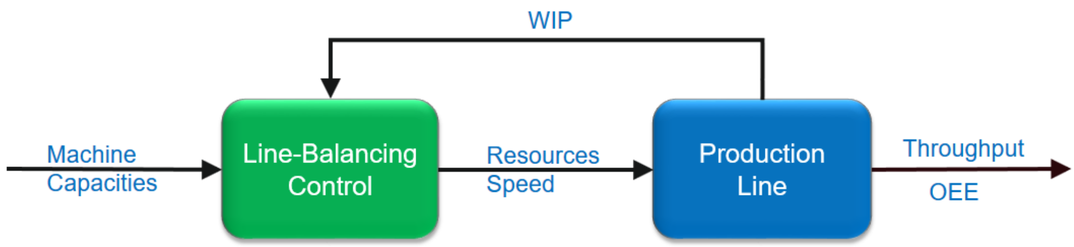
CHALMERS
UNIVERSITY OF TECHNOLOGY



Adaptive Line Balancing

Effective Control of Production Lines

Master's thesis in Production engineering

Alex Andreasson
Jakob Viklander

Department of Industrial and Material Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

Adaptive Line Balancing

Effective Control of Production Lines

Alex Andreasson
Jakob Viklander

Department of Industrial and Material Science
Chalmers University of Technology
Gothenburg, Sweden 2020

Adaptive Line Balancing
Effective Control of Production Lines

Supervisor: Daniel Nåfors, Department of Industrial and Material Science
Supervisor: Mikael Henningsson, FlexLink
Examiner: Björn Johansson, Department of Industrial and Material Science

Master's Thesis 2020
Department of Industrial and Material Science
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Adaptive Line Balancing
Effective Control of Production Lines
ALEX ANDREASSON
JAKOB VIKLANDER
Department of Industrial and Material Science
Chalmers University of Technology

Abstract

With raising prices on energy, companies will need to find solutions which lowers the energy consumption to cut their costs. One way of decreasing the energy consumption could be finding a solution which adapt the speeds or shut down machines and conveyors in production systems depending of the situation. In this report a new concept of an adaptive control system is evaluated. By counting the Work-In-Progress within segments the adaptive control system decides the speeds of the machines and conveyors in real time. The expected outcome of the production system is improved production rate, decrease of starvation and blocking, improvement of gentle handling with smoother operations and reduction of energy consumption and noise. The goal was to evaluate the adaptive control system by implementing it in a simulation model of a test system.

The project followed Banks methodology but the development of the simulation model was split into four steps where each step was verified separately to ensure functionality and quality of the model. The final simulation model was combined out of the four validated models to give it the functionality needed. The simulation model was then run with and without the adaptive control system and the results were compared.

The goal of the project was achieved and ALB was verified to be working in a virtual environment. In a physical test rig the system for counting WIP does not have the accuracy needed for a perfect implementation.

The results from ALB in the virtual environment shows that on average the software is able to reduce the machine speeds with 5% while still managing to produce the same throughput as a conventionally steered system. If implemented correctly the effect of this should yield a lower energy consumption, lower the noise, and prolong the life length of machines and conveyors, in factories.

Acknowledgements

We would like to express great gratitude to our project leader Joakim Jönsson and our two supervisors Daniel Nåfors and Mikael Henningsson who came with a lot of valuable input during the project. During emulation we got help and would like to thank Koen Hermans and Tommy Fässberg for their input and knowledge in Visual Components. In regards to the physical implementation we want to show gratitude towards Jonas Zedén whom with his knowledge in hardware and PLC-programming was of great help when putting the work into practise.

The authors, Gothenburg, September 8, 2020

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the reader to the background and purpose of the master thesis. In this chapter the background, purpose, aim and limitations are presented.

1.1 Background

The environmental impact of manufacturing industries has continuously increased and due to that it has become critical to investigate in how large the energy demand is for given processes, while also try to find ways to lower the amount of energy needed to perform the work (Cheng et al., 2020). The industrial sector consumes about 40% of the worlds energy and 46% of the industrial sectors energy is used by manufacturing industry (Li, Lennartson, Tang, Biller, & Matta, 2017). With raising prices on energy, a system with low energy demands per product will give competitiveness to a company in a stacked market (Li et al., 2017). To gain high competitiveness there is a need to have good productivity, this means that companies wants to maintain a high production rate whilst lowering the energy consumption (Jia, Zhang, Arinez, & Xiao, 2016). More than 85% of the total energy consumption of factories is spent on functions which aren't directly related to the production of parts (Gutowski et al., 2005). In order to decrease the total energy consumption, machines and conveyors can be temporarily switched off or run at a lower speed instead of being idle (Jia et al., 2016). However, switching off machines and conveyors causes a trade off between production rate and environmental impact unless it can be done at the correct times.

Several methods have been proposed to find times at which machines and conveyors may be turned off without impacting the production rate (Mashaei & Lennartson, 2013; Chang, Xiao, Biller, & Li, 2013). The methods which turn off machines and conveyors uses complex algorithms to perform the calculations. Often the adaptive control system needs to be tweaked to make it work to the current control system, although in many cases only modest efforts are needed to make a complete system work well (Wittenmark, 1988). If properly integrated, the algorithm should theoretically allow a company to save energy without impacting the production rate, or increasing the production rate with minimal increase in the environmental impact.

Algorithms which has the ability to steer a type of control systems are known as adaptive control systems. Adaptive control systems uses feedback to intelligently adjust the characteristics of the system to a satisfying state (P. Zhang, 2010). Adaptive control systems are most suitable for mechanical systems without specific time delays and systems with clearly defined dynamics. The control system often needs to have some knowledge to how the system structure is built and has a hard time to steer nonlinear, structure-variant and systems with large time delay processes (P. Zhang, 2010).

The interest to investigate and develop balancing control systems has risen due to belief in companies that there is unlocked potential in today's conventional steered systems. Conventionally steered systems are designed as linear systems which does not handle disruptions well. If the conventional system runs as intended it works fine, but if there are eg. unpredicted stops in the system the performance will go significantly down (Embention, 2020). In comparison an adaptive control can do changes on its own to balance out the disturbances and keeping the system more stable. If a system can be operating at a lower load with changeable speed there are possibilities of economical winnings for the users in terms of energy consumption and lower Work-In-Progress (WIP). If these economical winnings can come without having to invest in new hardware there is a win-win situation were the provider sells the software while the user will save money over time with lowered surrounding costs.

One type of adaptive control system is a software which is called Adaptive Line Balancing (ALB). ALB includes a number of control algorithms which will learn the correlation between machine capacities and use that information to systematically adjust the control parameters on-line. The purpose of ALB is to maintain the maximum production rate while minimising the load on the line. Expected outcome of ALB is:

- Improvement of production rate (Overall equipment efficiency (OEE))
- Decrease starvation and blocking events
- Reduction of load and energy consumption
- Improvement of gentle handling and smooth operation
- Reduction of noise and stress

The algorithm has shown positive results in initial simulations and the mathematical models has been proven. The next step is to take ALB from the research phase into a phase of verification, validation and system testing in reality.

1.2 Purpose and Aim

The aim with the thesis is to validate if ALB is a solution which can increase the production rates within factories without increasing the load on the system. ALB should be able to increase of production rate by managing real time monitoring of the WIP in the system and control the system based on the current situation.

While bringing ALB into the testing phase is the primary objective, a secondary object is to ensure that the algorithm has a possibility to later be introduced to the market. To make the testing as realistic as possible the system which ALB gets validated on needs to be of industry standard. One of the most common system used in the industry today is driven by PLCs. Due to that it is important to make sure that ALB and PLCs can work and communicate with each other in real time.

The thesis shall investigate and answer the following questions:

1. What are the prerequisites on a system to be able to implement ALB?

   It is already known that ALB is dependent of constantly getting input data from the system of the current WIP count, but how high does the precision need to be, and what are the possible solutions to achieve this precision? Furthermore, are there additional prerequisites on systems in order to be able to implement ALB?

2. What are the benefits with using ALB?

   What benefits may come with using ALB? With ALB the system should in theory become more efficient and if the theory works in reality, how much more efficient is the system going to be for the end user?

3. Where is it suitable to use ALB?

   Even if a system fulfills the prerequisites to be able to implement ALB, it is not certain that ALB is suitable to be used in that system. Are there some factors which determines if ALB will be beneficial on the systems and will there be drawbacks with using ALB?

1.3 Limitations

The algorithm is only tested on one specific system that FlexLink has chosen as suitable for the thesis. The simulation software does not have a built in function to measure energy consumption of the machines and conveyors. Dependant on what type of machine is simulated the values of energy will change, this adds complexity to the calculations. Due to the complexity, the energy consumption is only estimated. Due to the short time with a complete system, caused by the covid-19 pandemic, only twelve different scenarios is tested. Since the simulations are done in real time, the tests of each scenario is only run for 70 minutes. Other hardware solutions to get a working implementation of the algorithm is not further investigated due to economic constraints.

2

Theory

The chapter describes the theory of the thesis and gives the reader a context to the different parts of the production system that is needed for the adaptive control system to work.

2.1 Production Systems

Production systems can be viewed as a transformation process, as it transforms resources into parts and/or products which contains a higher value than the resources themselves (Holstein & Tanenbaum, 2012). To achieve the transformation process the system needs resources such as materials, machinery, labour and territory (buildings, area etc.) to be able to complete the transformation from material to product.

A production system can be seen as a process with multiple flows. There are physical flows such as Work-in-Progress and finished goods but there is also flows of information which can be used for planning and controlling the production process (Holstein & Tanenbaum, 2012). The physical flows is responsible for constraints such as the capacity of the system, and limit of the maximum output. The planning and control, which is dependent on the information flow, can be used to retrieve data which can be used to change the system to get improvements.

There is three general types of basic production systems used in the industry today, fixed position systems, batch systems and continuous system (Holstein & Tanenbaum, 2012). The fixed position system is used in large projects where materials are moved to the finished product rather than having the product move around. This type of system is used when producing for example buildings, airplanes and ships (Bellgran & Säfsten, 2010). An overview of the fixed position layout can be found in Figure 2.1.

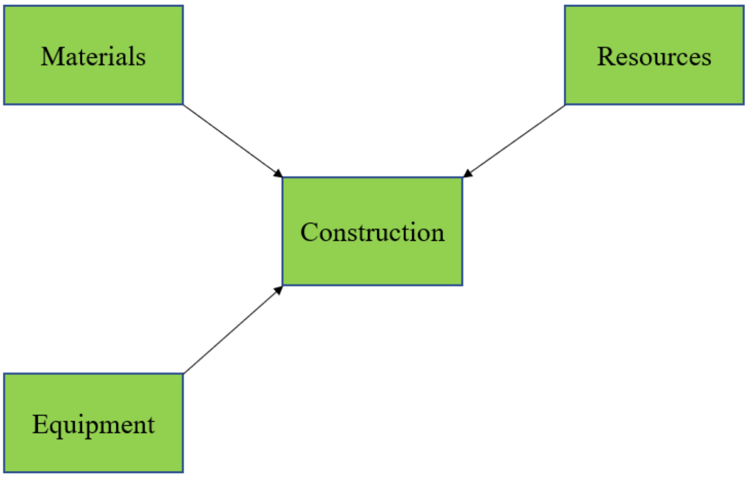


Figure 2.1: Fixed Position Layout

The batch systems or job-shop system uses general purpose equipment to produce smaller batches of product which are moved between the resources (Holstein & Tanenbaum, 2012). The specification of the products produced in batch systems can vary a lot between one batch to the next. This type of system can be used when producing many different product types in small volumes(Bellgran & Säfsten, 2010). An example layout of a batch system is shown in Figure 2.2

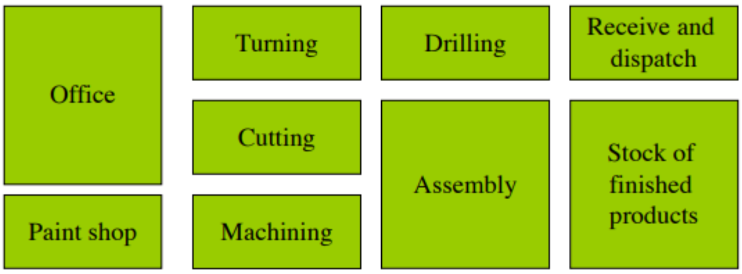


Figure 2.2: Batch System Layout (Bellgran & Säfsten, 2010)

The third common system is the continuous system were the products are processed while flowing through the system. To help the products move from process to process a common solution is to use conveyor systems to move products through the factory, see Figure 2.3 for overview of a continuous system. The continuous system is used in large volume applications were the products does not differ so much between one another (Bellgran & Säfsten, 2010). Tasks along the continuous system are often divided into shorter segments in which the workers can perform the tasks in a efficient way (Holstein & Tanenbaum, 2012).

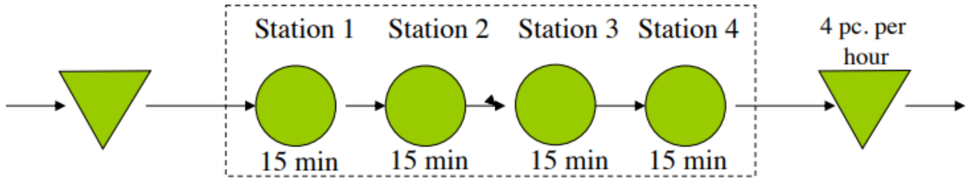


Figure 2.3: Continuous System Layout (Bellgran & Säfsten, 2010)

2.1.1 Conveyor systems

A conveyor system can be seen as a continuous system were the conveyor is used to make the products flow through the factory. The max speed of conveyor belts are restricted by several factors such as belt dimensions, product dimensions, weight on the belt, and belt load (Ananth, Rakesh, & Visweswarao, 2013). Some of these are constant factors whilst some of them vary depending on the variations and disturbances in production. The speeds of conveyor belts are then often calculated based on constant variables which are not accurate in reality due to these oscillating effects (Mosadegh, Fatemi Ghomi, & Süer, 2017). This results in conveyor belts either being run at a too high speed without gaining any benefits or running at a too low speed which results in time losses. Not only does running the conveyor belts at a higher speed not bring any benefits but it increases the energy consumption (S. Zhang & Xia, 2010), as well as the noise levels in the facility.

The electricity consumed by conveyor systems is a big part of energy consumption in companies and is a post which economically could be reduced. As an example a dry bulk terminal has conveyor systems consuming 50-70 percent of the total amount of energy. If it is then consider that 41 percent of the total amount of electricity in the world is produced by burning coal and the coal makes up for 45 percent of the worlds total CO2 emissions (He, Pang, Lodewijks, & Liu, 2018) there is a big reason to try to find a way to lower the energy usage. To reduce the energy usage there is a demand to find solutions which can reduce the amount of electricity needed. One of the alternatives is to find a speed control to the conveyors. With speed control it is suggested that the conveyor should not be run faster than needed. For example there is no need to run the belts at full speed if there is a low load on the belt since there will not be a higher production output by doing that (He, Pang, & Lodewijks, 2017).

2.2 Production control

Introduces types of production control theories of how to improve and increase the efficiency of a production system. The chapter also contain an explanation of a new type of adaptive control system.

