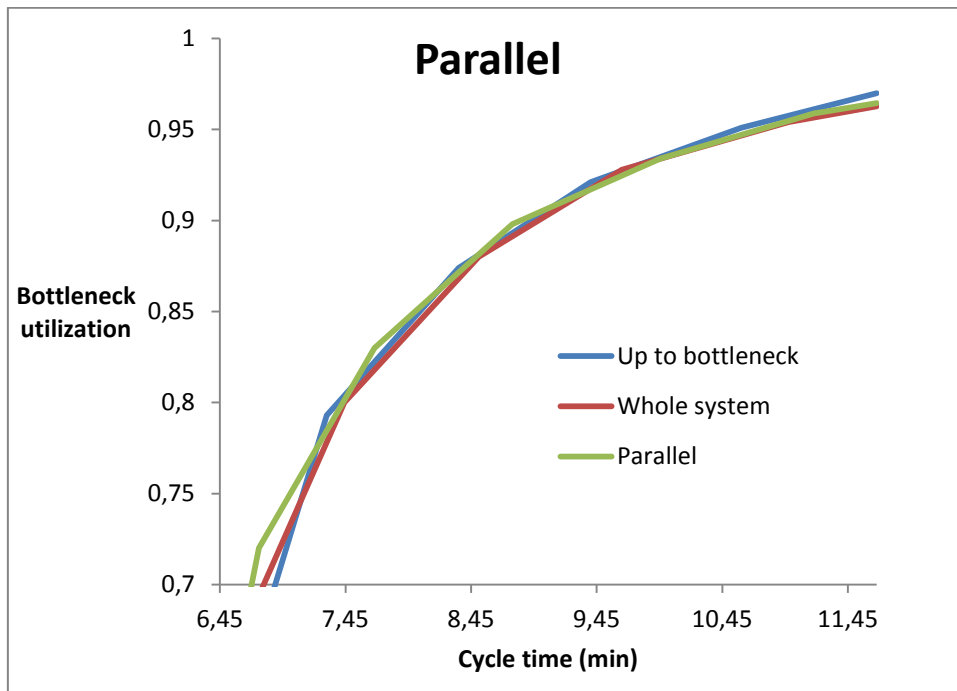


Figure 28. Performance of the CONWIP configurations of the Parallel system.

Results for the Motors system are seen in figure 29. Here we have three CONWIP configurations. We see a clear difference in the performance of the configurations. Starting with a CONWIP loop for the whole system we see an improvement when removing the assembly workstation, that is, the contrast with the “Whole system” loop and the “Components” loop. Then we see a slightly bigger increase in performance with the removal of the “Other components” workstation, that is, the contrast between the “Components” loop and the “Up to bottleneck” loop. In table 6 we see the contrasts in performance in cycle time with a 90% utilization level. Here we are making a comparison with a 22 day cycle time, which is approximately the historical cycle time in the assembly lines AL30 and AL35.

