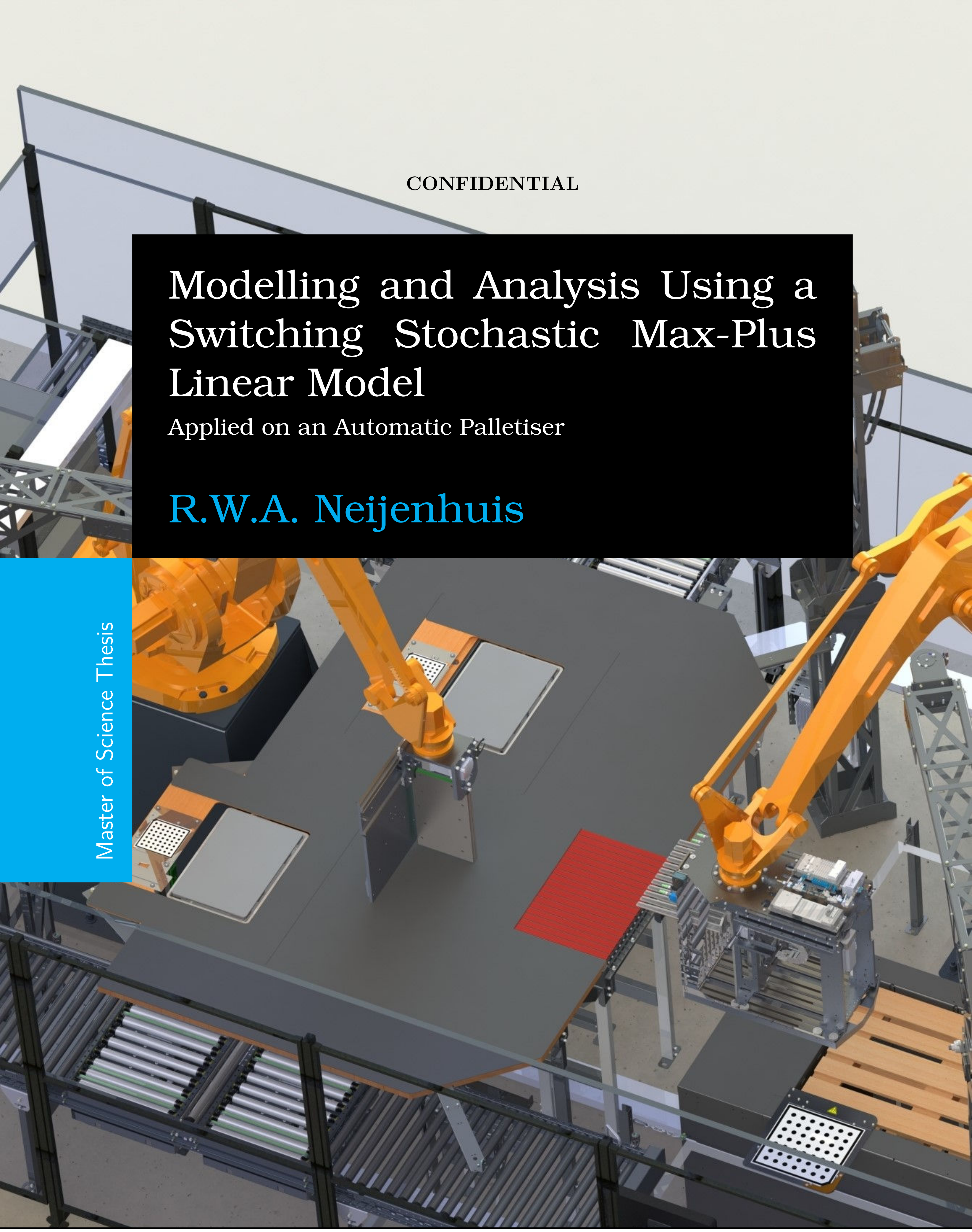


CONFIDENTIAL

Modelling and Analysis Using a Switching Stochastic Max-Plus Linear Model

Applied on an Automatic Palletiser

R.W.A. Neijenhuis

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

CONFIDENTIAL

Modelling and Analysis Using a Switching Stochastic Max-Plus Linear Model

Applied on an Automatic Palletiser

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

R.W.A. Neijenhuis

11th July 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Vanderlande Industries. Their cooperation is hereby gratefully acknowledged.

Abstract

Automated Case Picking (ACP) is a concept for warehouse automation made by Vanderlande Industries. A subsystem of the ACP concept is the automatic palletiser that is responsible for handling cases from a tray onto an order carrier. The research questions concern whether the automatic palletiser can be modelled as a Switching Stochastic Max-Plus Linear (SSMPL) model and what conclusions can be drawn on capacity, bottleneck and sensitivity issues.

The physical components of the automatic palletiser that are handling the cases are in succession a conveyor system, a tray lift, a Tray Unloading Robot (TUR), a PickUp Table (PUT), a Palletiser Robot (PR) and a carrier lift. These physical components and several software and vision based components are depending on each other and together determine the behaviour of the system.

The automatic palletiser is treated as a Discrete Event Dynamic System (DEDS) and is linear in the max-plus algebra as it contains synchronisations and no concurrency. Structural changes within the system are solved by describing it as a Switching Max-Plus Linear (SMPL) system. The system switches between modes that describe a certain system structure.