2.2.1 Theory of constraints (TOC)

The theory of constraints (TOC) was introduced in the book "the goal" written by Eliyahu Goldratt in 1984 (Goldratt, Cox, & Whitford, 2004). TOC is a way to make business run more smoothly while increasing business income (Butts, 2019). TOC rests on the principle that every system has at least one constraint which limits the performance of the system (Rahman, 1998). Hence in order to improve a system the company has to find these constraints, often called bottlenecks. The importance lies in reducing the impact that the bottleneck has for the business environment. By continuously seeking these constraints a company can refine the processes and with that improve efficiency (Butts, 2019).

TOC is built up with five steps of how a company should think and react in regards to exploiting the constraints (Şimşit, Günay, & Özalp Vayvay, 2014). An exploited constraint will make the process smoother which will improve a company's efficiency (Goldratt et al., 2004). The five steps are listed below:

1. **Identify the constraint**
   In the first step of TOC the business need to find the constraint(s) of the system, a good way to find the constraints can be to look into areas in the business in which there is excess WIP. Equipment and underlying systems is commonly found to be constraints since if they are not working correctly and have redundancies it can negatively affect the system (Butts, 2019). Constraint does not necessarily only exist inside the company, constraints from the outside can be eg. competitors to a specific market segment (Butts, 2019).

2. **Exploit the constraint**
   The second step in TOC is to exploit the constraint, these exploits have to be done using resources that the company has at hand (Butts, 2019). It could be to move processes to other departments or use different machines that already exists in-house. If there is a need, the business could hire a company to help out with some of the workload.

3. **Subordinate everyone to the constraint**
   In the third step, also known as the subordinate step, the business leaders have to make sure that the weak link gets support from the other links in the company (Butts, 2019). If the third step does not solve the constraint, the next step which is to elevate the constraint has to be initiated.

4. **Elevate the constraint**
   When elevating the constraint the leaders have to find new ways to solve the problem, this can include spending money to solve the constraint if necessary (Butts, 2019).

5. **Prevent inertia to become the constraint**
   The last step of TOC is to find a new constraint to do the process all the way from the beginning. There is always a constraint in a system that can be

improved upon to increase the efficiency within a company (Butts, 2019). The TOC is a continuous process which should be used to increase the financial performance over a long period of time.

2.2.2 Adaptive Control Systems

There are many different adaptive control systems which can be used to increase efficiency within production. One of these adaptive control systems is Adaptive Line Balancing (ALB). ALB offers a dynamic control of a production system since it allows for online changes to be carried out based on the current status of the production. Using the TOC methodology, ALB is constantly applying step 1, 2, and 3 on the production system. The advantage of dynamically applying TOC is that bottlenecks are rarely constant bottlenecks but rather the bottleneck shifts during production (Roser, Nakano, & Tanaka, 2003). This means that a "static" or offline implementation of TOC can not account for all of the dynamics in a production system.

ALB breaks down the structure of a production line into zones, segments, machines, and conveyors. A line may contain several zones and a zone is defined as a part of the production line which starts or ends with either a diverter, assembler, or merger. Each zone contains one or more segments, each segments starts and ends with a machine and in between the machines there is one or more conveyors. How segments are divided in a continuous system is over viewed in Figure 2.4.

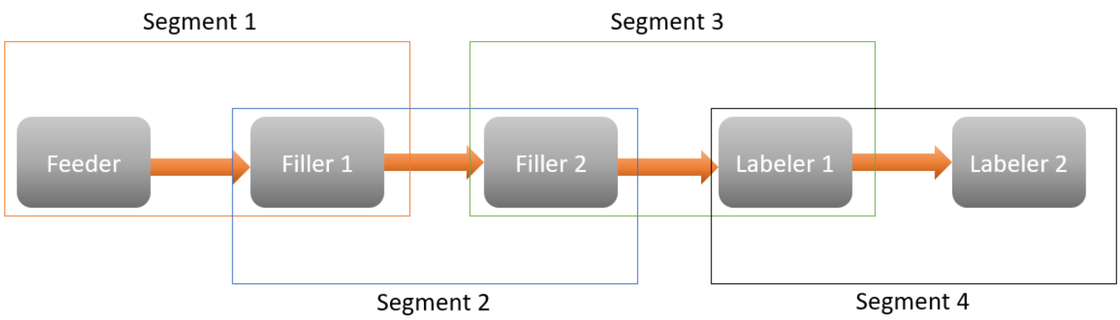


Figure 2.4: How ALB divides a system into segments

To identify bottlenecks in the system, ALB monitors the WIP in each segment. If there is a high WIP in one segment ALB may increase the speed of conveyors and the end machine of that specific segment and thereby lower the WIP and increase the throughput in that segment, hence exploiting the constraint. If the end machine in the segment is already running at full capacity, ALB will reduce the speed, or turn off, upstream and downstream machines to avoid wasting energy, hence subordinating everyone to the constraint. The feedback function can be seen in Figure 2.5.

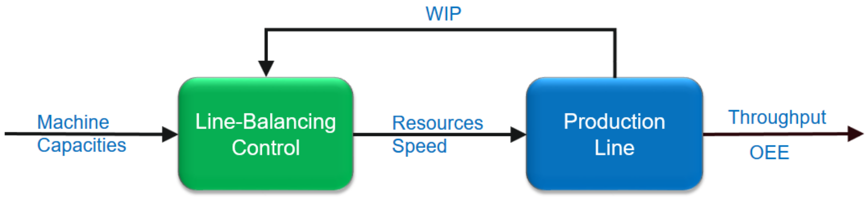


Figure 2.5: ALB feedback function

2.2.3 Production monitoring

The Sub-Chapter gives an introduction to what WIP is and how it affects companies economical. There is also explanations to different types of systems which can be used to count and monitor WIP within a production system.

2.2.3.1 WIP counting in production

WIP is the amount of products that has entered a system but are waiting to be completed or the value of these parts. WIP is part of a company's balance chart and include costs of raw material, labor and overhead costs which has gone in to moving the product to were it is in production. Finished goods that are stored in inventories are not to be counted as WIP and neither is raw material which has not entered the production yet. When the product moves through the production the production cost of the product increases, this cost is accounted as increased WIP cost of the product. The product goes from being counted as WIP to finished goods when it is complete and ready to be sold (Investopedia, 2020). There are a number of different ways to count WIP and a some of them are mentioned in this chapter.

Photoelectric sensors
The sensors that are used in the test rig is of a type called photoelectric sensors. This sensor is very flexible and can be used in many different applications. The photoelectric sensor is a device which can detect differences in light levels which are being sent from a light source. The sensor contains a light source, amplifier, signal converter and an output (Tri-Tronics, 2020). There are three types of photoelectric sensors, the thru-beam, retro-reflective and diffused reflective sensor.

Thru-beam sensor, also called opposed sensor, is a constructed as a two device unit, one which transmits the light and one that receives the light. The sensor detects an object when the light beam is interrupted between the two units. The advantages with the thru-beam sensor is that it is the most accurate and has the longest range out of the types of photoelectric sensors. It is also the preferred device in dirty environments making it a good choice in factory environments. Disadvantages is that it needs to be correctly installed. If the setup between the two units is wrong the system will not work as intended. Therefor it is important to install the sensors correct to get a working and functional system (Tri-Tronics, 2020).
Retro-reflective sensor, is built up with the light source and receiver encased into the same unit. The sensor needs a reflector to be able to give signals. The sensor

sends out the light which are reflected and sending it back to the light receiver. To detect an object the sensor gives a signal when the light path is interrupted. The retro-reflective sensor can be used on large objects as well as detecting fast moving objects. The advantage of these sensors is that they are cheaper than thru-beam sensors while also being the best alternative if the materials that needs to be detected are transparent (Tri-Tronics, 2020).

Diffuse reflective sensors, are just as retro-reflective sensors a single unit containing the light source and receiver in the same housing. The difference between the two is that the diffuse sensor does not require a reflector to be able recognise the returning light. The disadvantage due to that comes with the fact that the sensing distance is the lowest between the sensors mentioned in this chapter. So why is this solution used when there is more accurate alternatives? The diffuse reflective sensors are the easiest to install since there is only a single device, the unit is also cost efficient compared to the other sensor alternatives. The drawbacks other than shorter sensing distances is that it can be affected due to color, texture and dirty environments (Tri-Tronics, 2020).

Radio-frequency identification (RFID)

Another solution to monitoring WIP in a production system is to use Radio-frequency identification (RFID) which is a tag mounted on for example a pallet transporting the parts in a production system. The tag is equipped with a small radio transponder which can send and receive signals. When the tag gets triggered by a electromagnetic pulse, the tags sends digital information to a RFID reader. Often this digital data is a identifying number which can be used when for example storing parts in warehouses.

The advantage with using RFID is that the tag can be read without having line of sight, making it superior to barcodes and other scanning solutions (Baysan & Ustundag, 2013). Since there is no need of line of sight to read the tag, RFID provides a production site with less human intervention since the reading can be performed from longer distances. The readings are also fast and accurate so thereby there is no need to stop the line to perform the readings (Baysan & Ustundag, 2013).

Although the RFID technology is good and functional it comes with a price. The hardware cost and mainly the price of the tags is the reason to why RFID is not implemented as much as the technology is capable of in the industry today. There is reusable tags available on the market but they need to be manufactured in more durable materials and there is a need to reprogram them when used with different product types (Baysan & Ustundag, 2013). Also the amount of information needed to be stored in the tag will change the costs, more memory equals a larger cost (Baysan & Ustundag, 2013). Because of the price to the RFID tags these types of systems is most commonly used within closed-loop systems since the tags can often be placed on pallets which will be stationary giving the tag a long lifetime for the same type of products (Ullah & Sarkar, 2020). In open loop systems, using RFID on mass production of cheaper products would therefore most likely heavily increase the cost per product and is hence not a feasible option.

Benefits from using RFID technology is that companies can get increased visibility of were the products are within the system by scanning the tags. By knowing were products are in the factory the accuracy of WIP can be improved. With the information that RFID gathers, companies can use that data to support adjustments and controlling the production (Yin, Gao, & Tian, 2012). It can lead to companies being able to lower the amount of labor and stock needed within the company which would decrease cost significantly (Baysan & Ustundag, 2013).

Vision systems
Detecting and counting products can also be performed by using a vision system. Vision systems uses algorithms and filters to find objects within pictures and can be used to identify and count objects (Huansheng, Haoxiang, Huaiyu, Zhe, & Xu, 2019). A vision system uses a tracking region in which it identifies objects, and then there is a counting line which is used to keep track of the amount of objects which passes the line (Kim, Choi, Choi, & Ko, 2002). With help of algorithms there is possibility to create segments in a picture (Huansheng et al., 2019), these segments creates a possibility to count multiple lines using only one camera.

2.3 Simulation of production systems

Theories of how to proceed when creating a simulation model of a production system in a virtual environment. The chapter also contains how to verify and validate the model against a physical counterpart.

2.3.1 Digital Twin

A digital twin is a simulation model which is real-time based and models an actual product, process or system. The properties of the model makes it capable of describing the actual process dependent on certain conditions and giving the user predictions to how the system works in reality (Schleich, Anwer, Mathieu, & Wartzack, 2017). With implementation of a digital twin companies can use field-level data to decide between different actions to make sure that the system operates in the most optimal way (Kritzinger, Karner, Traar, Henjes, & Sihn, 2018). With an implementation of a digital twin companies can potentially find solutions which increase the accuracy and efficiency of production systems.
The original form of digital twin was constructed to give the user digital information of a physical process which was developed on its own but connected to the physical system. By using digital twins in the design process technicians can try the function and use the system practically before it has been physically assembled (Ungvarsky, 2019). This way the digital twin can be used to find childhood diseases and find solutions to them. This can reduce costly errors compared to finding the errors on a physical product (Ungvarsky, 2019). The digital model should contain all needed information about the physical system, the information could be gained by collecting the data from the real world application (Kritzinger et al., 2018).

2.3.2 Banks Methodology

Banks methodology is a methodology founded to be used as an aid in simulation projects. The methodology breaks down simulation projects to 4 phases consisting of a total of 12 different steps (Banks et al., 2010). Each phase contains several steps which are all shown below in the flow chart in Figure 2.6. The preparation phase consists of steps 1–4, the model building phase consists of steps 5–7, the analysis phase consists of steps 8–10, and the implementation phase consists of step 11–12 (Banks et al., 2010).

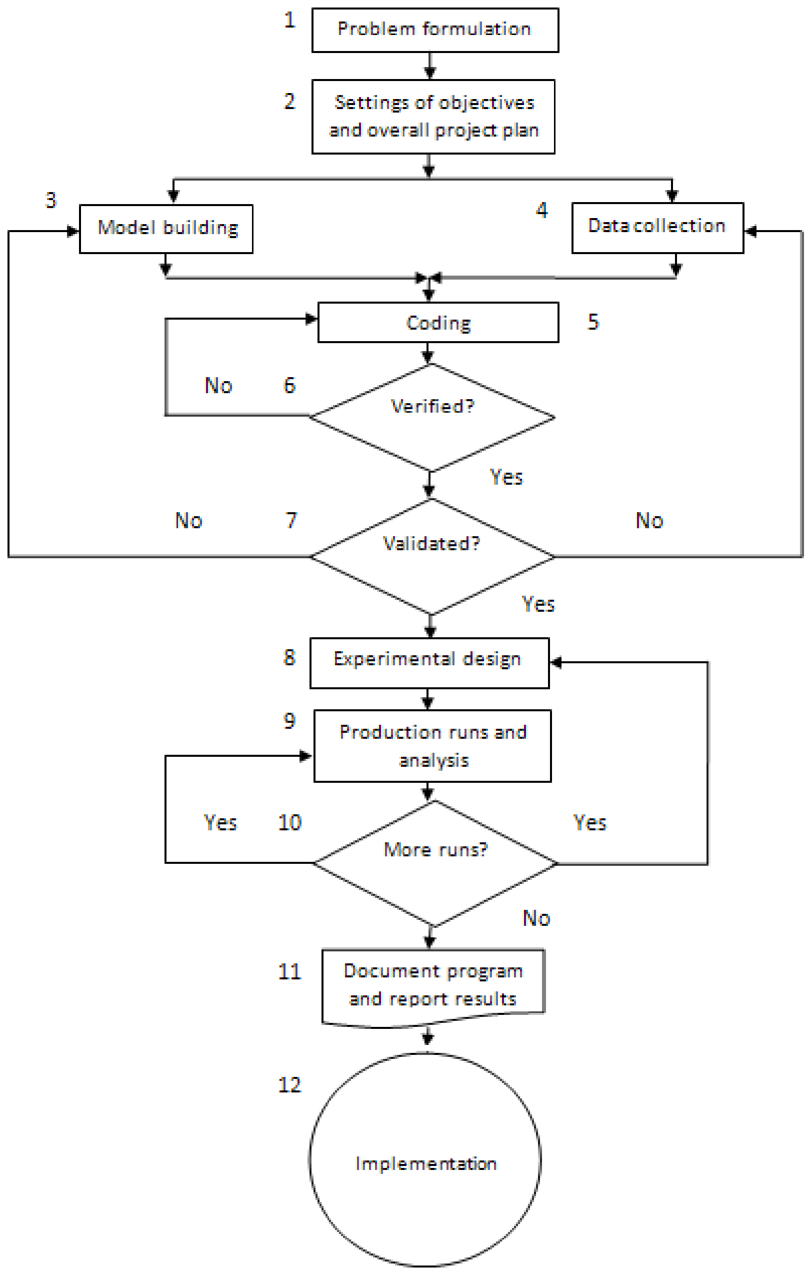


Figure 2.6: 12 steps of a simulation study (Banks et al., 2010)

Problem formulation The first step in the methodology is, as in almost any study, to properly formulate the problem. It is important that all stakeholders agree with the problem formulation to ensure that the correct problem is being investigated. When a problem formulation has been agreed upon the next step is to set objectives and an overall project plan.