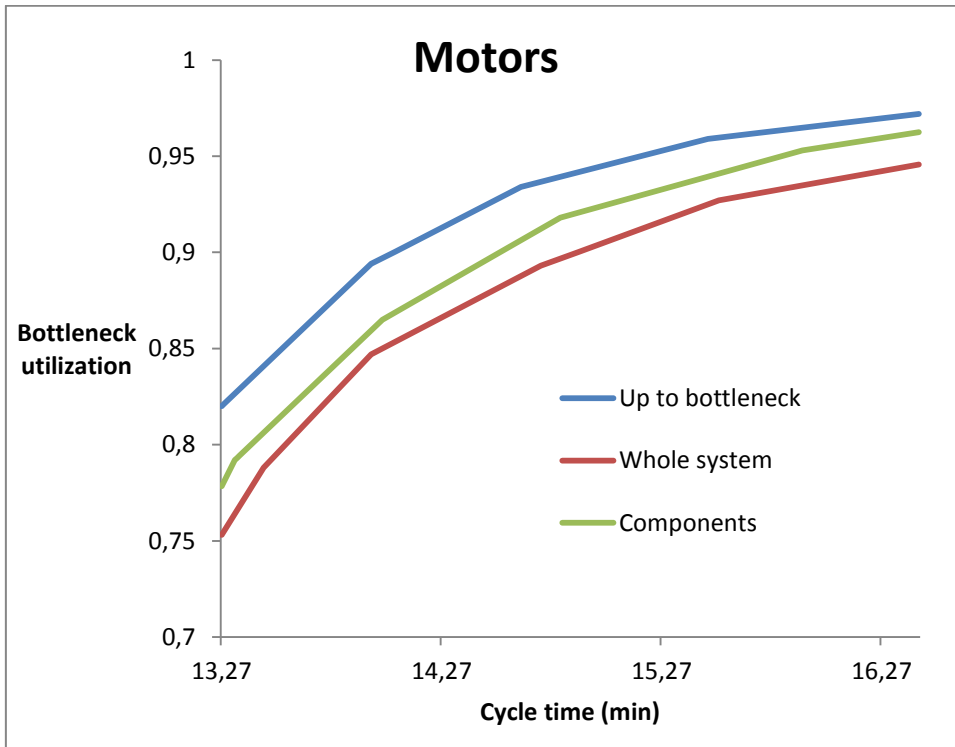


Figure 29. Performance of the CONWIP configurations of the Motors system.

Table 6. Differences in cycle times at a 90% utilization level.

Configuration	Relative cycle time	Cycle time in the case company (work days)
Whole system	1	22
Components	0.967	21.3
Up to bottleneck	0.944	20.7

6.3. Conclusions based on the results

Results from the Tandem system suggest that a CONWIP loop should not include workstations succeeding the bottleneck (as long as they are not also bottlenecks). Intuitively this makes sense as we can imagine that occasionally the stations after the bottleneck functioning slower than average which leads to WIP accumulating in front of

them. When these workstations are included in the CONWIP loop the accumulation of WIP will cause the bottleneck to starve, thus decreasing throughput. However in practice if there is ambiguity of the actual bottleneck then including more stations can be considered the safer choice.

Results for the Purchasing system show that excluding a parallel station that functions similarly as in our model improves the performance of the system. The important distinction here is that stations that can process more jobs simultaneously than the bottleneck will gain a significantly enhanced productivity with a higher WIP than is appropriate to keep in the bottleneck station. Thus it seems clear that extending a CONWIP loop over both, the bottleneck and the parallel “purchasing” type workstation, does not yield optimal performance. In our model excluding the purchasing workstation allows WIP to vary freely in the purchasing station, thus enabling it to produce faster when needed.

Results for the Parallel system show that in some cases there are no significant quantitative differences in performance for different configurations. Therefore it is appropriate to concentrate on optimizing other factors.

The results of the Motors system mirror the results seen with the Tandem and Purchasing systems. Based on the simulation we can give a rough estimate of the direct impact on cycle time that optimizing the CONWIP loop would have. As seen in table 6 the simulation suggests that we can directly shorten cycle time by approximately 1.3 workdays by adjusting the CONWIP loop to the configuration discussed in chapter 5.4.3. Naturally this assumes that WIP is also reduced the appropriate amount as CT cannot be reduced without reducing WIP as long as TH stays constant. This should occur naturally as long as TH is not increased, that is, as long as more motors are not sold.

6.4. Discussion on the simulation study and CONWIP implementation

This study has shown that there can be clear differences in the performance of different CONWIP configurations in the same system. This is important because the cost of using an inferior system can be significant without having any benefit associated with it. The CONWIP configuration used is not dependent on any physical limitations and so there is little reason to stick to a sub-optimal configuration. With these considerations we can summarize that optimizing the CONWIP configuration in use, is an extremely cost-efficient method for improvement.