The automatic palletiser is described by an SSMPL model in which stochastic parameters are used to describe the actions that take a variable amount of time. The model is written in a state-space representation which makes it suitable for implementation into MATLAB. Timing variables are based on simulation data performed by Vanderlande. Simulation of the model results in a structural difference with the reference simulation as a result of the stochastic characteristics. Cross-validation shows that the model gives the expected result when the identification and the test data are different.

Analyses on activeness of the robots, cycle times and on critical cycles conclude on the TUR being the system's bottleneck. It is recommended to research the change in accuracy due to fastening the robot.

Research on Max-Plus Scaling (MPS) functions matrix multiplication makes it possible to easily describe a sequence of max-plus matrix multiplications in terms of MPS functions. This is useful in simulation applications as performed in this thesis. However, due to characteristics of the internal cycle times of the system, the automatic palletiser in the present form does not qualify for implementation of MPS functions.

Contents

Acknowledgement

Nine months of work come to an end. This is not only the ending of my thesis project, but also the ending of my student life. The years in Delft were very nice and challenging. My expectation is that the future years will be different but also nice and challenging, I am looking forward to move on and face the coming opportunities. I would like to take this opportunity to thank some people.

Firstly, I would like to thank my parents for helping me whenever needed. I really appreciate your interest and support, both in study and life. Also thanks to my brothers, who I see as true friends.

Next, thanks to Ton van den Boom for being my supervisor and helping me in fulfilling this thesis. We had nice discussions on max-plus algebra, but also your thoughts on improving application issues were very helpful. Also thanks to Joost van Eekelen for being my supervisor from Vanderlande. Your supervision allowed me to make my own choices, but when needed, you put me back on track.

Lastly, thanks to my colleagues at Vanderlande who contributed in the fun I had in going to Veghel every day. And thanks to my friends who made my Delft experience unforgettable.

Chapter 1

Introduction

Vanderlande Industries is a company making added-value logistic process automation system. One of the systems made by Vanderlande is the automatic palletiser. This system is capable of placing delivered cases on a stack in an optimal, predetermined way. Modelling the automatic palletiser is possible in max-plus algebra, an algebra having maximisation and addition as the only operators.

There are two main goals of this thesis. One is to model the automatic palletiser in max-plus algebra. The other is to perform analyses on capacity, bottleneck and sensitivity of the automatic palletiser.

1-1 Context

Vanderlande Industries is the global market leader in baggage handling systems for airports, and sorting systems for parcel and postal services. The company is also a leading supplier of warehouse automation solutions [1]. Vanderlande is running a concept on warehouse automation called Automated Case Picking (ACP). The goal of ACP is to provide the tools to outperform traditional order fulfilment methods. Next to that, the automation is done in such a way that also the efficiency of work at stores where the orders are delivered is improved [2].

A subsystem of the ACP concept is the automatic palletiser. The palletiser is considered responsible for handling trays supplying ordered cases, handling order carriers and stacking the supplied cases onto an order carrier (not necessarily a pallet), potentially including slip sheet insertion, final wrapping and labelling, ready for further transport and handling [3]. Capacity issues of the palletiser are of interest to the company as this is an important characteristic for potential customers. Simulations are performed in order to make estimations on capacity. The simulations performed by Vanderlande make use of software provided by the robot manufacturer which produces very accurate simulation results of the robot movements. A drawback of this simulation is that the fastest simulation speed is only twice as fast as reality.

Man-made systems that consist of a finite number of resources (processors or memories, communication channels, machines) shared by several users (jobs, packets, manufactured objects) which all contribute to the achievement of some common goal (a parallel computation, the end-to-end transmission of a set of packets, the assembly of a product in an automated manufacturing line) are usually referred to as a Discrete Event Dynamic System (DEDS) [4]. The automatic palletiser can be seen as a DEDS:

- Finite number of resources: Several cooperating robots
- Several users: Ordered cases
- Common goal: Getting stacked onto an order carrier

A model that describes a DEDS is usually non-linear in conventional algebra. However, there exists an algebra in which DEDSs that do not involve concurrency can naturally be described by a linear model [4], that is the max-plus algebra. In the application of the palletiser, concurrency appears when an ordered case must choose among the robots. Concurrency situations do not occur for the palletiser, and so it applies for being linear in max-plus algebra. Systems described in max-plus algebra are usually written in a state-space form with a fixed structure. The structure of the palletiser is not the same for every ordered case which can be solved by adding switching to the state space model [5].

1-2 Problem Statement

The goal of this thesis project is to perform analyses on the automatic palletiser using max-plus algebra. Therefore, the main research question of this thesis is:

Is it possible to model and analyse the ACP automatic palletiser with the use of max-plus algebra?

In order to answer this main research question, it is split into two sub-questions. The former sub-question concerns the modelling part, the latter concerns the analysis part.

Although Vanderlande's palletiser has the correct characteristics to be modelled in max-plus algebra, possibly the complexity of the system can cause difficulties. In order to test this the first sub-question is:

Can Vanderlande's ACP automatic palletiser be modelled correctly using a Switching Stochastic Max-Plus Linear model?