Setting of objectives and overall project plan
The project plan should contain the problem formulation, time plan, objectives of the project and which level of detail it should use (Skoogh, 2019). Once a project plan is finished, the project can move on to the two next steps.

Model building
The third step *Model building* or *Conceptual model building* entails building a simple model which describes the system and its logical relations which is usually visualised as a sketch or a flowchart (Skoogh, 2019). The conceptual model is usually changed several times during the project in order to be able to describe the system well enough for the purposes of the project. During these changes it is important not to increase the complexity of the conceptual model more than necessary as increasing the complexity requires more resources for the subsequent steps (Banks et al., 2010). The level of complexity to strive towards should be decided based on the scope and the level of detail previously determined in the project plan (Robinson, 2015).

Data collection
Data collection concerns both the collection and management of input data. How much input data that is needed is determined by the complexity of the model. This means that the data collection should be performed alongside the conceptual modeling or as a iterative process between the two (Banks et al., 2010). Proper data collection is one of the most important steps in a simulation project and often the most time consuming (Skoogh & Johansson, 2008). This is due to the fact that without proper data the simulations will lack credibility, rendering the results useless or even harmful since they may lead to incorrect recommendations (Banks et al., 2010). Since both the conceptual model and the desired input data may change a lot during the course of the simulation project, both these two steps will go on alongside the subsequent steps.

Coding
Once a sufficient foundation has been built in the preparation phase, the *coding*, or *the translation of the conceptual model* may begin. This step entails converting the conceptual model into a computer-recognisable format to be able to simulate it using a computer. This step may be done with or without a specific simulation software depending on what the goal of the simulations are.

Verification
The simulation model itself needs to be verified in order to ensure that it is working as intended. This step is as several of the previous steps often a very iterative process where several iterations of coding and verification normally is needed. The

verification may be performed through many different methods.

Validation
To assert that the verified model accurately reflects the real system, it also needs to be validated. To validate a model it needs to be somehow compared to reality, this can be done in many ways and if there are inaccuracies in the model the developer should go back to either the input data management to assert that it is correct, or to the conceptual modeling to investigate why the model is not accurate enough.

Experimental design
If a company is carrying out a simulation study, there has to be something they want to test or analyse. This means that there should be some form of experiments on the simulation model which should be carried out. In order to do so the analyst should first design the experiments. In this step the analyst should clearly state how the experiments should be carried out, which parameters should be altered and extracted, and how many times the system should be run (Banks et al., 2010).

Production runs and analysis
Once the design of the experiments has been agreed upon it is time to carry out the experimental production runs and analysing them. The runs should be carried out as stated in the experimental design. The analysis may be performed in many different ways and if it is done with simulation software there are often built in functions to aid the analyst in analysing the data.

More runs?
From the analysis the analyst should be able to determine if enough data has been gathered for the objective of the study or if there is a need for more data. If more data is needed the analyst might only need to do more runs but there might also be a need to go back and look over the experiments to ensure that they entail all of the aspects necessary before doing additional runs.

Document program and report results
When the analyst is content with the results from the simulations, it is important to properly document the project and report the results. The documentation should not only include the obvious parts such as methodology and results, but also documentation for the simulation model/program. This allows for the model to be used again, either for other projects or for other to replicate the simulations to get a better understanding for the correlations of the model and the real system (Banks et al., 2010). Having proper documentation for the program also allows for decision makers to try the model themselves which should increase the analyst's credibility. The results themselves is recommended to be continuously reported throughout the project. This allows the people who are not working with the project on a daily basis to still stay updated which allows for misunderstanding, and recommendations to arise early, overall increasing the quality of the results (Musselman, 1998).

Implementation
The final step in a simulation project is *Implementation.* The success of this step is fully dependant on how well all of the previous steps have been performed. A well performed simulation study should have generated enough data and precision to be able to know what changes to the system would be beneficial. If the previous steps have been poorly executed, the data may give false indications. Step 7 is the most prominent step to cause faulty data if it is not performed properly (Banks et al., 2010).

2.3.3 Verification

Verification is the process of checking the model to assert that it is working as intended and it can be done in many different ways (Skoogh, 2019). Simulations with animations may be studied in order to see that entities in the simulation are moving as intended and that they are not disappearing. Using a debugger to be able to pause and step through the execution of code is another good method to use since it allows the developer to assert that the logic is acting correctly based on the current state of the model. Checking the inputs and outputs of the model allows the developer to get a good indication if the model has reasonable results. If the outputs and logic in the model is working as intended, the model should be properly verified (Banks et al., 2010). However, a verified model is not guaranteed to accurately reflect reality. A verified model only means that the model is working as the conceptual model but the conceptual model may very well be too simplified or contain incorrect assumptions.

2.3.4 Validation

Validation is the process of asserting that the simulation model is accurately enough reflecting the real system. And as with verification there are numerous ways of performing this process. Naylor and Finger(1967) proposed a methodology which is recommended by Banks (2010) to help with the validation process which consists of three steps. These steps are:

1. Build a model that has high face validity.
2. Validate model assumptions.
3. Compare the model input-output transformations to corresponding input-output transformations for the real system.

To check the high face validity of the model, it should be checked by persons who have a good knowledge of the system which is being modeled. If several persons whom are well acquainted with the real system thinks that it looks probable then the face validity of the model is high (Banks et al., 2010). Once the model has a high face validity, the model assumptions should be validated, the way to validate these depends a lot on which assumptions have been made. Banks (2010) divides model assumptions into two different categories: structural assumptions and data assumptions. Structural assumptions mainly concern assumptions from the conceptual model while data assumptions mainly concerns assumptions from the data input management. Structural assumptions may be validated through observing the real system to see if they are accurate enough. Data assumptions are validated through controlling the reliability of the input data (Banks et al., 2010). If both the structural and data assumptions are accurate enough, the validity of the assumptions is validated. Once the assumptions are validated, the final validation step is to compare input-output transformations for the real system and the model. This is commonly done by either using the model to predict how the real system will react to specific inputs or by using historical input data to run the model and compare the output to the historical output data. Since most simulation models are created to test the impact that future changes would have on specific parameters, the outputs used to validate the model should be the same as the ones that are to be tested later in the experiments (Banks et al., 2010). Once all of the above validation steps have been completed the model should accurately enough reflect the real system and hence be validated.

3

Methodology

The overall methodology of the thesis followed Banks methodology with some slight modifications to handle the complexity of the simulations (Banks et al., 2010). Since the goal of the final simulations was to have a simulation model controlled by virtual PLCs which were controlled by ALB, four models of the system was built. However, only the final model was validated and used for the experiments later on. This is due to the fact that each model was built to only verify some of the total functionality. The first model was only the basic system without any added control. The next two models were built with ALB and virtual PLCs separately and the final model was with all of the separate pieces together. This allowed for each part to be verified separately and problems with each part could be fixed before the parts were integrated. Even though this required a lot more work before the final simulation model could be built, it significantly lowered the work required to build and verify the final model. Furthermore this methodology assured that the quality of the final model was high. A simplified workflow is presented below in Figure 3.1, the model building blocks in the figure are further explained in section 3.2.

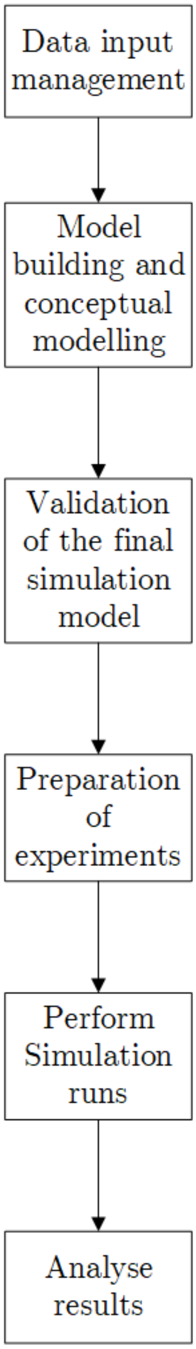


Figure 3.1: Flowchart of the order the methods was executed

3.1 Data input management

As the project began the majority of data about the real system was neither available nor possible to collect since the system was not currently operable. Some examples of this sort of data was the highest speed that it was reasonable to run machines and conveyors with. Both the conveyors and the machines had known maximum speeds that they could be run with. However, the max speed at which they can be run is not the same as the max speed that it is reasonable to run the system with when there are products in the system. Hence these would need to be investigated once the real system was operable they could not be used in the first simulation models. In order to have data to test the simulation models with, the unavailable data was estimated based on the capacities of the motors.

The real system in this project was a mock up made specifically to test the functionality of ALB. This means that several of the parameters were data that had to be decided rather than collected. Two of these were the takt time and failure times. These two were decided to be part of the experiments later on, to give the system different characteristics to test several scenarios. Other parameters such as how long the maximum queue which was allowed was decided together with the PLC-programmer at FlexLink whom have extensive knowledge of their production systems. All of the parameters which was to be determined were added in both the simulations and the real system which meant that those parameters were exactly the same in both versions.

3.2 Model building and conceptual modelling

The work was divided into four sections to ensure that each step of the process got verified before going forward. The approach was chosen to ensure each individual model worked as intended before moving on to the next model. To build the models a conceptual model was created for each model. The conceptual model was then translated and verified before the next model was started. The methodology of the model building is shown below in Figure 3.2.

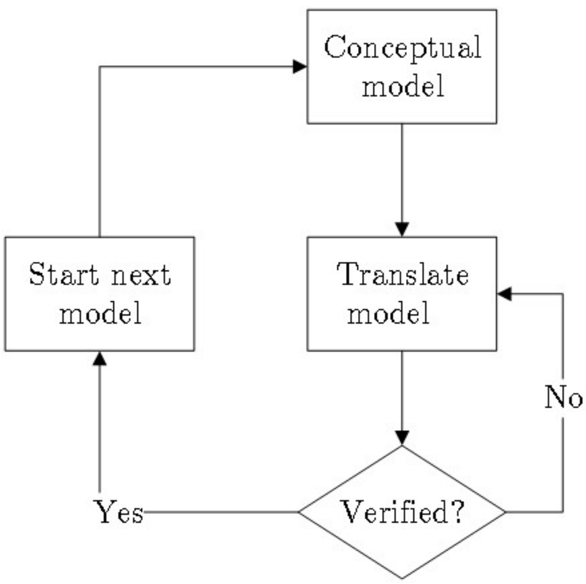


Figure 3.2: Process of verification in model building

3.2.1 Model 1: Base model

The first model was built using Visual Components (VC). Since this model was the base for the subsequent models, a limited amount of logic was implemented. The basic geometry of the test rig was built and the minimum physics required to run it was implemented. This means that the base model only had constant speeds which were set before the simulation started on all of the components and no control logic was implemented. Further information about the test rig can be found in Chapter 4.1.

3.2.2 Model 2: ALB model

The implementation of ALB into FlexLink Design Tool (FLDT) has been tried before at FlexLink but has never been verified with satisfactory results. ALB is written as a python script which can be run independent or implemented into VC. Since earlier implementations of ALB has been using FLDT as simulation software they have needed to use RabbitMQ with an input panel called Human Agent Interface (HAI) to manage the communication between software and simulation. In Figure 3.3 an overview of how the communication was implemented in earlier experiments is shown.

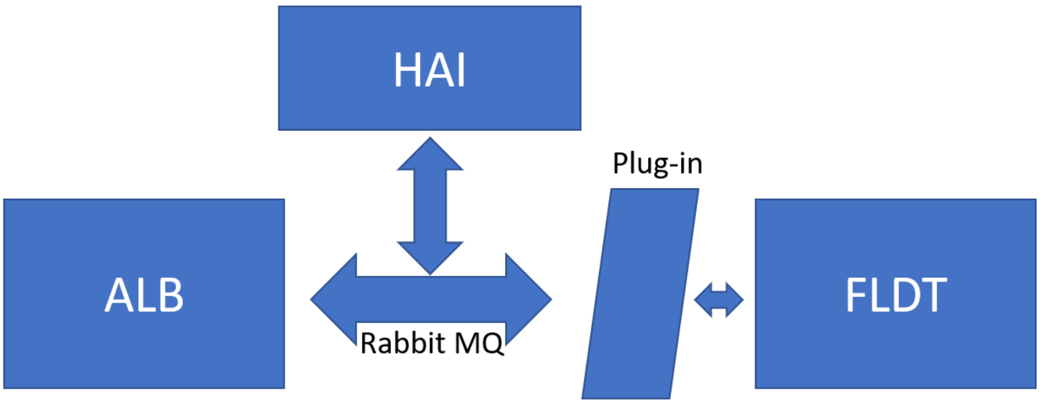


Figure 3.3: Overview of earlier ALB implementation

Although this implementation has been used before and have been demonstrated to work. The amount of time and work it would take to set it up every time as well as having a server to connect to was unnecessary due to the tests needed to be performed was quite small. Therefore, the need to find a simpler and more usable solution arose. It was decided to try with an implementation where ALB would to be directly called from the Python files within VC by translating the scripts to Python 2 and adding them into the site packages in VC. The overview of the solution can be found in Figure 3.4. With this solution, there would not be any need to setup a server for each and every test since there would not be any communication problems between different software languages. The solution of implementing ALB straight into VC was not the most realistic solution in regards to being implemented as a real world application. In a real world implementation there would be some hardware which the data has to go through and there would be a panel for the operator to overlook the production and change input values into ALB dependant on the need in a factory. But, to be able to verify the functionality of ALB it was enough to implement it directly in VC.

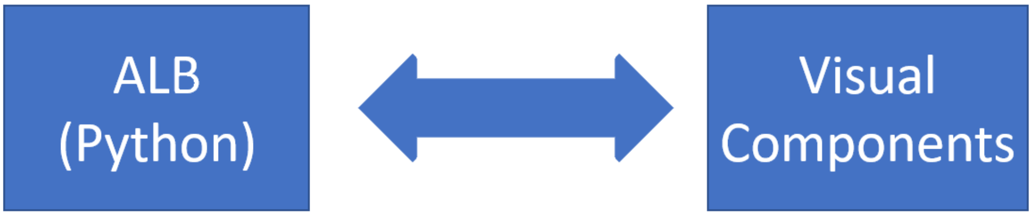


Figure 3.4: Overview of new OPC-UA solution

To create a simulation which works as the real system there was a need to create a component which could act as the steering unit of machines and conveyors. To create such component which would work as a PLCs in the real system the modelling function of VC was used. The components created was named MachineController, SegmentObserver and MainController and these were given functions of what ALB needs to terms of inputs and also a method to receive the speeds which ALB calculates and sends to the operating units. The MachineController is connected to all of the machines in the production line and hold all of the information about the machines (maximum speeds, ID, segment ID, etc.). The MachineController also receives the speeds for the machines from the main controller and sets each machine to the desired speed. A SegmentObserver was also created which worked as a counter which counts the WIP in each segment and also holds the properties of that segment. The MainController is connected to the MachineController, all of the conveyors and all of the SegmentObservers. It collects all of the data that ALB need from the connected components and builds the input files for ALB. When the system is running, MainController continuously calls ALB and then sends the output speeds to the MachineController and all of the conveyors.