The problem is that the routings in a real company are much more complicated than used in these simulations and so it can be difficult to find the theoretical optimal configuration. However there are other important requirements for a practical CONWIP configuration than high bottleneck utilization and low CT. Requirements such as simplicity and robustness can make the problem of finding a theoretically optimal configuration obsolete. Without a robust configuration a small unexpected change in the production environment can derail a previously theoretically optimal system into chaos.

The production environment of companies is subject to continuous change. If a CONWIP configuration is not easy to understand and high maintenance, that is, not simple, then a company is unable to make the appropriate changes and tweaks to the CONWIP configuration required by the changing production environment. This process can turn a highly efficient system into a highly inefficient system in a few years depending on how much change the company has undergone.

As for the best configuration for a system like our Parallel case, a loop consisting of the whole system would be the most robust as the bottleneck is free to vary inside the loop. On the other hand a loop consisting only of the bottleneck and the workstation leading to it would be the simplest alternative. One could argue that these two alternatives are equally good and that the configuration of two CONWIP loops is the worst choice due to its relative complexity.

The Motors system gives some further rationale for the suggestions presented in chapters 5.4.2. – 5.4.3. However the final suggestion presented in chapter 5.4.4. was

deemed too complicated for the scope of this study. From a practical perspective the suggestion in chapter 5.4.3. is a worthwhile transitional stage to the suggestion in 5.4.4. Therefore it should be considered and studied first.

7. CONCLUSIONS

This thesis has made an effort to study and explain some of the concepts regarding the fundamental behavior of a production facility from the perspective of the case company. The results were then used to find effective approaches for improvement. Some of the more useful concepts discussed were: Little's Law, lead time considerations, variability, buffering and push and pull systems. Other relevant topics concerning production control which were excluded from this thesis due to scope limitations are: cellular manufacturing, batch size and setup issues.

Another task of this research has been to help utilize some of the more practical works of academics. In the past the practical implementation and academic research in production control have not mixed well together. This has perhaps conditioned managers to avoid heavy theory for the fear that it will yield little benefit. This thesis try's to improve this situation by presenting useful concepts for the case company.

The research question emphasizes improvement. Two focus points for improvement were suggested: reduction of lead times and reduction of buffering. In order to reduce lead times and buffering in the case company, this thesis suggests the following: (1) modify the CONWIP loops in use to give improved support for the use of the CONWIP protocol, (2) replace WIP-buffering with capacity buffering in workstations where capacity is not expensive, (3) emphasize the reduction of variability, (4) as variability is reduced, reduce the amount of WIP. Fifth, it is suggested that the role of DBR is reviewed in order to clarify the guidelines of the case company's production control.

Specific improvement examples were presented regarding CONWIP based on the theory discussed and the analysis of the case company. The purpose of these suggestions is twofold: (1) to give practical examples of how to proceed with the modification of the CONWIP loops, (2) to introduce new perspectives and stimulate new ideas concerning the case company's production control.

To introduce useful vocabulary and further improve intuition some simple concepts of queuing theory have been introduced. Even though an average production facility is much too complicated to be modeled with a queuing system accurately, we can still exemplify some common relations in practice with concepts from queuing theory.

An important benefit of generalizing practical issues under a theory framework is that we gain vocabulary to use when discussing these issues. As an example, if a manager recognizes when a practical situation is related to pooling, she will understand the problem much faster and be able to communicate with her colleagues more efficiently about the situation. Also a new perspective is achieved to contrast the daily routine present in production facilities, for instance the perspective of a queuing system. The more perspectives that we understand of a system the better we are able to make decisions regarding the system.

Much of this thesis has concentrated on the CONWIP protocol. There are a few reasons for this: it is a simple and a powerful method for many production facilities and it can be considered to be a good fit for the case company; the case company already uses a variation of CONWIP; presently there is not much information available on the implementation of CONWIP. The theoretical portion presented preceding CONWIP can be considered to be essential for understanding the issues concerning CONWIP.