By correctly it is meant that the Switching Stochastic Max-Plus Linear (SSMPL) model results validate the reference simulation results. The established model can be used for many different objectives and in many different ways. A few are chosen to perform in this thesis which is the reason for the second sub-question:

After establishing a Switching Stochastic Max-Plus Linear model, what can be said on capacity, bottleneck and sensitivity of the ACP automatic palletiser?

Bottleneck and sensitivity analysis will result in data that can be used in order to adjust the system such that a higher capacity can be reached by the automatic palletiser. Vanderlande's goal is that the automatic palletiser reaches a capacity of 800 cases per hour.

1-3 Outline of Thesis

Firstly, general information is given on the automatic palletiser system and on max-plus algebra. Chapter 2 gives an overview of the palletiser system and the interfaces linked to it. Furthermore, the different components of the palletiser are explained in detail. Chapter 3 provides the theory on max-plus algebra, the applications and modelling techniques.

Next the performed implementations and analyses are described. Chapter 4 describes how the palletiser is modelled as a SSMPL model and how the correctness is validated. Chapter 5 states the analyses performed with the established SSMPL model, this includes bottleneck and sensitivity analysis.

Next Chapter 6 is dedicated to Max-Plus Scaling (MPS) functions and some applications. Although the theory is promising, the palletiser system is not suitable for useful implementation in an MPS function model. However, some interesting applications are investigated and explained.

Finally, a conclusion of the thesis is presented in Chapter 7. Here, also recommendations for future research are presented.

Chapter 2

The Automatic Palletiser

Automated Case Picking (ACP) is a concept which designs an integrated system that helps customers to pick certain orders from a warehouse, automatically. A customer who has a warehouse is able to teach the ACP system on the cases that are stored within the warehouse. The storing of cases in a warehouse is also done automatically. Consequently the system is able to handle certain orders. This is done by pulling the required cases out of the warehouse in a certain order such that they can be placed on pallets (or other carriers) in an optimal way. Then these stacked carriers are ready to leave the warehouse and go to the designated location.

This chapter gives an insight in a subsystem of the ACP system: The automatic palletiser. The theories used in this study are applied on this palletiser in Chapter 4. From now, “the system” refers to the automatic palletiser. The system has the purpose to handle cases that come into the system in trays. The goal of every case is to finally be located at the correct place on a carrier, this place is predetermined and optimised. The specifications of these carriers are adaptable depending on the customers’ demands. Section 2-1 gives an overview of the system. Section 2-2 zooms in and specifies the different components of the system. Details of the system are taken from [3].

2-1 Overview

The goal of the system is to put the input of the system, which are the cases, on the correct position on the carrier. In between there are several machines that handle the cases in their own way. First of all, a notion has to be made on the way that the cases are handled while they enter the system. The cases are stored with the use of trays, multiple similar cases can be stored in one tray. However, not all cases from a tray are necessarily needed, the tray containing the unnecessary cases leaves the palletiser and goes back into the greater system. Next to that, there is no initial preference indicated by the system for particular cases within one tray.