The input data needed for the ALB software to work is to receive the WIP data from every segment. If this is not fulfilled the software will be unable to do the necessary calculations. From the WIP values the software will perform calculations and then respond to the system by sending machine and conveyor speeds. The value of these speeds are a percentage value to the max speed of the component which is receiving the data. Thereby, the value sent from ALB to the system will be between 0 and 1.

ALB also needs additional information about the system such as length of conveyors etc. This is fed to the software through a config file. An example of a config can be found in Figure 3.5. These values are given to ALB once but can be changed by an operator depending on how the system should act.

```
#-----------------------------------------------------------------------------------------
# Input data example:
#-----------------------------------------------------------------------------------------
# """{ "nrOfSections":4, "outToConsole":false, "minMachineSpeed":[1.0,1.0,1.0,1.0,1.0],
#     "maxMachineSpeed":[100.0,100.0,100.0,100.0,100.0],
#     "machineCapacity":[600.0,120.0,120.0,120.0,120.0],
#     "targetWip":[12,12,12,12],
#     "maxWip":[50,50,50,50],
#     "actualWip":[79,2,2,1],
#     "minTransportationTime":[[2.4038463],[2.4038463,2.4038463],[2.4038463],[2.4038463,2.4038463,2.4038463]],
#     "maxTransportationTime":[[240.38463],[240.38463,240.38463],[240.38463],[240.38463,240.38463,240.38463]],
#     "segmentsLength":[5.5,3.8,2.4,8.0],
#     "conveyorsLength":[[5.5],[1.7,2.1],[2.4],[2.3,3.4,2.3]],
#     "productLength":[0.05,0.05,0.06,0.05],
#     "targetQueue":[3,3,3,3],
#     "nrOfConveyorsInSegment":[1,2,1,3],
#     "minCycleTimeDefinedByUser":[1.0,1.0,1.0,1.0,1.0],
#     "maxCycleTimeDefinedByUser":[5.0,5.0,5.0,5.0,5.0],
#     "actualFilteredCycleTime":[0.8801812,0.88023835,0.9189492,1.084081,1.1013604],
#     "calculatedCycleTime":[4.0,1.0,5.0,5.0,5.0],
#     "ts":1493981528897,
#     "log":{},
#     "big_FLOAT":9999999.0
#     "measuredCycleTime": float array
#     "productsAfterMachine": int array
#     "calculatedMachineOutput": float array  --- RELATIVE MACHINE SPEED
#     "calculatedConveyorSpeed": array of array of float
# }"""
#-------------------------------------------------------------------------------------------
```

Figure 3.5: Example of how a system config can look like in ALB

3.2.3 Model 3: Virtual PLC

OPC-UA or Open Platform Communications Unified Architecture is the industry standard for secure and reliable exchange of data between machinery. OPC-UA is an independent platform used to create smooth information flows between components from different manufacturers (OPCFoundation, 2020). Since VC has built in support for OPC-UA this allows the user to connect the simulation environment to a OPC server in an easy way as long as the server is set up correctly. The virtual PLCs were run using Siemens PLCsim Advanced software which also has support for OPC-UA. Since FlexLink's standard is to have separate PLCs for conveyors and machines, the test rig has two separate PLCs. This means that in order to make the simulations with VC and virtual PLCs realistic, two separate PLCs were emulated. If both of the PLCs are from FlexLink they will normally be communicating directly with each other. However, it is not uncommon for FlexLink to only be the hardware owner of the conveyor systems in production lines, i.e. the controllers for the processing machines may often be of other hardware. When that is the case, communication between the PLCs can not be guaranteed. Since the direct communication is purely an advantage, having a setup where no direct communication is allowed between the two PLCs means that the results are valid for both of the above scenarios. Hence in the simulations for this project the two emulated PLCs will not have any direct communication. To allow for this setup, both of the emulated PLCs hosts a OPC UA server to which VC have a separate client connecting to each. With the two clients connected to the different servers, the signals from the VC model was paired up manually to the corresponding variables on the server. Overview of the solution can be found in Figure 3.6.

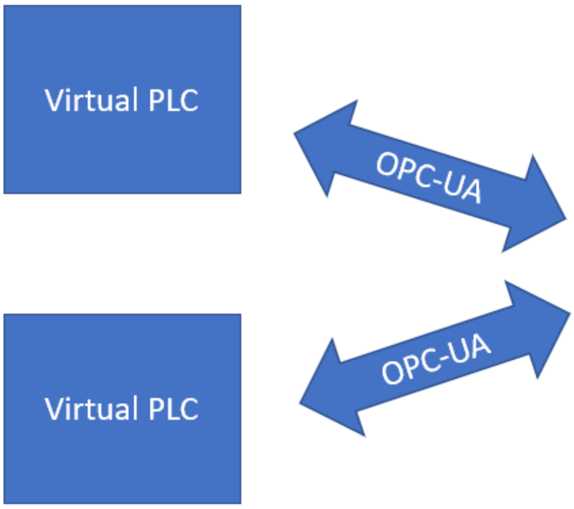


Figure 3.6: Overview of setup Visual Components to virtual PLCs

The process of connecting the variables between the PLCs and VC was done using the User Interface (UI) in VC. The UI is simple and allows the user to browse the contents of the server and the simulation environment to select and pair the desired variables. The PLC however does not allow values in the OPC-UA server to be written directly to inputs which means that separate variables were added in the PLC programs to allow for them to take values from the server and then mirror them to the inputs. One of the problems with this setup is that in the simulation environment it's often desirable to use a high simulation speed to quickly collect a lot of data. But since the PLCs have filter times for the sensors and some delays in their logic, a high simulation speed kills the functionality of the PLCs.

3.2.4 Model 4: Final model

Since ALB lacks several functionalities of a complete control system, it can not properly control a production line without another controller, in this case a PLC. This means that ALB is not allowed to have any direct connection to the production line, it should only exchange data with the PLCs. To replicate this in the simulations, ALB is not allowed to exchange any data with the simulation model. Instead ALB collects data from the PLCs from the OPC-UA servers. In order to do so, a script was written which sets up two Python OPC-UA clients using the Free OPC-UA repository (Roulet-Dubonnet, 2020). The approach works as following, the script collects data from the OPC-UA servers, updates the inputs for ALB, calls the ALB function and writes the outputs from ALB to the servers. While the simulation is active ALB will be running and sending its outputs independently of the production state. The overview of the system can be found in Figure 3.7. The PLCs however have several functions which all override the values from ALB. Examples of functions which overrides ALB are emergency stop, full queues, machine breakdowns etc. This means that ALB is setting the speeds in the production line during the times when the PLCs would be using the "standard speeds" (from HMI or set in their logic), which means that ALB is only complementary to the normal production control.

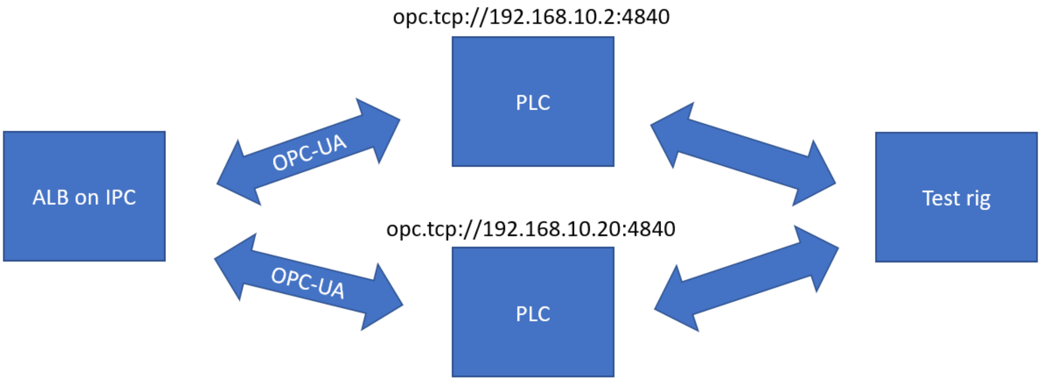


Figure 3.7: Overview of setup between ALB to test rig

For the aspect of testing the functionality and setting up the connections it was easier to have both the emulated PLCs and the Python scripts running on the same device instead of having them on separate computers. In the real world however, the PLCs and ALB would never be run on the same hardware. To mimic that, the connections were also tested when the emulated PLCs were run on one computer and the scripts were run from another computer connecting the two computers through a switch. The test of using multiple devices through a switch ensured that there was no problem with the functionality when the softwares were run from different devices.

3.2.5 Verification of simulation models

Since the four models were built separately to be able to test their individual functionality, they were also verified separately. The first model was verified by several different techniques, the first one was studying the model animation and looking at the statistics of the machines to ensure that they were operating as intended. Beyond studying the animations and statistics, one of the employees at FlexLink continuously inspected the functions to assert that they were correctly coded.

The second model, with ALB integrated, was verified through extensive use of the debuggers and prints to assert that values were changing and being communicated as intended. The calls for ALB itself was checked with the developer of ALB to ensure correct inputs were given to ALB and that the outputs were feasible. Beyond the code checking and prints the model animation was also studied to ensure that the adding of ALB did not obstruct the previously verified functionality from model 1.

The third model, with virtual PLCs, was verified by looking a lot at the data sent from the PLCs and assuring that it reached the simulation model. Several short test runs were made where data was sent between the virtual PLCs and the simulation model to ensure that all data was sent and received correctly.

Since all of the previous models were verified, the final model was just the addition of them all on top of each other, the chunk of verification of the final model was already done. However, even though most of the functions in the model was verified separately, the layering of them had to work as well. The interplay of the functions was verified by making short runs and studying if all of the variables were sent and received correctly. Furthermore, since ALB was now run from a standalone Python script communicating with the PLCs the Python script had to be verified as well. The verification of the Python script was done by using the built in debugger in PyCharm and also by studying the interplay of the PLCs and ALB, asserting that the information from ALB was successfully communicated with the PLCs.

3.3 Validation of the final simulation model

The validation process for the model was performed by doing the same tests in reality as in the simulation environment. The first tests was performed by doing small runs of the simulation model while comparing the behaviours and properties to the test system (Murray-Smith, 2015). The behaviours that was examined was how the product flow acted in the model by looking at spacing between products as well as the flow through the system. This method does not give complete validation, but it can help giving an overview to if the model compares to the real system. When validating there is need for both quantitative and qualitative methods to get a satisfying result (Murray-Smith, 2015).

When the behaviours of the test system and simulation model matched each other the next step was to investigate if the speeds between the two systems were the same. If not the simulation model would need to be adjusted. To do this the two systems, the test rig and simulation, was ran for one hour. The test was done to compare the throughput numbers of the systems and if those did not match, adjust the speeds in the simulation model to match the test rig. The speeds from the PLC program was implemented into the simulation. Although, there was some questions regarding if the machine speeds were credible and had the same speed as the number sent to them. To be able to secure that the machine was working in the intended speed the conveyor after the machine was setup to run at the intended machine speed. The machine was adjusted to match the conveyor speed by changing the gear ratio of the motor which drives the motorised screw and thereby moves the products forward.

As a final validation step the model was run whilst data was collected from the PLCs. The collected data was throughput, WIP values and speeds in the system. The same data was then collected from running the simulation model with the same inputs. These two data sets was compared to determine how accurate the simulation model was.

3.4 Preparation of experiments

The chapter contains explanations of the experiments chosen and how these were conducted. Further there is information of how to gather the correct data for validation of ALB both in simulation in VC as in reality on the test rig.

3.4.1 Experimental plan

The experiments consisted of running the model with constant speeds or with ALB. To give a result on how ALB would perform compared to the constant speed model in several different scenarios, some of the input parameters to the system was changed to slightly change the characteristics of the system. The parameters which were altered were: Takt time, feeding pace and breakdowns. Since the takt time and feeding pace are correlated the feeding pace was set with regard to the takt time. Whilst both takt time and feeding pace are just scalar values, the breakdowns in this regard entails both frequency and length of down times. The experiments contained in total four different breakdown patterns and three different takt times and feeding paces.This gave a total of 12 different systems to be tested both with and without ALB.

The breakdown patterns were determined by both mean time between failures (MTBF) and mean time to repair (MTTR). The length of the times were categorised as either long, medium or short. For MTTF short meant $\approx$ 10-15 s, medium $\approx$ 1 min and long $>$ 4min. For MTBF short meant $<$ 5min, medium was not used and long meant $>$ 15 min. The full breakdown patterns are presented below in Table 3.1.

Table 3.1: Breakdown times for the experiments

Machine	Filler 1		Filler 2		Labeler 1		Labeler 2	
Factor	MTBF	MTTR	MTBF	MTTR	MTBF	MTTR	MTBF	MTTR
Case 1	Long	Medium	Long	Medium	Long	Medium	Long	Medium
Case 2	Short	Short	Short	Short	Short	Short	Short	Short
Case 3	None	None	None	None	None	None	None	None
Case 4	Short	Short	Long	Long	Short	Short	Short	Short

The three takt times chosen and the throughput that they would correlate to are shown in Table 3.2.

Table 3.2: Takt times and corresponding throughput for the experiments

Takt time (s)	Throughput (products/s)
1	1
0.67	1.5
0.5	2

Three takt times and four different breakdown patterns resulted in 12 different test cases to be tested both with and without ALB. The 12 test cases were ordered as shown in Table 3.3.

Table 3.3: All test cases ordered by test ID

Test ID	Takttime	Breakdown case
Test 1	1	Case 1
Test 2	1	Case 2
Test 3	1	Case 3
Test 4	1	Case 4
Test 5	0.67	Case 1
Test 6	0.67	Case 2
Test 7	0.67	Case 3
Test 8	0.67	Case 4
Test 9	0.5	Case 1
Test 10	0.5	Case 2
Test 11	0.5	Case 3
Test 12	0.5	Case 4

To be able to compare the results from ALB and conventional control, the runs with ALB was performed first and then the average speed of each machine and conveyor. The constant speeds in the other simulation was then set from the average speeds of the ALB runs. Setting the speeds like this assumes that the energy consumption is linearly dependent on the speed of which machines and conveyors are run. When it comes to the machines this assumption is very dependent on what type of machine it is and can hence be hard to estimate its accuracy. However, with the conveyors Shirong Zhang and Xiaohua Xia (2009) shows several different models to calculate the energy consumption and all of them are close to linearly dependent on the speed of the conveyors when the other parameters are constant. Hence the linear assumption was deemed accurate enough.

3.4.2 Output data collection

To gather relevant data, two main data types were chosen for collection. The first one was the throughput of the system, this one was chosen since it's a very important factor of how well the system is performing. The second one was the speed of conveyors and machines. This factor was chosen since it directly relates to the energy consumption of the system and that is the main factor that this thesis aims to investigate, if ALB can reduce.

The retrieval of data from the simulation model was conducted by using the statistical function in VC. In the statistical function the user can choose the desired parameters to monitor. In this case the machine and conveyor speeds were chosen to be monitored together with the systems total throughput. The chosen data parameters from the statistical functions are presented as graphs in real-time while running the simulation. When the run is completed the data can be saved as an

Excel-file directly from the software.

To gather data from the test rig a logger function was implemented in the python script. Since the data from the test rig should match the simulation the script was called upon to log the same data as the simulation, those data being machine and conveyor speeds and also total throughput. To be able to control that the sensors used for counting worked as intended, the WIP in every segment was also monitored. The monitoring of WIP is of great importance since the algorithms efficiency decreases significantly if it count on the wrong WIP values. If the sensors used for counting is not good enough, there needs to be a better solution for the WIP counting. The logger function writes all the data to a .txt-file twice a second. With the help of python functions the data from the .txt-file could be extracted in values and then get plotted to be able to compare to the simulations graphs.