A simulation study was performed to further improve the understanding of CONWIP configuration issues and to give a quantitative perspective on behavior previously explained only based on literature and intuition. Some of the simulation results were according to expectations while some were not. These types of simulations and issues regarding CONWIP implementation in general are a good candidate for further research. Additionally simulation was used to evaluate the effects of the suggestions regarding CONWIP implementation. The simulation results showed that the CONWIP configuration used does indeed have an important effect on the performance of a routing.

Even though in production control buzzwords such as QRM, TOC and TPS (and many

more) are very popular, they have intentionally been kept to a minimum. This was done because buzzwords come and go but the fundamentals stay. In fact, it could be argued that the reason that the field of manufacturing management is filled with buzzwords is because the fundamentals are not understood. Instead of realizing that the performance of a production facility is largely determined by a few basic relations, such as variability and buffering, we turn to the latest fad for an easy fix.

REFERENCES

- Benders, J. & J. Riezebos (2002). *Period batch control: classic, not outdated*. Production & planning control 13:6, 496–506.
- Bonney, M.C., Zongmao Zhang, M.A. Head, C.C. Tien & R.J. Barson (1999). *Are push and pull systems really so different?* International Journal of Production Economics 59:1-3, 53–64.
- Bonvik, Asbjorn M., Yves Dallery & Stanley B. Gershwin (2000). *Approximate analysis of production systems operated by a CONWIP/finite buffer hybrid control policy*. International Journal of Production Research 38:13, 2845–2869.
- Burbidge, John L. (1996). *Production flow analysis for planning group technology*. New York: Oxford University Press Inc. 179 p. ISBN: 0-19-856459-7
- Gaury, Eric & Jack P.C. Kleijnen (2001). *Short-term robustness of production management systems: A case study*. European Journal of Operational Research 148:2, 452–465.
- Goldratt, Eliyahu M. & Robert E. Fox (1986). *The Race*. Great Barrington: North River Press Inc. 179 p. ISBN: 0-88427-062-9
- Goldratt, Eliyahu (1990). *The Haystack Syndrome*. Great Barrington: North River Press Inc. 262 p. ISBN: 0-88427-089-0
- Goldratt, Eliyahu M. & Jeff Cox (2004). *The Goal: A Process of Ongoing Improvement*. ed. 3 Great Barrington: North River Press Inc. 390 p. ISBN: 0-88427-178-1
- Goldratt, Eliyahu M. (2008). *Standing on the Shoulders of Giants: Production concept versus production applications The Hitachi Tool Engineering example*. [Online]

Available from World Wide Web: <URL:
http://goldrattschools.org/pdf/shoulders_of_giants-eli_goldratt.pdf

Hopp, Wallace J. (2008). *Supply Chain Science*. New York: McGraw-Hill/Irwin. 230 p.
 ISBN 978-0-07-340332-8

Hopp, Wallace J., Mark L. Spearman & David L. Woodruff (1990). *CONWIP - A Pull Alternative to Kanban*. *International Journal of Production Research* 28:5, 879–894.

Hopp, Wallace J. & Mark L. Spearman (2000). *Factory Physics: Foundations of Manufacturing Management*. ed. 2. New York: McGraw-Hill/Irwin. 701 p. ISBN 0-256-24795-1

Hopp, Wallace J. & Spearman, Mark L. (2004). *To Pull or Not to Pull: What Is the Question?* *Manufacturing & Service Operations Management* 6:2, 133–148.

Hopp, Wallace J. & Mark L. Spearman (2011). *Factory Physics*. ed. 3. New York: Waveland Press, Inc. 720 p. ISBN 1-57766-739-5

Little, John D.C. (1961). *A Proof for the Queuing Formula: $L = \lambda W$* . *Operations Research* 9:3, 383–387.