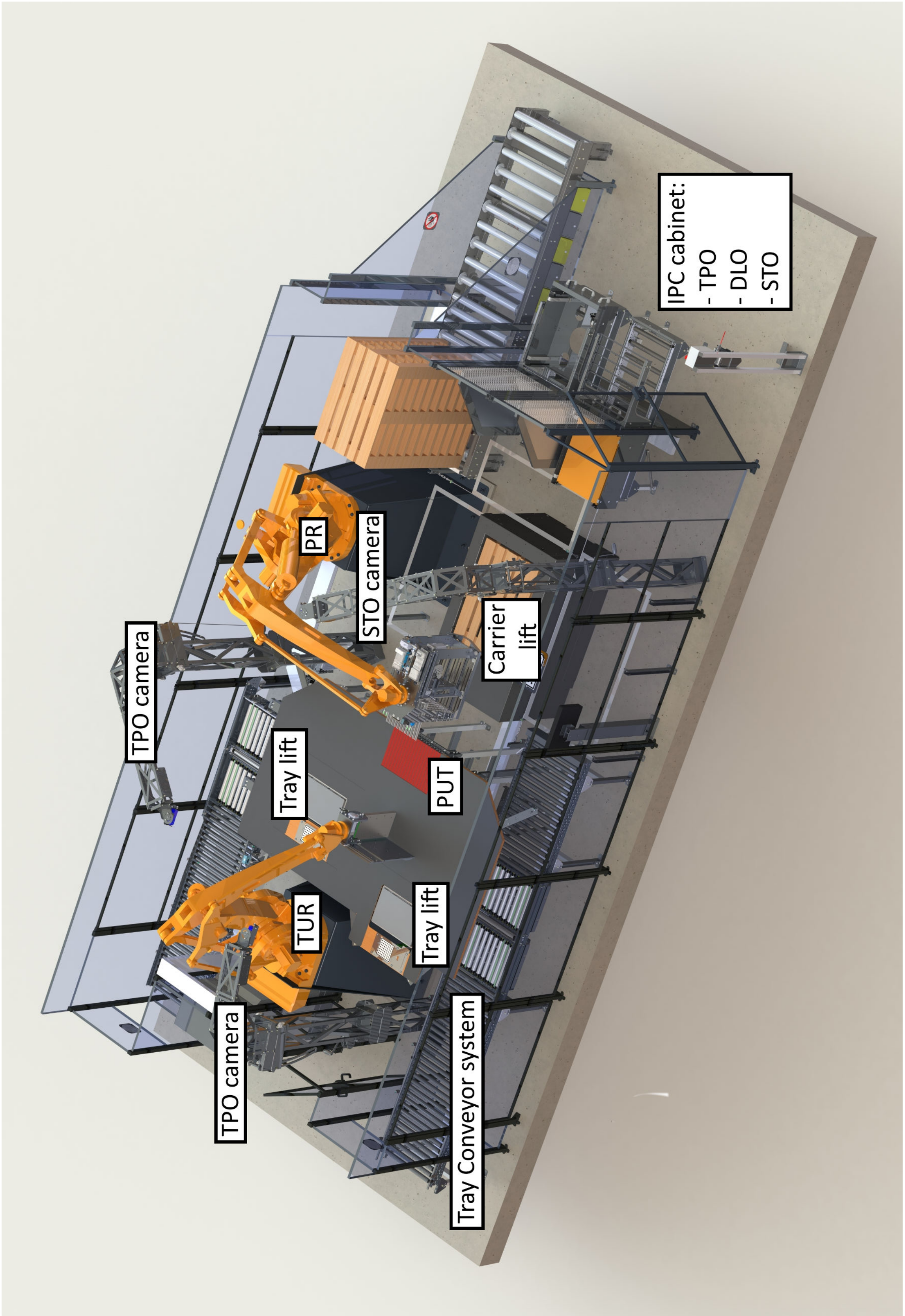


Figure 2-1: The different components of the ACP palletiser [6]

The trays that contain cases come into the system and are moved forward by a conveyor system. The conveyor system directs the trays in an alternating way to two different tray lifts. The alternating behaviour is a design decision and not a restriction of some sort. While these trays are lifted, their bottom side is lifted as well such that it has the same height as the sliding table. The two parallel conveyor systems are positioned under the sliding table.

When a tray is lifted to the height of the sliding table, it can be unloaded from its containing cases by the Tray Unloading Robot (TUR). When the needed cases are unloaded, the tray is lowered again and depending on its content redirected to a certain spot. For this project, the interest is in knowing what happens with the cases. The particular choice for which case is taken from the tray, is made when the tray is in the tray lift. This is based on data such as the location of the cases within the tray in combination with its dimensions and destination. When this is decided the TUR pushes the case away from the tray towards its destination on the PickUp Table (PUT).

When the case is positioned on the PUT, the Palletiser Robot (PR) picks it up and puts it on the carrier, e.g. a pallet or a roll cage. The position at which the case is placed is predetermined. The carrier is placed on a carrier lift and this lift is lowered between certain placements of cases. This makes it possible for the PR to place the cases on the stack within an optimised height range. In some stacks, slip sheets are placed between layers of cases to improve the stability of the stack. These sheets are placed by the PR as well. The moments on which the lift lowers and when slip sheets are placed are predetermined. The carrier lift is the outbound of the system and furthermore the boundary where the scope of this project stops.

Every case is handled in this way by the palletiser in the order of which the cases are delivered to the system. The system cannot choose for a certain order in which the cases are handled, that is out of its capabilities.

2-2 Components

The system can be divided into several components that are explained in more detail. These components are the following:

- Tray conveyor system & Tray lifts
- Tray Pattern detection Object (TPO) & Descrambling Logic Object (DLO)
- Tray Unloading Robot (TUR)
- PickUp Table (PUT)
- Palletiser Robot (PR)
- STack check Object (STO) & Carrier lift

The position in the system of the different components can be seen in Figure 2-1.

2-2-1 Tray Conveyor System & Tray Lifts

The tray conveyor system is the part where the infeed of the trays containing cases takes place. In an alternating way, the trays are directed to one of the two lanes that lead to the tray lifts. Once this is done, first only the tray insert is lifted, which is the floor of the tray. When this is done a top view photo is made from the tray which is used by TPO, this is explained in Section 2-2-2. When this photo is made the tray is lifted further until the tray insert is at the same height as the sliding table. After all cases that needed to be taken from the tray are pushed off, the tray is lowered again. This lowering starts as soon as the TUR pushed the last case just from the tray. Once the tray is at low position and moved away from the lift, the next tray in this lane moves into the lift and the same steps are done again. These proceedings are done by both lifts and related lanes in parallel.

2-2-2 Tray Pattern detection Object & Descrambling Logic Object

TPO and DLO are responsible for determining the positions of the cases in the trays and consequently making a plan to move a case from the tray to the PUT. First a top view photo is made of the tray, this photo gives the coordinates of the cases in three dimensions. As has been said, the photo is taken at the moment that the tray has lifted the tray insert. However, when multiple cases are taken from a single tray, a photo is taken every time between two cases taken away from the tray. This photo is taken at the moment that the TUR is out of the camera view, consequently TPO makes its calculations again.

With the information provided by TPO, an optimal solution is made for the TUR on how it should push a certain case from the tray towards the PUT. This path choice is determined and optimised by DLO. DLO also decides which case should be taken from the tray in situations when multiple cases are located in a tray.