4

Test Case

An introduction to the production system built to validate ALB on. The chapter contains explanations to how the system works as well as explanations to how the simulation model was built out of information gathered from the real system.

4.1 The test system

The physical system that is built with the purpose to test ALB on is built as a closed-loop system, even though the ALB software is developed for open-loop systems. The idea behind this is because there is no good solution for how to feed a open-loop system over a long period of time without having someone available to put in products during long runs. This can be troublesome since a test can have thousands of products running through the system and if there is not a constant in-feed the tests will not show the wanted results. Because of that there has to be a simplified solution to solve the feeding problem. In this case the solution is to use a closed loop instead of having the open loop which the system really is made for. Due to the simplification the last conveyor segments will be used as a large buffer which will have a constant speed and will not be steered by ALB. The buffers only function is to bring back the products to the feeder to get an even flow without starvation. The buffer is pointed out in figure 4.1.

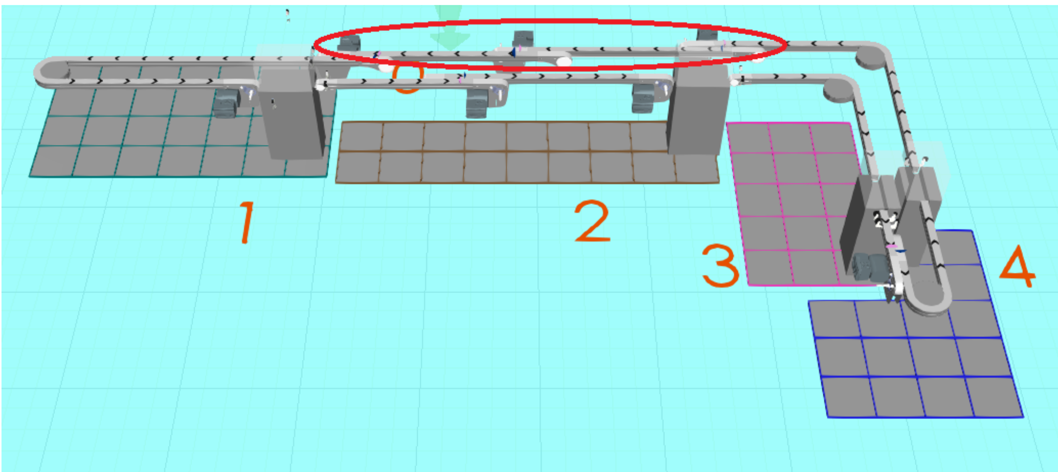


Figure 4.1: Buffer in the system

The feeding to the system is performed by using a pneumatic stop driven with compressed air which can be used to release the product with a specific interval or on signal. The interval can be adjusted by an operator depending on which speed the line needs to be run in. Depending on the speed of the in-feed, the system will adapt due to the change of WIP values in the first segment. The feeder can be viewed in figure 4.2.

Figure 4.2: The pneumatic air stop which controls the in-feed of the system

Around the line there is sensors placed for ALB to maintain knowledge about the WIP count in the different segments of the system. The sensors are placed in the beginning and end of each segment to let ALB know the current WIP in the specific segment. A segment always contains two machines, one in the start and one in the end of the specific segment. Between these machines there can be multiple conveyors moving the products forward. The sensors used in the rig are diffuse reflective sensors which are emitting and receiving light to determine if there is an object passing by and thereby counted by the system. A picture of the used sensors can be found in figure 4.3. Further information about the sensors can be found in subsubsection 2.2.3.1.

Figure 4.3: The diffuse reflective sensors used on the test rig

The test system is operated through two separated PLCs where one handles the machines while the other handles the conveyors. This is a small simplification of a system out in the industry since in this case there is only one PLC handling all the machines. In a real application each machine realistically has its own PLC, the simplification was used since there is no real machines in the test rig. The machines in the test rig are made of motorised screws which can be used to simulate a cycle time. The motorised screw has adjustable speed and can be used to make the system behave as if it would have had machines with an operating time. The screw can be viewed in figure 4.4.

Figure 4.4: One of the motorised screws which is used to mimic a machine

4.2 Simulation model

To make the simulated system as close to reality as possible the correct length of the beams and conveyor segments was needed. Some of the parts are of standard length in the FlexLink catalog such as end-drives, idlers and wheel bends. The lengths of the beams connecting end-drives, idlers and wheel bends are not of standard length and therefor needed to be measured. To find the correct lengths to the system measuring tape was used on the physical rig located in the FlexLink workshop.

Depending on what functionality the model needed to have there was alternatives to which software was most suitable when developing the simulation model. FlexLink wanted the model to be built in their own software FLDT which is developed specifically to give the conveyors the functionality they have in reality. The software is based on the Visual Components (VC) software, but with an E-catalog which only includes products that FlexLink provides to their customers. The product parts in FLDT have added functionality to give the system as realistic function as possible. The advantage of using FLDT over the standard VC software in the simulation would have been that the functionality of the conveyors is better. This is because the parts in FLDT has built in software which understands that the parts connected to each other should have matching properties such as eg. speed and acceleration. The function of the whole conveyor will be affected depending on which value is sent to the end-drive. When it came to build up the machines in the virtual world, FLDT does not have the modelling characteristics needed to build machines which matches the actual system. Therefore, the need of using VC came into fruition anyway. VC has a modelling function where the user can add signals to a component object and these are then given function through Python scripts.

FLDT as a software is really easy to use when it comes to building layouts and conveyor systems. It is based on a drag and drop system where the parts are picked from an E-Catalog existing in the software. When starting to build a layout the user chooses a part from the E-Catalog and drops it in the 3D world in the software. The software will then give the user alternatives to which part should be used to continue the conveyor. For example the user get the alternative to turn the conveyor, make it straight, give it an uphill angle or to end the conveyor. If the user then wants to connect a machine to the conveyor the user simply drags in a machine from the E-Catalog into the 3D world and connects it to the conveyor through the built in snap function in FLDT.

Since VC and FLDT are compatible with each other and uses the same file format, models built in FLDT can easily be imported into VC. Despite that the programs can share layouts the model from FLDT lost some function to how the conveyors should be operating after opening the file in VC. The loss of function happened because VC does not understand that the parts that a conveyor is built up with (End-drive, Idler and Beams) has a relation to each other and gets its inputs from the end-drive. To co-op with that, the software team at FlexLink have developed a component called SpeedChanger which can be connected to a conveyor. The Speed-

Changer will finds all components that are connected in the conveyor configuration and is used as a source which can be used to change inputs to the whole conveyor segment. The SpeedChanger lowers the difference between the model in FLDT and VC respectively to the communication coming through the SpeedChanger-component in VC instead of the end-drive unit in FLDT. When using the SpeedChanger it should be noted that it will try to take control over machine speeds if there is a machine connected to the conveyor. Therefor, it is important to remove unwanted components from the SpeedChangers component list. If not, the simulations will not work as intended. The removal of machines in the component has to be done every time the model is opened in VC.

The VC software also includes a connectivity function from which there is a possibility to connect the model to virtual PLCs to further test functionality towards a real life application. This is convenient since further into the project it will be investigated to how an application with ALB should be able to control one or multiple PLCs in a real context. The reason to why FlexLink wants to know if ALB can be connected to PLCs is because they want to sell the software to companies which already has a conveyor system in place and many of those systems are ran by PLCs. So instead of having to sell the customer new hardware, FlexLink wants to be able to implement new software to the existing system to give a more efficient production.

5

Results

The chapter contains the results from tests and experiments in the simulation model and test rig. It also presents a comparison of the results from the runs with ALB and without.

5.1 Validation results

The final validation was conducted by comparing the machine speed curves from the test rig and simulation model to see if the graphs had similarities and in best case looked more or less alike. During the validation the test rig was run with breakdowns according to case 1 (see Table 3.1) and with the takt time set to 1. The simulation model was run with the same parameters and compared to the test rig. The resulting machine speeds for all of the machines are compared below in Figures 5.1–5.4.

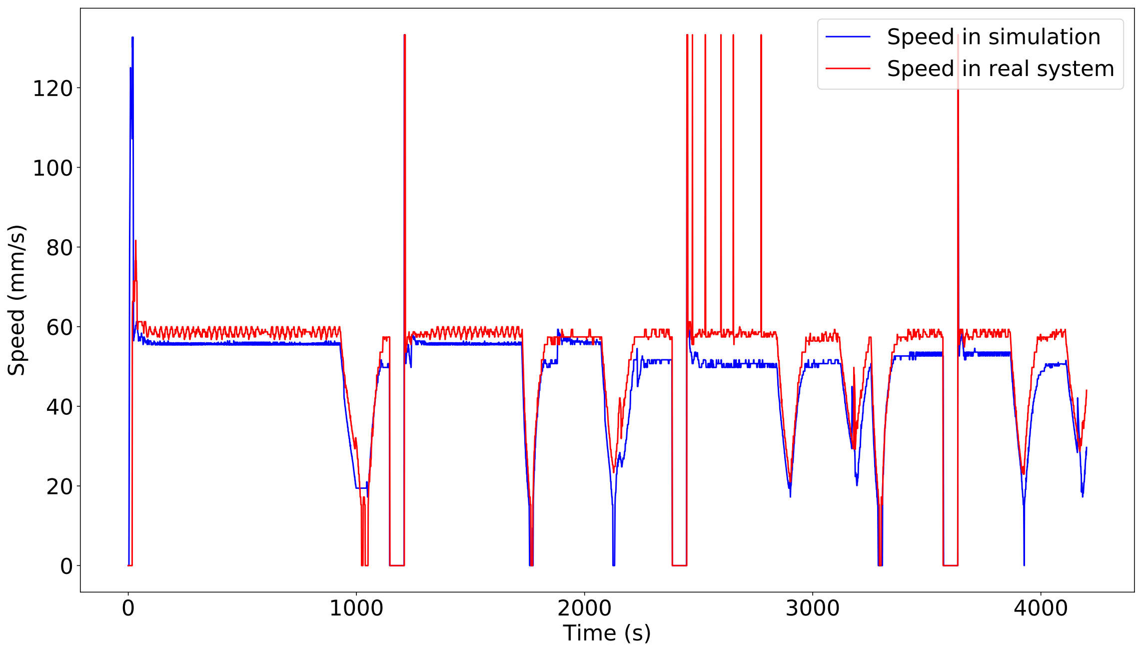


Figure 5.1: Comparison of the resulting speeds for filler 1 in the real system and in the simulation environment.

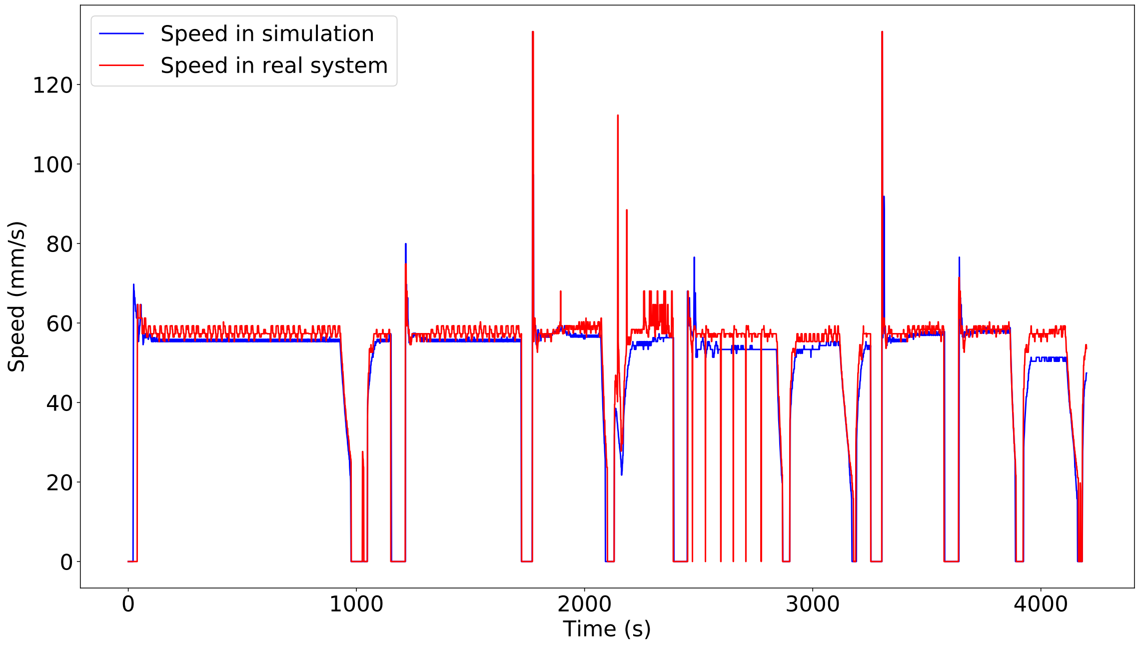


Figure 5.2: Comparison of the resulting speeds for filler 2 in the real system and in the simulation environment.

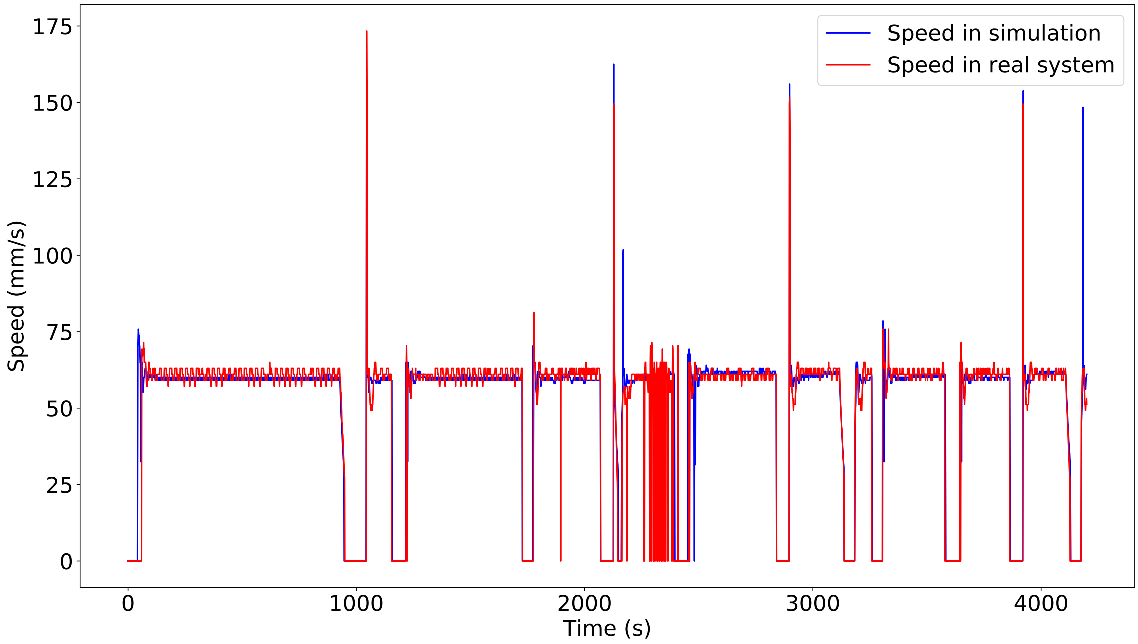


Figure 5.3: Comparison of the resulting speeds for labeler 1 in the real system and in the simulation environment.