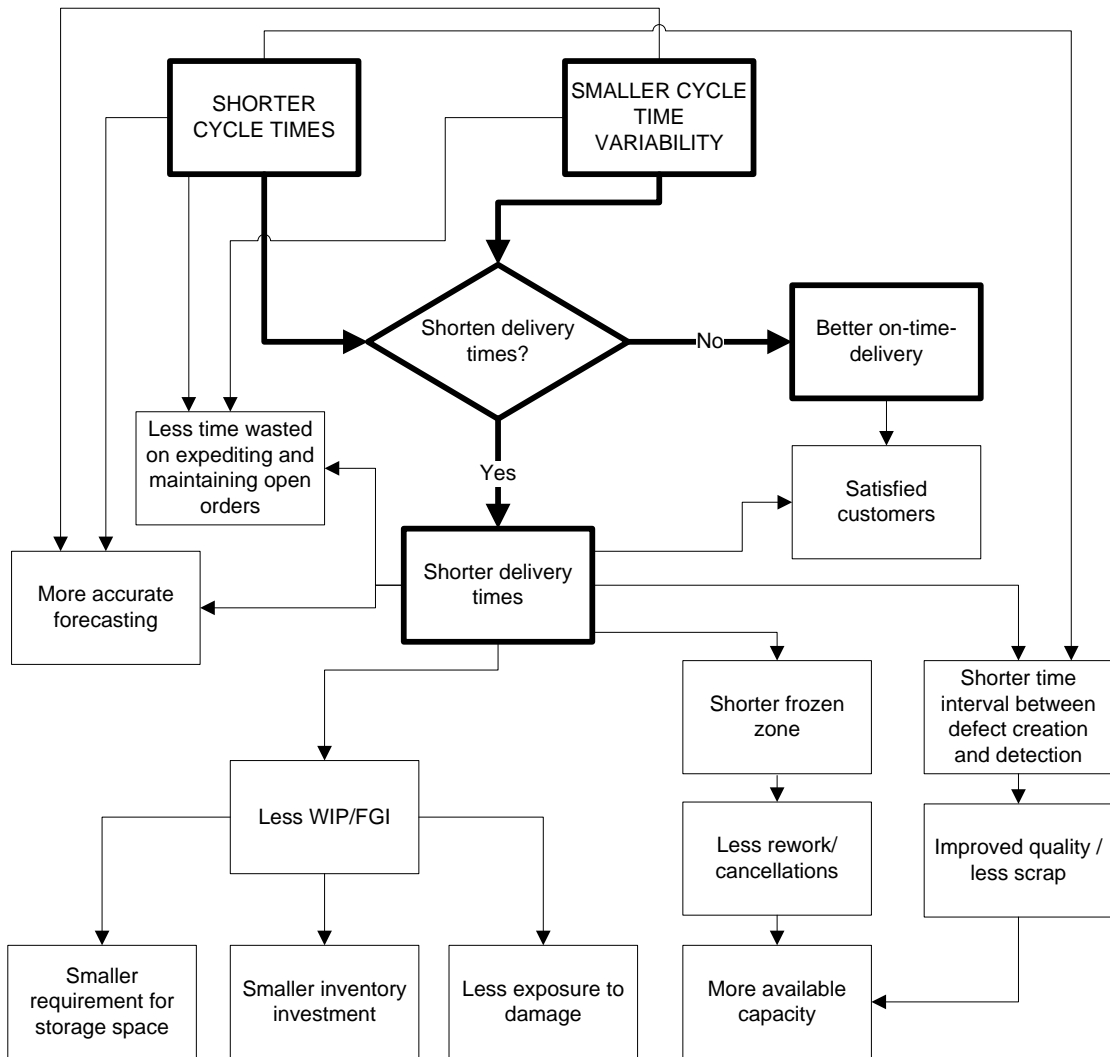
Little, John D.C. & Stephen C. Graves (2008). *Little's Law*. In: *Building Intuition: Insights from Basic Operations Management Models and Principles*, 81-100. Dilip Chhajed, Timothy J. Lowe. New York: Springer Science + Business Media, LLC ISBN: 0-38773-698-0