2-2-3 Tray Unloading Robot

The TUR is responsible for moving cases from the tray towards the PUT. DLO gives orders to the TUR on what it should do. This order includes the destination of the case on the PUT including the orientation. The TUR has two different locations from which it starts pushing cases as there are two tray lifts. There is no choice for the TUR in choosing the tray from which it takes a case, this is predetermined. This is determined in the way that the trays are handled in an alternating manner. However, one tray should be finished first before the next tray can be handled. This means that if, from a particular tray three cases should be taken, the TUR moves to this tray three successive times. Between the tray and the PUT, the TUR pushes the cases over a sliding table. This sliding table has the same height as the tray in upper position and the PUT. The TUR can only place a case on the PUT when the PUT is free, i.e. the previous case has been taken away from the PUT.

2-2-4 PickUp Table

The PUT is the position where the TUR places cases and where the PR picks them up. This makes the PUT a link between the two robots where all cases pass. A note should be made about the PUT floor which lowers and rises every case cycle. The lower position of the PUT floor makes it possible for the PR to put its gripper into the PUT such that it can pick up the case. The high position of the PUT floor ensures friction that is needed at the moment that the TUR pushes the case on the PUT. After the case is placed, the floor is lowered such that the PR can come into the PUT. And after the case is picked up, the floor is lifted such that the TUR can place the next case on the PUT. The PUT has two points to which the placed cases are aligned. Additionally the cases are placed on the PUT in two different orientations, length-wise and width-wise. This results in four different ways on how cases are placed on the PUT.

2-2-5 Palletiser Robot

The PR is the robot that is responsible for moving cases from the PUT towards the destination on the carrier. As soon as a case is placed on the PUT and the TUR has moved far enough away from the PUT, the PR can go into the PUT to pick up the case. The position at which the case is placed on the PUT and the position on the stack where the case has to be placed is predetermined and thereby also the path of the PR is predetermined. The PR needs to get a signal provided by STO that the stack is approved and that the next case may be placed, STO is explained in Section 2-2-6. The PR does not only place the cases on the stack. It is also responsible for placing the pallets on the pallet lift and placing slip sheets between stack layers. The system that is described in this thesis has roll cages instead of pallets, therefore the task of placing pallets by the PR is not taken into account.

2-2-6 STack check Object & Carrier Lift

STO is responsible for monitoring the building of the stack of cases on the load carrier. This works in a similar way as TPO does. Above the stack, there is a camera that makes a top view photo of the stack. With the information provided by this photo, the stack is checked on whether the path that the PR will make is free of obstacles. This approval is needed for the PR before it is allowed to act. The photo is made after the carrier lift is lowered. When no obstacles are in the way of the PR's path, an approval is sent to the PR. The PR needs approval because it has a predetermined path in which it moves.

The carrier lift lowers the carrier of the stack while the stack is being build by the PR. The moments at which the carrier is lowered by the lift is predetermined in order to make the movement of the PR optimal. The carrier lift contains of two lifts, the pin lift and the fork lift. First only the pin lift has the carrier on it, halfway down the forklift takes over the carrier. This is done to make it possible to place a new carrier on the pin lift immediately after the previous stack is finished. At this moment the fork lift is still handling the previous carrier while the next stack is already being built on the pin lift.

Chapter 3

Max-Plus Linear Systems

The model that is made of the automatic palletiser system has the purpose to draw conclusions on capacity, bottleneck and sensitivity issues. Information on paths that the cases follow or the accuracy of the placing of the cases are not within the interest of this project are therefore not taken into account in the model. An example of something within the scope of interest is knowing how long it takes to place 1000 cases on carriers by one automatic palletiser system. This scope makes certain *events* that happen at certain moments the interesting part and are the modelling variables therefore. Due to this, the automatic palletiser is described as a Discrete Event Dynamic System (DEDS). The event counter is related to the cases that go through the system. DEDS's are described by [4] as follows: *"This class essentially contains man-made systems that consist of a finite number of* ***resources*** *shared by several* ***users*** *which all contribute to* ***the achievement of some common goal****."* In the case of the palletiser the resources are the robots, the users are the cases and the achievement of the common goal is being placed correctly on the carrier.

The palletiser is treated as a DEDS and the time behaviour of internal events is modelled. A DEDS qualifies for a Max-Plus Linear (MPL) system if it contains synchronisation but no concurrency. Synchronisations require the availability of several resources or users at the same time. An example of this is that a case can only be picked up from the PickUp Table (PUT) if:

- The case is at the PUT,
- the PUT floor is lowered,
- the Tray Unloading Robot (TUR) has left the PUT area
- and the Palletiser Robot (PR) is available.

Concurrency appears when some user can choose among several resources. This does not appear in the palletiser system, the cases (users) just follow a path along the robots (resources) which is predetermined.

The existence of synchronisation and absence of concurrency qualifies the system for being linear in the max-plus algebra which is the motivation for modelling the system as an MPL model. The basis of max-plus algebra is explained in Section 3-1. An MPL model uses has max and plus operations from conventional algebra which are described respectively as addition and multiplication in the max-plus algebra. The linearity is described as in conventional algebra but then with the operations from max-plus algebra. The state $x(k)$ of an MPL system consists of the time instants at which the internal events of the system happen for the k'th time. MPL systems and the theory behind it is explained in more detail in Section 3-2.