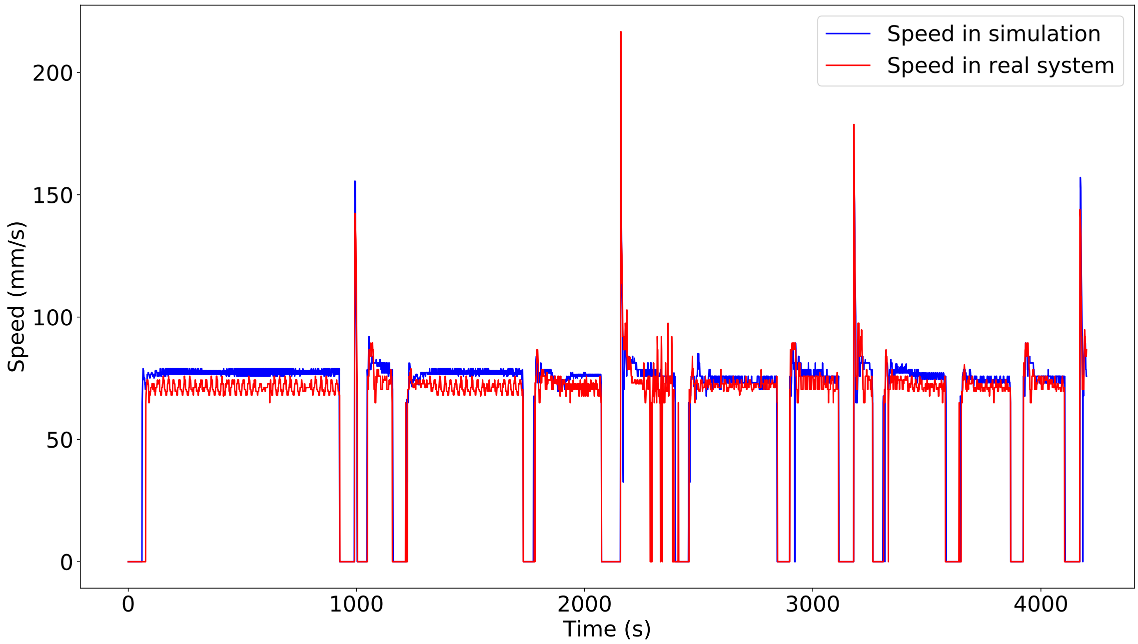


Figure 5.4: Comparison of the resulting speeds for labeler 2 in the real system and in the simulation environment.

The graphs of the machine speeds shows that with the same input the rig and simulation works very similar and appears to have the same behaviours during the run. The real system does have a less stable behaviour with some additional spiking both to high values and to zero. This is mainly because the sensors in the test rig often incorrectly signaled for full queue and stopped the preceding machine, causing all of the machines to react to the unexpected stops. This behaviour was noticed during the run and is caused by faulty sensors and was not considered a deviation between the two models, but rather a fault in the real system.

The throughput was also collected in the real test rig and compared to the throughput from the simulation model. The results from this test is presented below in Figure 5.5.

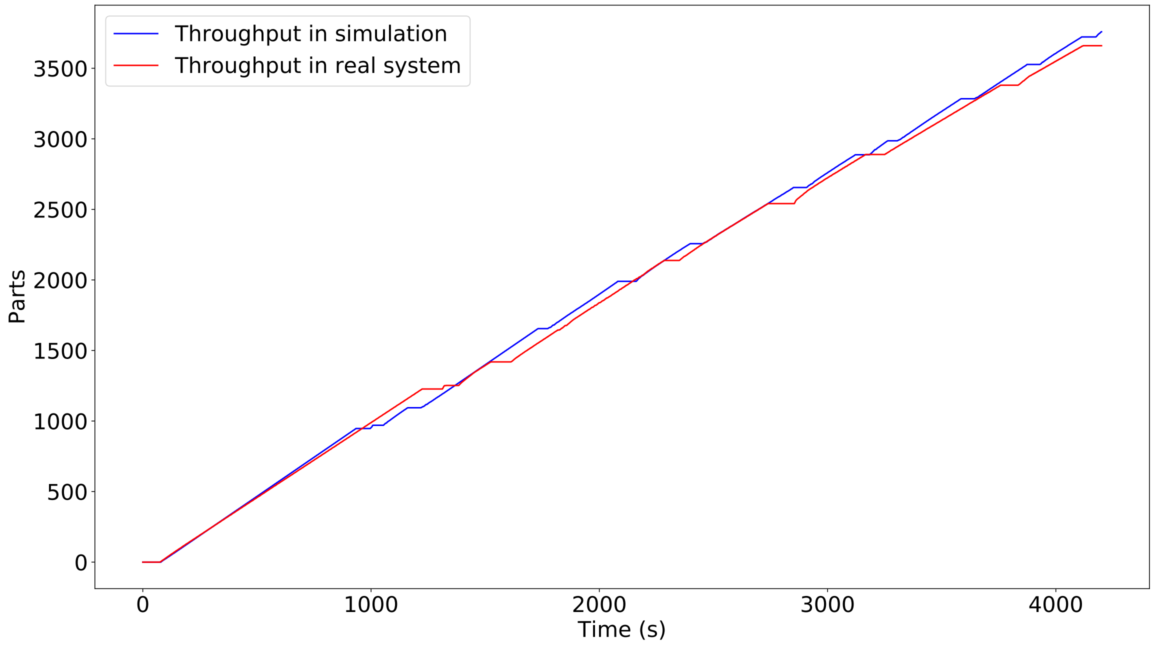


Figure 5.5: Comparison of the throughput from the real system and the simulation model.

The resulting throughput of the two systems were very close to each other whilst their behaviour slightly differed. This is somewhat explained by the fact that the throughput counter in the real system was not placed in the same place as in the simulation model. This causes them to count the products at different points which results in different behaviours. However, since the total throughput from the production run was close, the small difference in their behaviour is considered negligible.

Overall the simulation model very closely resembled the real system and was determined to be of high precision.

5.2 Simulation Results with ALB

In the simulation environment, ALB managed to find a steady state for almost all of the tests. When a breakdown occurred ALB adapted the whole system and after the breakdown ALB reached a new, often the same as before, steady state. In Figure 5.6 a typical example of how the speed of the machines vary during a run is shown.

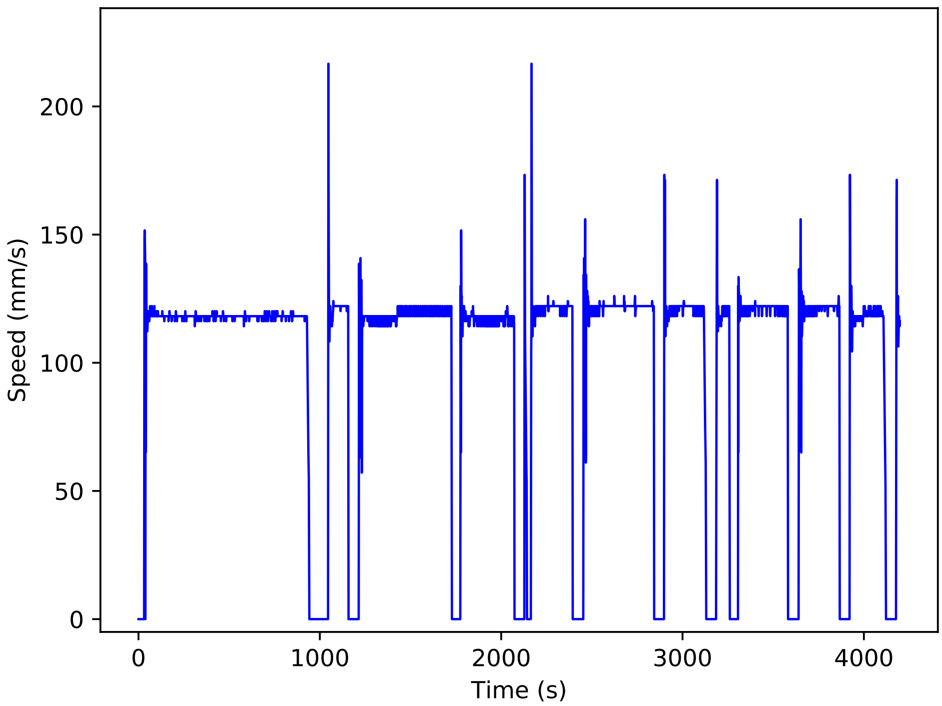


Figure 5.6: Speed curve of labeler 1 from test 9 with ALB.

However, when there is a lot of short failures and the production pace is high, ALB struggles to find a steady state and the speeds oscillate a lot. Figure 5.7 shows a speed curve for filler 2 when it is oscillating a lot.

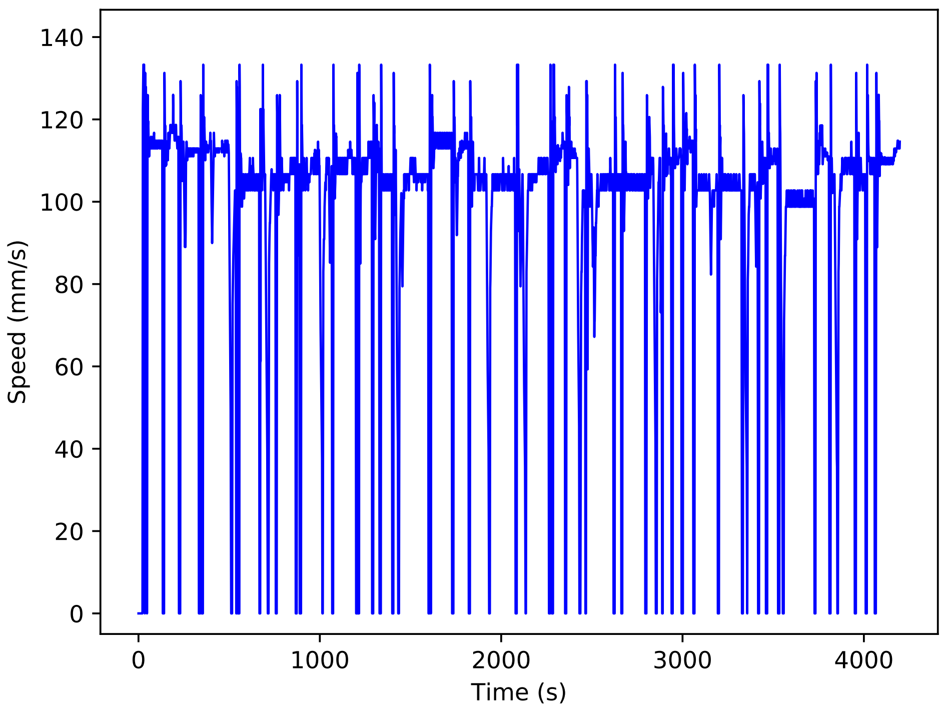


Figure 5.7: Speed curve of filler 2 from test 10 with ALB.

The throughput was collected from all of the tests and compared to the requested output, which is the amount of products that would be produced if the system kept the takt time. The throughput results are presented in Table 5.1

Table 5.1: Throughput from the tests using ALB, compared to the requested throughput.

Test ID	Throughput	Requested throughput	Difference
Test 1	3759	4200	-10,50 %
Test 2	4032	4200	-4,00 %
Test 3	4566	4200	+8,71 %
Test 4	3297	4200	-21,50 %
Test 5	5543	6300	-12,02 %
Test 6	5874	6300	-6,76 %
Test 7	6384	6300	+1,33 %
Test 8	4803	6300	-23,76 %
Test 9	7330	8400	-12,74 %
Test 10	7688	8400	-8,48 %
Test 11	8984	8400	+6,95 %
Test 12	6391	8400	-23,92 %

The system did not manage to produce according to the takt time when there were breakdowns in the system and it slightly overproduced when there were no breakdowns. Although, one reason for this is because there is no algorithm in ALB which makes the system work faster after a breakdown. So if the breakdowns were removed, the system would most likely follow the takt time.

The runs with ALB also showed a smooth behaviour when it reached the steady state that it had. During the steady state, the segments had a constant WIP, which did not change more than up and down with 1 product. The products also had a consistent spacing where they were evenly spread along the conveyor and did not queue up in front of the machines.

5.3 Simulation results without ALB

The test without ALB showed a very rigid behavior where the machines were either running at the set speed or they were down. No behavioural changes were noticed between the different takt times. The speeds normally behaved as the speed curve shows in Figure 5.8

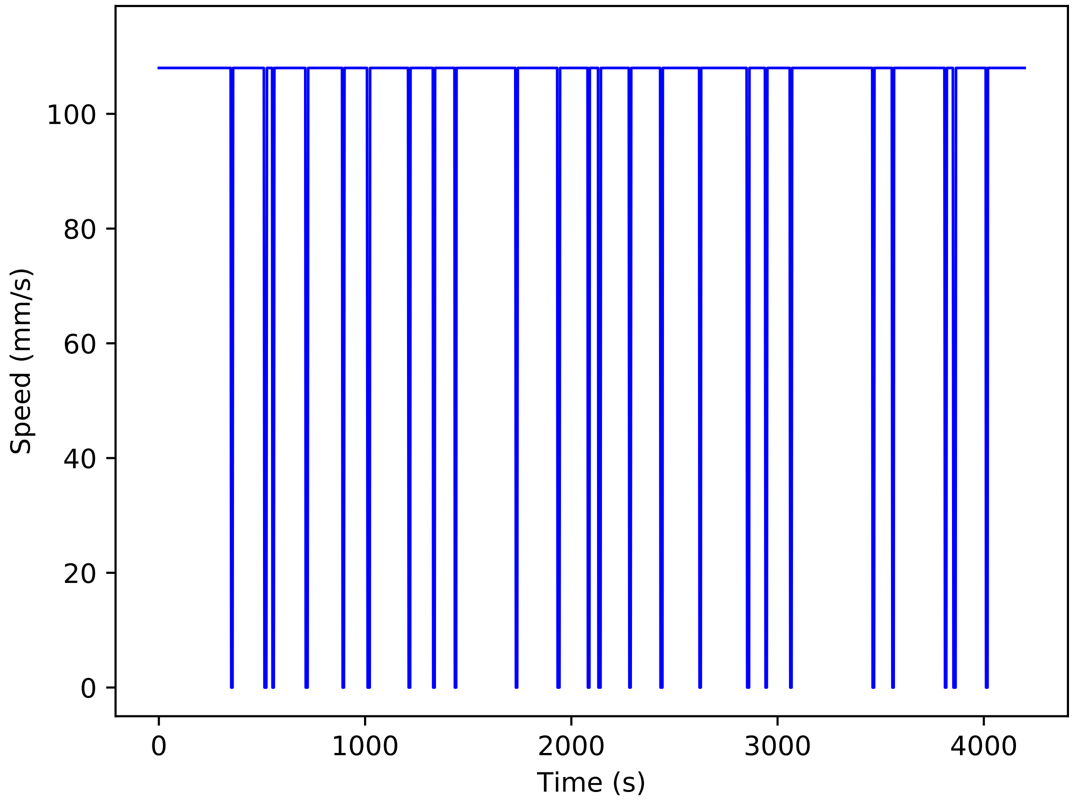


Figure 5.8: Speed curve of filler 2 from test 10 without ALB.

As with the ALB tests, the throughput was collected from each test and compared with the requested throughput. The collected throughput and the comparison is presented in Table 5.1.

Table 5.2: Throughput from the tests without ALB, compared to the requested throughput.