Roderick, Larry M., Don T. Phillips & Gary L. Hogg (1992). *A comparison of order release strategies in production control systems*. *International Journal of Production Research*. 30:3, 611–626




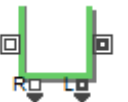

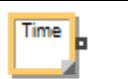
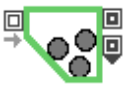

- Schrageheim, Eli & Boaz Ronen (1990). *Drum-Buffer-Rope Shop Floor Control*. Production and Inventory Management Journal. 31:3, 18–22.
- Schrageheim, Eli, J. Cox & Boaz Ronen (1994). *Process flow industry—scheduling and control using theory of constraints*. International Journal of Production Research. 32:8, 1867–1877.
- Schrageheim, Eli & H. William Dettmer (2000). *Simplified Drum-Buffer-Rope A Whole System Approach to High Velocity Manufacturing*. [Online] Available from World Wide Web: <URL: <http://www.goalsys.com/books/documents/S-DBRPaper.pdf>
- Spearman, Mark L. & Michael A. Zazanis (1988). *Push and Pull Production Systems: Issues and Comparisons*. Operations Research: 40:3, 521–532.
- Suri, Rajan (1998). *Quick Response Manufacturing: A Companywide Approach to Reducing Lead Times*. New York: Productivity Press. 544 p. ISBN: 1-56327-201-6
- Suri, Rajan (2010). *It's About Time: The Competitive Advantage of Quick Response Manufacturing*. New York: Taylor & Francis Group. 210 p. ISBN: 978-1-43-98-0595-4
- Suri, Rajan (2010b). *It's About Time: The Competitive Advantage of Quick Response Manufacturing; Appendix A*. New York: Taylor & Francis Group. 32 p.
- Zwillinger, Daniel (2003). *CRC Standard Mathematical Tables and Formulae*. ed. 31. Boca Raton, London, New York, Washington: D.C.Chapman & Hall/CRC Press LLC 910 p. ISBN 1-58488-291-3




APPENDIXES

APPENDIX 1. Effects of shorter cycle times and smaller cycle time variability



APPENDIX 2. Explanation of the blocks used in simulations

Block	Name	Description
	Create	The create block creates the jobs that flow through the system. In our models jobs are created so that there are always ample jobs to be released into the system by the gate block(s).
	Information	The information block extracts information from the jobs that flow through it. In our models it is used for two purposes: to record CT and to calculate a serial number for the jobs to be set by the “set” block.
	Set	The set block sets some attribute for the jobs flowing through it. In our models it sets the serial number for the jobs in models “purchasing” and “parallel” and in all our models it assigns the time that a job is released to the systems in order to calculate the CT.
	Queue	The queue block accumulates the jobs that are not authorized to be released to the next phase yet.
	Gate	The gate block enforces the CONWIP protocol by limiting releases to the system based on the status of the system and the parameter (CONWIP limit) assigned to it.
	Time	The time block gives the time in the simulation. We use it as an input for the “set” block in order to give the jobs a time attribute when they are released to the system.
	Unbatch	The unbatch block divides jobs into multiple parts.
	Workstation	The workstation block is a combination of two blocks: queue and activity. Queue is explained above and activity causes a delay for the job. In our models activity determines the processing time and the amount of jobs that are

		processed simultaneously.
	Display	The display block displays a value assigned to it. In our models it is used to display the utilization level of the bottleneck and the CT of the system.
	Batch	The batch block combines multiple components into one. We use it to match components from different routings based on their serial number.
	Exit	The exit block exits the jobs from the model. It displays the number of jobs exited.