Parameters in the model almost all represent time spans of processes that happen between different events. These processes do not always take equal time in reality. One can choose to use average values for this in the model and make it deterministic. However to make the model more realistic, several timing variables are made stochastic. MPL models with stochastic parameters have certain characteristics explained in Section 3-3.

It is stated in [5] that adding switching to MPL systems can solve modelling issues that arise due to characteristics changes within an MPL system. Switching is used for changing the structure of the system which is needed at certain moments when for example events are added, removed or when the order of events changes. The form of the state stays the same, but the determination of the state evolution changes. The mode of the Automated Case Picking (ACP) automatic palletiser system also needs switching, the theory behind switching is explained in more detail in Section 3-4.

3-1 Max-Plus Algebra

A DEDS can be linear in the max-plus algebra which makes it an MPL systems. Max-plus algebra is different from conventional algebra as the only two defined operations are addition and taking the maximum. This section shows the basics of max-plus algebra and some extensions. Ideas and theories in this section are based on [7].

3-1-1 Max-Plus Algebra Basics

Max-plus algebra is defined for real numbers united with $-\infty$. Furthermore, $-\infty$ and 0 are defined by other symbols as shown in (3-1) and (3-2).

$$\varepsilon \stackrel{\text{def}}{=} -\infty \tag{3-1}$$

$$e \stackrel{\text{def}}{=} 0 \tag{3-2}$$

The set of numbers that is used within max-plus algebra is summarized into the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon\}$. Max-plus algebra relies on two operations illustrated by $\oplus$ and $\otimes$, respectively called max-plus addition and max-plus multiplication. These operations are defined for $a, b \in \mathbb{R}_{\max}$ is in (3-3) and (3-4).

$$a \oplus b \stackrel{\text{def}}{=} \max(a, b) \tag{3-3}$$

$$a \otimes b \stackrel{\text{def}}{=} a + b \tag{3-4}$$

It follows that ε behaves as 0 in conventional algebra due to the characteristics shown in (3-5) for all $a \in \mathbb{R}_{\max}$.

$$\varepsilon \oplus a = \max(\varepsilon, a) = a \quad \text{and} \quad \varepsilon \otimes a = \varepsilon + a = \varepsilon \tag{3-5}$$

Likewise, for max-plus multiplication e behaves as 1 in conventional algebra for all $a \in \mathbb{R}_{\max}$ as shown in (3-6).

$$e \otimes a = e + a = a \tag{3-6}$$

Powers can also be used in max-plus algebra which results in Equation (3-7) for $a \in \mathbb{R}_{\max}$ and $k \in \mathbb{N}$.

$$a^{\otimes k} = \underbrace{a \otimes a \otimes \ldots \otimes a}_{k \text{ times}} = \underbrace{a + a + \ldots + a}_{k \text{ times}} = k \times a \tag{3-7}$$

Taking the power to 0 is defined as $a^{\otimes 0} \stackrel{\text{def}}{=} e(=0)$.

3-1-2 Max-Plus for Matrices

The rules introduced for max-plus algebra can also be used for vectors and matrices. Within a matrix A the entry at row i and column j is denoted as a_{ij}. For matrices A, B with $A \in \mathbb{R}_{\max}^{n \times m}$ and $B \in \mathbb{R}_{\max}^{m \times k}$, matrix multiplication is defined as in (3-8).

$$[A \otimes B]_{ij} \stackrel{\text{def}}{=} \bigoplus_{l=1}^{m} a_{il} \otimes b_{lj} = \max_{l \in \{1,\ldots,m\}} \{a_{il} + b_{lj}\}, \quad i \in \{1, \ldots, n\}, j \in \{1, \ldots, k\} \tag{3-8}$$

We see that like in conventional algebra, the number of columns of matrix A should be equal to the number of rows of matrix B. The max-plus multiplication is illustrated in Equation (3-9) with an example where $A = \begin{bmatrix} 5 & 7 \\ 0 & 3 \end{bmatrix}$ and $B = \begin{bmatrix} 7 & 1 \\ 1 & 0 \end{bmatrix}$.

$$\begin{aligned} A \otimes B &= \begin{bmatrix} (5 \otimes 7) \oplus (7 \otimes 1) & (5 \otimes 1) \oplus (7 \otimes 0) \\ (0 \otimes 7) \oplus (3 \otimes 1) & (0 \otimes 1) \oplus (3 \otimes 0) \end{bmatrix} \\ &= \begin{bmatrix} \max(5+7, 7+1) & \max(5+1, 7+0) \\ \max(0+7, 3+1) & \max(0+1, 3+0) \end{bmatrix} \\ &= \begin{bmatrix} 12 & 7 \\ 7 & 3 \end{bmatrix} \end{aligned} \tag{3-9}$$

For matrices with appropriate size, i.e. $A, B \in \mathbb{R}_{\max}^{n \times m}$, also matrix addition in max-plus algebra is possible. This operation is defined as in (3-10).

$$[A \oplus B]_{ij} \stackrel{\text{def}}{=} a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}), \quad i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\} \tag{3-10}$$

Here it is seen that the restriction on size of A and B is similar to what is the case with matrix addition in conventional algebra, they should be of same size. The max-plus addition is illustrated in Equation (3-11) with an example, for the same A and B as in Equation (3-9).