Test ID	Throughput	Requested throughput	Difference
Test 1	3817	4200	-9,12 %
Test 2	3934	4200	-6,33 %
Test 3	4451	4200	+5,98 %
Test 4	3259	4200	-22,40 %
Test 5	5529	6300	-12,24 %
Test 6	5914	6300	-6,13 %
Test 7	6408	6300	+1,71 %
Test 8	4805	6300	-23,73 %
Test 9	7306	8400	-13,02 %
Test 10	7945	8400	-5,42 %
Test 11	8988	8400	+7,00 %
Test 12	6454	8400	-23,17 %

As with ALB, the system did not manage to reach the requested throughput when there were breakdowns and it slightly overproduced when there were no breakdowns. As with ALB there is no function to make up for the lost time when machines are down and that is why the requested throughput was not met when breakdowns were included in the system. The takt time when the system was functional, in between the breakdowns should however be accurate.

5.4 Comparison ALB vs Conventionally steered system

The simulations of the system with and without ALB shows that without breakdowns the systems slightly overproduce to the asked number of products, while not reaching the desired number when breakdowns were implemented. Although, it could be that if the time of the breakdowns were removed, the systems would most likely hold the takt time given.

To be able to estimate the efficiency of ALB the data from the tests steered with and without the control system had to be compared to each other. The data that differed between the tests are listed in Table 3.3. From the tests the corresponding data sets was brought together and compared against each other. The calculation used to express the difference between the data sets can be seen in Equation 5.1. And all the comparisons from the 12 tests can be seen in 5.9.

$$\frac{Throughput_{ALB}}{Throughput_{Traditional}} \tag{5.1}$$

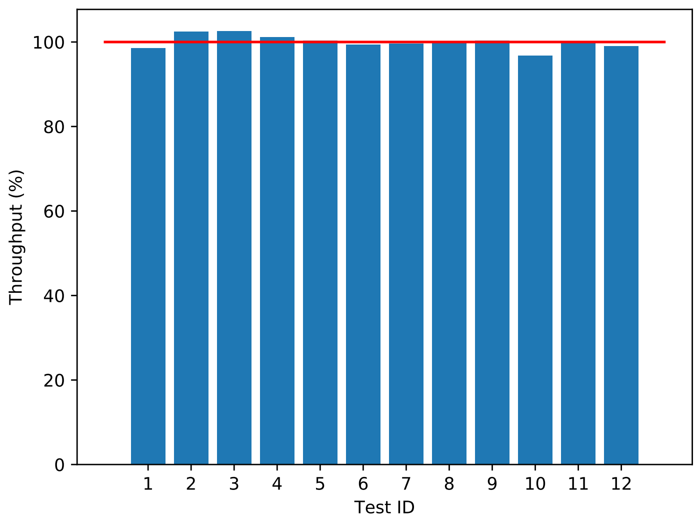


Figure 5.9: Comparison of throughput with use of ALB compared to traditional steering

The results from the comparison of throughput shows that depending on how the system is operating in terms of takt time and the amount of breakdowns affects the throughput of ALB. The data shows that ALB in some situations have a higher throughput and in certain situations a lower throughput. The test result showed that the ALB does not have a clear indication that the throughput will increase using the software. What it does show is that ALB is rather even to a traditionally steered system, which if ALB shows to be more energy efficient is a large improvement.

Since ALB regulates the speeds within the system in regards to the WIP it was important to compare how fast the machines in the system was running when ALB choose the speeds compared to when the PLC program steered the speeds. This data was collected during the same tests as the throughput was collected and then the data sets was compared. The calculation used to express the difference between the data sets can be seen in Equation 5.2. The result which shows the percentage of speed which ALB runs the system with compared to the conventionally steered system. The results can be found in Figure 5.10.

$$\frac{AverageMachineSpeed_{ALB}}{AverageMachineSpeed_{Traditional}} \tag{5.2}$$

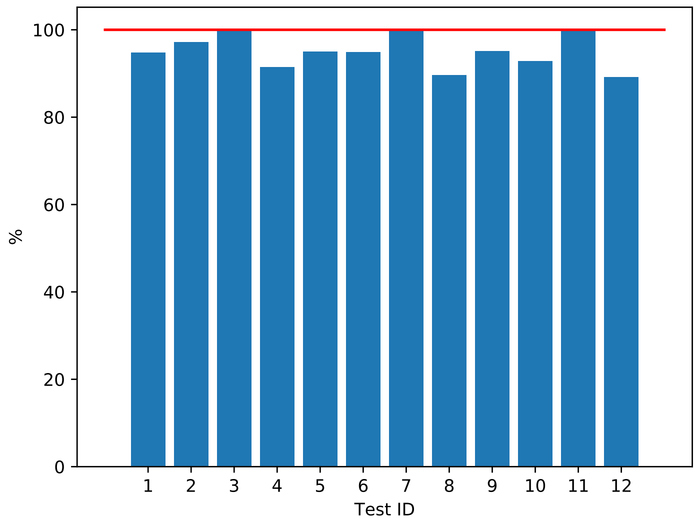


Figure 5.10: Difference of machine speeds in % with use of ALB compared to traditional steering

The comparison between the data sets show that ALB never runs the machines in the system with a higher speed than what the conventional system does. Out of the scenarios tested, ALB showed that in the optimal setting in the test environment it will run the machines 11% slower than a conventionally steered system. While 11% was the best result in the study, the average between all the test showed that ALB managed to lower the speed by 5.03%. The tests also showed that ALB never ran the machines faster than the conventional system as shown in Figure 5.10. This result states that in the right settings ALB can be a valuable asset to companies since it could be used to decrease the energy consumption of a production system with up to 11% according to the study from Zhang and Xia (2009).

Figures 5.9 and 5.10 shows that with the use of ALB the conveyors will be run on average 5.03 % lower speed while the deviation in throughput was on average ±1,14 %. These results shows that there were deviations between the runs with ALB and without ALB but overall neither of them had a significantly higher throughput than the other one. Further discussion on the deviations can be found in chapter 6.2.4. With the lowered machine speeds not only is there an energy factor which will be reduced. The lowered system speeds will lower the sound volume in the factory which will make the work environment more pleasant for people working nearby the system. Also with lower speed to the system, the system life time should increase with a reduce of wear in the machines as well smoother behaviours.

5.5 Summary of results

The simulation model which was of high accuracy showed that with little difference in throughput, ALB can lower the average speeds of a production line with on average 5%. The highest reduction of speeds found in the tests were 11% and the lowest was 0%. The lowest decreases were found in the tests where the system had no disturbances which lead to ALB operating in the same way as a traditionally steered production system. However, for all systems which had disturbances, ALB managed to lower the average speeds of the machines by a minimum of 3%, with minimal impact on the throughput. This means that according to the results in this study, ALB should be an improvement to all feasible systems which have disturbances.

6

Discussion

6.1 Validation of simulation model

The validation process did not fully follow the methodology proposed by Naylor and Finger (1967). The deviation from the methodology was that there were quite few different inputs tested on the real system in terms of takt times and breakdown patterns. This means that there is a risk that the real system and the simulation model would act different in some scenario which has been missed in the validation process. However, since ALB continuously changes the input speeds of the conveyors and machines it was determined that it was enough to test one test case. This meant that instead of doing several different test cases, one test case was instead studied in depth to assert that the system and the simulation model acted in the same way. There is of course a risk that some behaviour that would have come up with the test rig during the other cases have been missed but this is deemed improbable since there is no reason for the system to act differently due to these parameters being changed.

Something that was not presented in the validation results is the problems with the WIP counting in the real system. The WIP values often got offset in the PLCs and had to be manually adjusted during the test runs. When left unchecked for some time, the system could accumulate a lot of extra WIP in the counters. This lead to the system thinking that there were a lot of products on a segment whilst the segment was in fact empty or only had a few products there. Figure 6.1 shows how the WIP in segment 2 changed during a run when the WIP was not manually adjusted.

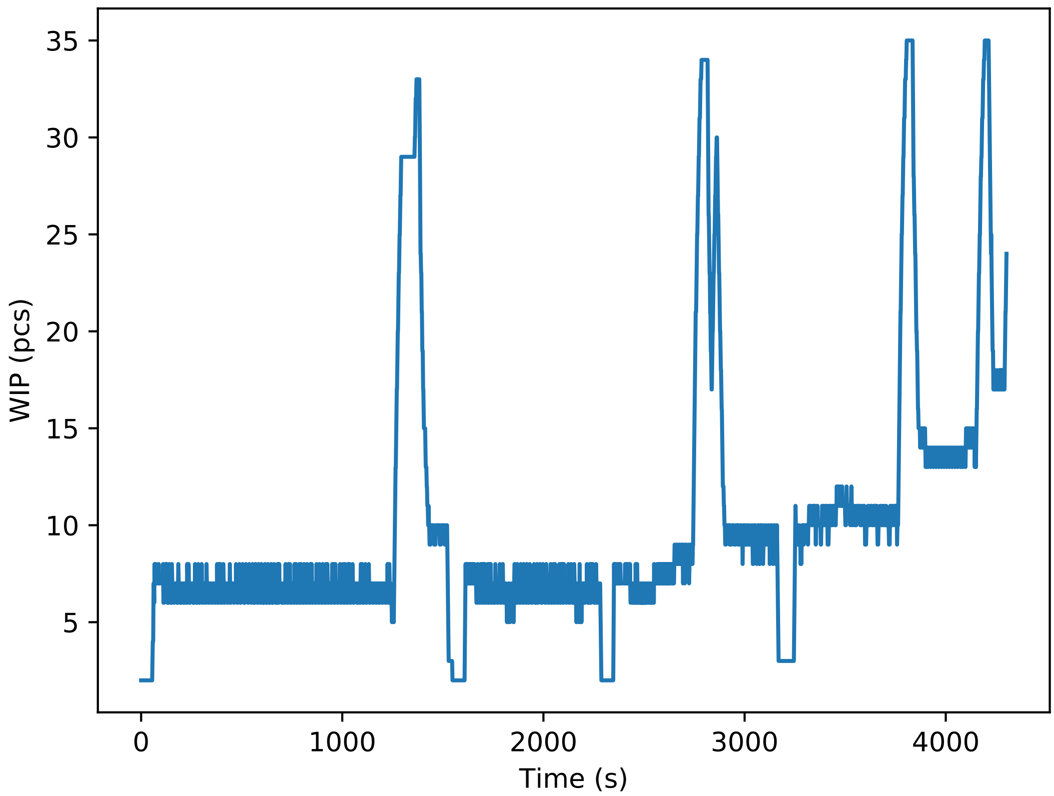


Figure 6.1: WIP of segment 2 when it was not adjusted.

The WIP would, almost always after a queue had been formed, increase by some products. This was not due to more products being in the segment but rather that the counters in the PLCs did not manage to count down when they left the segment. This means that the WIP counting in the test rig was not good enough for running the system with ALB but the simulation models had a perfect WIP counter in order to be able to actually evaluate ALB. This was a intentional deviation between the simulation model and the real system and means that the simulation models have not been validated towards the actual test rig but rather towards the ideal version of the test rig.

6.2 ALB

Within the research and testing of ALB the algorithm has shown advantages and disadvantages within use and implementation. In this chapter some of the specifics of the algorithm has been gathered to be discussed around.

6.2.1 Similar solutions

This thesis evaluated the adaptive control system ALB as a solution for lowering energy consumption in production. However, there are other adaptive control systems and there are other possible solutions to the problem. Siemens have a system which they call Line control unit (LCU) (SIEMENS, 2019). ALB and LCU have very similar functionality, they both change the speeds online to diminish the impact of disturbances. LCU seems to have a more complete set of functions to account for more factors than ALB and is presented as something closer to a finished product. A major difference between them is that ALB wants to perform calculations every second or even several times per second whilst LCU updates in magnitudes of minutes. This should mean that the computational power required for LCU is less. However, no results on how well LCU performs have been found.

Basic on/off control strategies have been shown to reduce the energy consumption of production systems better than ALB (Chang et al., 2013). The on/off control is included in ALB which means that ALB should perform at least as well as a system with only on/off control. Of course this may depend on the system they are tested on but also the reason why ALB does not implement the on/off control as well is because ALB is more realistic, it is blind to a lot of information. Unlike the results presented for the on/off control, no information that ALB would not have in a real scenario has been sent to it. But this shows the potency of adding more information flows to ALB.

6.2.2 Implementation obstacles

It is problematic that the ALB algorithm is so dependent on correct WIP values in all the segments at all times. As mentioned in earlier chapter the sensors used in the test rig combined with the PLC does not have enough accuracy needed for a good implementation in a factory today. To make ALB viable to the industrial market, there is a need to find a solution which secures that the WIP count is correct. Two potential solutions to the WIP counting of the test rig without bigger investments is to try other types of sensors and/or increasing the speed of the input signal to the PLC. The sensors used in the test rig today are diffuse reflective sensors which is a simple type of sensor that do not need a mirror to return the signal. There is a possibility that by using a sensor with higher precision than the current ones there is a possibility to get a sensor solution can work. Another potential reason to why the WIP count gets an offset by missed products could be that the I/Os to the PLC is to slow. In such case the PLC does not have the capacity or priority to read all the inputs that is sent to it. Today the I/Os to the PLCs is through the field distributors a potential upgrade which would solve the problem could be to install an external input card to the PLC. Such input card would have much faster inputs than the ones used today from the field distributor. The PLC could also be setup to prioritise those the signals on the input card. If a smaller investment solution such as new sensors or an external input card does not help correct the WIP count, there is a possibility to investigate in vision systems and see if there is a possibility to combine that with a sensor system. The vision system could be used to secure that the amount of WIP in a segment matches with the value in the WIP counters. If the counters misses a product the system should be able to recognise the differences. The system would then need to have a function to adjust the counter value to be correct with help from the vision system.

When running the system in reality it was noted that after a stop in a machine there were transients to max speed and to zero. A lot of these transients are caused by the fact that the full queue sensors gets triggered. The full queue sensors are a safety feature to make sure that the machines always have empty space on the conveyor to send out products. If a full queue is triggered the machine located before the conveyor will stop. The transients is a behaviour which should be removed if possible since it might affect energy consumption to be worse, but first and for most lower the lifetime of machines due to the behaviour of acceleration from zero to max speed. A possible route for removal of the transients is to use a low pass filter to filter out the transients.

6.2.3 Production rate

As shown in the results, ALB would cause the system to slightly overproduce when there were no breakdowns. This may be fixed by further tweaking the tuning parameters in the inputs given to ALB. However, since the output production pace was rather close to the requested, more tweaking was determined to be too time consuming.

When it comes system which have breakdowns, ALB does not manage to regulate the production pace to hit the requested production rate. This is due to the fact that ALB is blind to when there are breakdowns in the system and also ALB does not know if it is behind or ahead in production. There is currently a second software under development which is meant to add a second feedback loop to ALB. This software would then monitor the breakdowns, throughput, and OEE to learn the correlations of the system. Adding these inputs would allow ALB to account for outer factors and hence be able to recover for lost production or slow down to avoid overproduction. Adding functions which dynamically updates ALB on such parameters would most likely allow it to hit the production rate with a lot higher precision.

ALB has been proven in the results to be able to maintain the same production rate whilst lowering the average speed of the machines. This means that it is very likely that ALB can increase the throughput without increasing the average speed och machines and conveyors in a system.

6.2.4 Deviations in throughput

Each test was supposed to have the same throughput both with and without ALB. This means that the speeds for each run without ALB was calculated based on the results from the runs with ALB. A comparison of the throughput from the tests are presented below in Table 6.1.