$$A \oplus B = \begin{bmatrix} 5 \oplus 7 & 7 \oplus 1 \\ 0 \oplus 1 & 3 \oplus 0 \end{bmatrix} = \begin{bmatrix} 7 & 7 \\ 1 & 3 \end{bmatrix} \tag{3-11}$$

The null and identity matrix are also defined in max-plus algebra. The null matrix is denoted by $\mathcal{E}$, all entries of this matrix are equal to ε. Notice that the entries are not equal to 0, but to $-\infty$. As a consequence, for matrix $A \in \mathbb{R}_{\max}^{n\times m}$, addition and multiplication with the null matrix results in Equations (3-12)-(3-15).

$$A \oplus \mathcal{E}^{n\times m} = A \tag{3-12}$$
$$\mathcal{E}^{n\times m} \oplus A = A \tag{3-13}$$
$$A \otimes \mathcal{E}^{m\times k} = \mathcal{E}^{n\times k} \tag{3-14}$$
$$\mathcal{E}^{k\times n} \otimes A = \mathcal{E}^{k\times m} \tag{3-15}$$

The identity matrix in max-plus algebra E is defined as in (3-16).

$$[E]_{ij} = \begin{cases} e & \text{for } i = j \\ \varepsilon & \text{otherwise} \end{cases} \tag{3-16}$$

For multiplication, the identity matrix also has the characteristics as they are known from conventional matrix operations which is shown in Equations (3-17) and (3-18).

$$A \otimes E^{m\times m} = A \tag{3-17}$$
$$E^{n\times n} \otimes A = A \tag{3-18}$$

For square matrices, the k'th multiplication power can be calculated of that matrix as shown in Equation (3-19). This operation works similarly as taking the power of a scalar which was shown in Equation (3-7).

$$A^{\otimes k} \overset{\text{def}}{=} \underbrace{A \otimes A \otimes \ldots \otimes A}_{k \text{ times}} \tag{3-19}$$

The power multiplication is defined for $k \in \mathbb{N}_{\geq 1}$. For $k = 0$ the power is defined as $A^{\otimes 0} \overset{\text{def}}{=} E$.

3-1-3 Communication Graphs

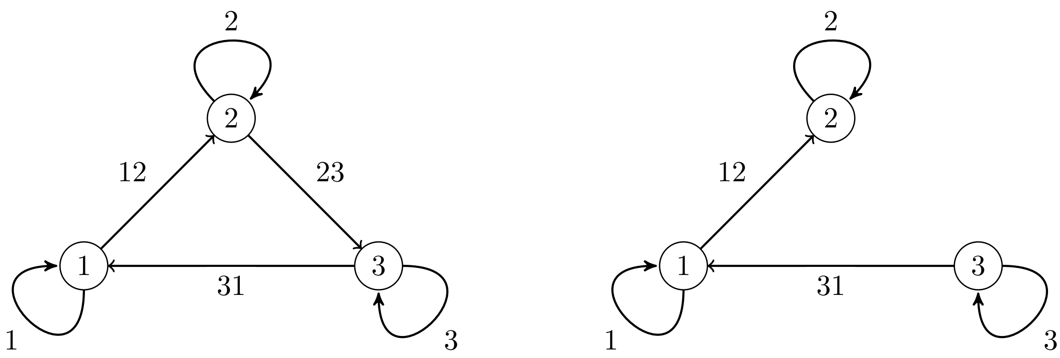
A graph contains nodes and arcs and every arc should point from one node to another to make the graph a directed graph. A graph, set of nodes and set of arcs are respectively denoted by $\mathcal{G}$, $\mathcal{N}$ and $\mathcal{D}$. When matrix A is square, it can be associated to a directed graph. This associated graph is called the communication graph of matrix A and is denoted by $\mathcal{G}(A)$.

When an arc $(j, i) \in \mathcal{D}$ in the communication graph $\mathcal{G}(A)$ point from node $j \in \mathcal{N}$ to node $i \in \mathcal{N}$ with weight w, entry a_{ij} of matrix A is equal to w. If there is no arc pointing from node $l \in \mathcal{N}$ to node $k \in \mathcal{N}$, entry a_{kl} is equal to ε. For matrix A given in (3-20) the communication graph is shown in Figure 3-1a.

$$A = \begin{bmatrix} 1 & \varepsilon & 31 \\ 12 & 2 & \varepsilon \\ \varepsilon & 23 & 3 \end{bmatrix} \tag{3-20}$$

A path is a combination of arcs that connects two nodes. A graph is called strongly connected when for any pair of nodes $i, j \in \mathcal{N}$, there exists a path from i to j. The graph in Figure 3-1a is a strongly connected graph. The matrix associated with a strongly connected graph is called irreducible, so A from Equation (3-20) is irreducible.

When the arc pointing from node 2 to node 3 is taken away, the resulting graph in Figure 3-1b is not strongly connected any more. This due to the fact that there is no path from nodes 1 and 2 to node 3. The matrix associated to a graph that is not strongly connected is called reducible.



(a) Communication graph of matrix A, denoted by $\mathcal{G}(A)$.

(b) $\mathcal{G}(A)$ with the arc from node 2 to node 3 removed.