Table 6.1: Comparison of the throughput from the tests with and without ALB

Test ID	Throughput without ALB	Throughput with ALB	Difference
Test 1	3817	3759	-1,52 %
Test 2	3934	4032	2,49 %
Test 3	4451	4566	2,58 %
Test 4	3259	3297	1,17 %
Test 5	5529	5543	0,25 %
Test 6	5914	5874	-0,68 %
Test 7	6408	6384	-0,37 %
Test 8	4805	4803	-0,04 %
Test 9	7306	7330	0,33 %
Test 10	7945	7688	-3,23 %
Test 11	8988	8984	-0,04 %
Test 12	6454	6391	-0,98 %

In the tests the largest difference in throughput was measured to be -3.23% and +2.58% but was in most cases within ±1.5% and on average it was ±1.14%. The difference in throughput is considered low enough to claim that they were not significantly different. This means that the resulting speeds of the test are comparable. Of course even lower deviations would have been desirable since that would make the comparison of the speeds even more just, but the extra time needed to decrease the deviations was not justifiable for such a small improvement.

6.2.5 Drawbacks

A drawback of ALB is that the algorithm needs to know the configuration of the system such as lengths of the segments etc. If ALB is to be implemented all the data needs to be extracted and added to the algorithms configuration file. This can be done easily if there is a eg. FLDT model from when the system was originally built. From that model all data could be extracted fairly easy. But if there is no documentation, all data will have to be extracted manually on site.

Another drawback is that to make the system work, is that a WIP counting system has to be implemented to accompany the line itself. Depending of what type of system is used to give ALB accurate values, the implementation will vary in complexity. The test rig that has been used to validate ALB consists of four segments which each includes two sensors to collect the data needed for the algorithm to work. In a production system out in industry the amount of segments might be a lot higher and that would increase work for implementation and setup significantly.

As of today ALB has only been shown to have possibility for implementation on systems where the products are transported as single units on a conveyor. This is due to the choice of trying out a WIP counting system with the use of photoelectric sensors, which should be the cheapest solution of counting WIP. Now when it has been shown that today's sensor solution does not have the accuracy needed there could with new WIP solutions be other systems where ALB could work as well. Depending on the construction and accuracy of a vision system, there might be a possibility that ALB could be functional in systems were multiple products pass a line at once as well.

6.2.6 Suitable systems

As shown in the results, ALB has little or no effect on a system which is not changing and does not have breakdowns. In a system as such, ALB would only be useful for calculating a set value for machines and conveyors, which may just as well be done without using a complex software.

Systems with a lot of dynamic behaviours such as breakdowns and changes in production pace is where ALB would be a good addition to the control system. But as mentioned several times before, in order for ALB to function there are high requirements on the precision of the WIP counting. Which means that ALB today would probably be best suited to be used in systems with a lot of dynamic behaviour and which have RFIDs on the products.

Even though ALB has been developed for open loop systems it was indicated in this thesis that it might also be suitable for closed loop systems. Using it for a closed loop system may however need some additional changes in either the system or in the software.

6.2.7 Effects of lower speeds

The results showed how ALB, with lower average speeds on the machines can still accomplish the same throughput. However, when running the system with ALB there is a lot more fluctuations in the speeds than without. This behaviour is most likely increasing the energy consumption but since no exact energy calculations have been done, it is unclear whether the total energy consumption is lower with ALB or not.
Better estimations of the energy consumption could be done by performing more in depth calculations on data from the simulation model. But to be able to precisely determine the effect ALB has on the energy usage, the best way would be to measure the energy on a physical rig and compare the extracted data from runs with and without ALB.

The lower speeds and the spacing of the products along the conveyors means that fewer products will collide with each other. If they do collide anyway, they will collide with a lower momentum. This should result in lower noise levels in a production line using ALB. Furthermore, running machines and conveyors at lower speeds should prolong their lifespan since it lowers the stress which they are exposed to.

6.3 Future work

Today ALB is not a finished application which can be directly implemented into companies in the manufacturing industry. To be able to present ALB to the industrial market, more work needs to go in to investigating and develop a method for more accurate WIP counting.

As a simplification the feeder solution used in the tests was set to constantly feed the system. To get a more fluent solution the feeder should be connected to ALB which would also create a more stable WIP value in the first segment.

To make ALB a complete product for the market an usable UI should be implemented so that a worker can change the values of parameters directly from ALB instead of having to change the source code. Also, further experimentation with filters to find a consistent stable behaviour of ALB should be done.

7
Conclusion

This thesis has, by applying Banks twelve steps of a simulation study, managed to evaluate ALB as a potential production control system. The thesis has shown that ALB manages to create a smooth behaviour in the system using the solution of counting WIP within predefined segments. However, it has also shown ALB is not suitable for all kinds of systems since several prerequisites needs to be fulfilled for it to be able to operate efficiently. The system also needs to have certain dynamics to create the optimal platform for ALB to bring its benefits.

The shown outcomes of ALB was that it logged lower average speeds of machines and conveyors while still being able to produce the same throughput as a conventionally steered system. The advantages that should come by using the algorithm was to lower energy consumption and noise in the facility while increasing running life of the machines in the system. Although, ALB has also shown that it may cause a volatile behaviour with speed transients if no other control is added. The volatile behaviour may cause the opposite effect on energy consumption, life length and noise due to sudden jumps to zero or max speed of the motors if not regulated in an efficient way.

Finally the thesis shows that with efficient use of an adaptive control system there is a possibility to save energy by changing speeds of machines and conveyors depending of the current situation in the system. This adaptive control system still needs further development to gain accuracy. But in the future adaptive control systems potentially could be the solution for many producing systems to lower their energy usage and environmental impact in order to gain advantages within stacked markets.

References

Ananth, K. N. S., Rakesh, V., & Visweswarao, P. K. (2013). Design and selecting the proper conveyor-belt. *International Journal of Advanced Engineering Technology*, *4*, 43–49.

Banks, J., Carson II, J. S., & Barry, L. (2010). *Discrete-event system simulation.* London: Pearson Education.

Baysan, S., & Ustundag, A. (2013). The cost–benefit models for rfid investments. In A. Ustundag (Ed.), *The value of rfid: Benefits vs. costs* (pp. 13–22). London: Springer London. doi: 10.1007/978-1-4471-4345-1_2

Bellgran, M., & Säfsten, K. (2010). *Production development : design and operation of production systems.* Berlin: Springer Science & Business Media.

Butts, R. (2019). Theory of constraints (toc). *Salem Press Encyclopedia.* Retrieved from http://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=ers&AN=119214150&site=eds-live&scope=site&custid=s3911979&authtype=sso&group=main&profile=eds

Chang, Q., Xiao, G., Biller, S., & Li, L. (2013). Energy saving opportunity analysis of automotive serial production systems (march 2012). *IEEE Transactions on Automation Science and Engineering*, *10*(2), 334–342. doi: 10.1109/TASE.2012.2210874

Cheng, M., Shao, Z., Gao, F., Yang, C., Tong, C., Yang, J., & Zhang, W. (2020). The effect of research and development on the energy conservation potential of china's manufacturing industry: The case of east region. *Journal of Cleaner Production*, *258*, 120558. doi: 10.1016/j.jclepro.2020.120558

Embention. (2020). *Adaptive control and its applications in the industry.* Retrieved 2020-06-01, from url=https://www.embention.com/news/adaptive-control-and-its-applications/

Goldratt, E. M., Cox, J., & Whitford, D. (2004). *The goal : a process of ongoing improvement.* Great Barrington: North River Press. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=cat07470a&AN=clc.a30b98b56a504e6f87de6c89a9b1324d&site=eds-live&scope=site&custid=s3911979&authtype=sso&group=main&profile=eds

Gutowski, T., Murphy, C., Allen, D., Bauer, D., Bras, B., Piwonka, T., . . . Wolff, E. (2005). Environmentally benign manufacturing: Observations from japan, europe and the united states. *Journal of Cleaner Production*, *13*(1), 1 - 17. doi: 10.1016/j.jclepro.2003.10.004

He, D., Pang, Y., & Lodewijks, G. (2017). Green operations of belt conveyors by means of speed control. *Applied Energy*, *188*, 330 - 341. doi: 10.1016/

j.apenergy.2016.12.017

He, D., Pang, Y., Lodewijks, G., & Liu, X. (2018). Healthy speed control of belt conveyors on conveying bulk materials. *Powder Technology*, *327*, 408 - 419. doi: 10.1016/j.powtec.2018.01.002

Holstein, W. K., & Tanenbaum, M. (2012). *Production system.* Retrieved 2020-06-01, from url=https://www.britannica.com/technology/production-system

Huansheng, S., Haoxiang, L., Huaiyu, L., Zhe, D., & Xu, Y. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, *11*(1), 1 - 16. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=edsdoj&AN=edsdoj.7c6934f380df492db7538cfffd3895d1&site=eds-live&scope=site&custid=s3911979&authtype=sso&group=main&profile=eds

Investopedia. (2020). *Work-in-progress (wip).* Retrieved 2020-06-01, from url=https://www.investopedia.com/terms/w/workinprogress.asp

Jia, Z., Zhang, L., Arinez, J., & Xiao, G. (2016). Performance analysis for serial production lines with bernoulli machines and real-time wip-based machine switch-on/off control. *International Journal of Production Research*, *54*(21), 6285 - 6301. Retrieved from http://proxy.lib.chalmers.se/login?url=http://search.ebscohost.com.proxy.lib.chalmers.se/login.aspx?direct=true&db=buh&AN=118439170&site=ehost-live&scope=site

Kim, J.-W., Choi, K.-S., Choi, B.-D., & Ko, S.-J. (2002). Real-time vision-based people counting system for the security door. In *International technical conference on circuits/systems computers and communications.* 2002, Phuket, Thailand, pp. 1416-1419..

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, *51*(11), 1016 - 1022. doi: 10.1016/j.ifacol.2018.08.474

Li, J., Lennartson, B., Tang, Y. G., Biller, S., & Matta, A. (2017). *Sustainable production automation.* New York: Momentum Press. Retrieved from https://ebookcentral.proquest.com/lib/chalmers/detail.action?docID=4786752

Mashaei, M., & Lennartson, B. (2013). Energy reduction in a pallet-constrained flow shop through on–off control of idle machines. *IEEE Transactions on Automation Science and Engineering*, *10*(1), 45-56. doi: 10.1109/TASE.2012.2225426

Mosadegh, H., Fatemi Ghomi, S., & Süer, G. (2017). A control theoretical modelling for velocity tuning of the conveyor belt in a dynamic mixed-model assembly line. *International Journal of Production Research*, *55*(24), 7473 - 7495. Retrieved from http://search.ebscohost.com.proxy.lib.chalmers.se/login.aspx?direct=true&db=buh&AN=127420756&site=ehost-live&scope=site

Murray-Smith, D. J. (2015). Methods for the invalidation/validation of simulation models. In *Testing and validation of computer simulation models: Principles, methods and applications* (pp. 85–111). Cham: Springer International

Publishing. doi: 10.1007/978-3-319-15099-4_7

Musselman, K. J. (1998). Guidelines for success. In J. Banks (Ed.), *Handbook of simulation: Principles, methodology, advances, applications, and practice* (pp. 721–743). Georgia: John Wiley & Sons.

Naylor, T. H., & Finger, J. M. (1967). Verification of computer simulation models. *Management science*, *14*(2), 92–106.

OPCFoundation. (2020). *What is opc?* Retrieved 2020-06-01, from https://opcfoundation.org/about/what-is-opc/

Rahman, S.-u. (1998). Theory of constraints. *International Journal of Operations & Production Management*, *18*(4), 336–355. doi: 10.1108/01443579810199720

Robinson, S. (2015). A tutorial on conceptual modeling for simulation. In *Proceedings of the 2015 winter simulation conference, 2015.* Huntington Beach, USA, pp. 1820–1834.

Roser, C., Nakano, M., & Tanaka, M. (2003). Comparison of bottleneck detection methods for agv systems. In *Proceedings of the 2003 winter simulation conference, 2003.* New Orleans, LA, USA, pp. 1192–1198. doi: 10.1109/WSC.2003.1261549

Roulet-Dubonnet, O. (2020). *Free opc–ua.* Retrieved 2020-14-06, from https://github.com/FreeOpcUa/python-opcua

Schleich, B., Anwer, N., Mathieu, L., & Wartzack, S. (2017). Shaping the digital twin for design and production engineering. *CIRP Annals*, *66*(1), 141 - 144. Retrieved from http://www.sciencedirect.com/science/article/pii/S0007850617300409 doi: 10.1016/j.cirp.2017.04.040

SIEMENS. (2019). *Line integration at the food beverage industry - line control unit (lcu).* Retrieved 2020-14-06, from https://support.industry.siemens.com/cs/document/109762454/line-integration-at-the-food-beverage-industry-line-control-unit-(lcu)?dti=0&lc=en-AZ

Skoogh, A. (2019). *Project methodology.* [PowerPoint presentation]. Retrieved from https://chalmers.instructure.com/courses/7313/files/folder/Lectures?preview=257887

Skoogh, A., & Johansson, B. (2008). A methodology for input data management in discrete event simulation projects. In *Proceedings of the 2008 winter simulation conference, 2008.* Miami, USA, pp. 1727-1735.

Tri-Tronics. (2020). *What is a photoelectric sensor?* Retrieved 2020-06-01, from url=https://www.ttco.com/what-is-a-photoelectric-sensor

Ullah, M., & Sarkar, B. (2020). Recovery-channel selection in a hybrid manufacturing-remanufacturing production model with rfid and product quality. *International Journal of Production Economics*, *219*, 360 - 374. doi: 10.1016/j.ijpe.2019.07.017

Ungvarsky, J. (2019). Digital twin. *Salem Press Encyclopedia of Science.* Retrieved from http://search.ebscohost.com/login.aspx?direct=true&AuthType=sso&db=ers&AN=133861316&site=eds-live&scope=site&custid=s3911979&authtype=sso&group=main&profile=eds

Wittenmark, B. (1988). Implementation and application of adaptive control. *IFAC Proceedings Volumes*, *21*(7), 9 - 14. doi: 10.1016/S1474-6670(17)53793-3

Yin, C., Gao, Q., & Tian, J. (2012). Wip tracking and monitoring system based on rfid. In *2012 third international conference on intelligent control and information processing, 2012.* Dalian, China, pp. 365-368. doi: 10.1109/ICICIP.2012.6391469

Zhang, P. (2010). Chapter 2 - industrial control engineering. In *Advanced industrial control technology* (p. 41 - 70). Oxford: William Andrew Publishing. doi: 10.1016/B978-1-4377-7807-6.10002-6

Zhang, S., & Xia, X. (2009). A new energy calculation model of belt conveyor. In *Africon 2009.* Nairobi, Kenya, pp. 1–6. Retrieved from `https://ieeexplore-ieee-org.proxy.lib.chalmers.se/document/5308257`

Zhang, S., & Xia, X. (2010). Optimal control of operation efficiency of belt conveyor systems. *Applied Energy*, *87*(6), 1929 - 1937. doi: 10.1016/j.apenergy.2010.01.006

Şimşit, Z. T., Günay, N. S., & Özalp Vayvay. (2014). Theory of constraints: A literature review. *Procedia - Social and Behavioral Sciences*, *150*, 930 - 936. doi: 10.1016/j.sbspro.2014.09.104