Figure 3-1: Example graphs

3-1-4 Calculations with Communication Graphs

In Section 3-1-5 a direct link between square matrices and directed graphs is given. Calculations with these matrices give information on the corresponding communication graphs. By looking at $\mathcal{G}(A)$ in Figure 3-1a, it is possible to determine the maximal possible weight of a path of length two from node 1 to node 2. There are two possible paths of length two from node 1 to node 2. The first is taking arc $(1,1)$ and then arc $(1,2)$, the second is taking arc $(1,2)$ and then arc $(2,2)$. Their total weights are respectively equal to 13 and 14, so the latter path has the maximal weight.

This conclusion could also be drawn by doing calculations with the associated matrix A from (3-20). By calculating $A^{\otimes 2}$, the maximal weight possible of a path of length two from node 1 tot node 2 is given by $\left[A^{\otimes 2}\right]_{21}$. To illustrate this the calculation is shown in Equation (3-21).

$$A^{\otimes 2} = \begin{bmatrix} 1 & \varepsilon & 31 \\ 12 & 2 & \varepsilon \\ \varepsilon & 23 & 3 \end{bmatrix}^{\otimes 2} = \begin{bmatrix} 2 & 54 & 34 \\ 14 & 4 & 43 \\ 35 & 26 & 6 \end{bmatrix} \tag{3-21}$$

It can be observed that $\left[A^{\otimes 2}\right]_{21}$ is equal to 14. The advantage of calculating powers of the matrix associated to a graph is that all maximum values of paths of a certain length from one node to another can be read.

3-1-5 Circuits in Communication Graphs

A circuit within a directed graph is a path that starts at a node and ends at the same node. A circuit is called an elementary circuit if every node in the circuit is only present once. This results in four elementary circuits for graph $\mathcal{G}(A)$ in Figure 3-1a. Three elementary circuits contain one arc, these are the circuits $\{(1,1)\}$, $\{(2,2)\}$ and $\{(3,3)\}$. Circuits only containing one arc are called self-loops. The last elementary circuit goes around all nodes and is equal to $\{(1,2),(2,3),(3,1)\}$.

A circuit γ has a certain circuit weight denoted by $|\gamma|_{\mathrm{w}}$. This circuit weight is equal to the sum of the weights of all the arcs within that circuit. The length of a circuit γ is defined as the number of arcs within a circuit and denoted by $|\gamma|_{\mathrm{l}}$. With the weight and length of a circuit defined, the average circuit weight is defined by the quotient of the two $\frac{|\gamma|_{\mathrm{w}}}{|\gamma|_{\mathrm{l}}}$. The set of all circuits of the communication graph of matrix A is denoted by $\mathcal{C}(A)$.

Circuits within a graph with maximum average circuit weight are called critical circuits. $\mathcal{G}(A)$ has one critical circuit given by $\{(1,2),(2,3),(3,1)\}$ which has average circuit weight $\frac{12+23+31}{3} = 22$. The critical graph $\mathcal{G}^c(A) = (\mathcal{N}^c(A), \mathcal{D}^c(A))$ is the graph consisting of all the nodes and arcs that are part of a critical circuit.

3-1-6 Cyclicity

Cyclicity is a characteristic of a graph $\mathcal{G}$, denoted by $\sigma_{\mathcal{G}}$ which is defined differently for strongly connected graphs and not strongly connected graphs. The definitions given in this section are adopted directly from [7]:

- If $\mathcal{G}$ is strongly connected, then its cyclicity equals the greatest common divisor of the lengths of all elementary circuits in $\mathcal{G}$. If $\mathcal{G}$ consists of just one node without a self-loop, then its cyclicity is defined to be one.
- If $\mathcal{G}$ is not strongly connected, then its cyclicity equals the least common multiple of the cyclicities of all maximal strongly connected subgraphs.

Graph $\mathcal{G}(A)$ in Figure 3-1a is strongly connected with elementary circuits having length 1 or 3. The greatest common divisor of the lengths of all elementary circuits and thus also the cyclicity of $\mathcal{G}(A)$ is equal to $\sigma_{\mathcal{G}(A)} = 1$.

When matrix A is associated to a communication graph that has at least one circuit, the cyclicity of matrix A is also defined. This cyclicity is different from the cyclicity of the communication graph of A. The cyclicity of matrix A, having at least one circuit, is denoted by $\sigma(A)$, and is defined to be equal to the cyclicity of the critical graph of A. The critical graph is explained in Section 3-1-5. For the graph shown in Figure 3-1a, the cyclicity of the critical graph is equal to 3. As a conclusion the cyclicity of matrix A is equal to $\sigma(A) = 3$.

3-2 Max-Plus Linearity

The explained max-plus algebra is interesting in situations where systems are described with addition and maximum-operators. Some systems that are explained with these operators are linear in the max-plus algebra. When that is the case, the system can be written in a state-space form with max-plus operators instead of the conventional state-space form with normal matrix-vector multiplication. The linear equations in max-plus algebra have interesting characteristics, some of them are explained in Sections 3-2-2 and 3-2-3. Section 3-2-4 explains the use of linear max-plus expressions for MPL systems.

3-2-1 Linear Equation

Linear equations within max-plus algebra take the form shown in Equation (3-22) with $A \in \mathbb{R}_{\max}^{n\times n}$ and $x, b \in \mathbb{R}_{\max}^{n}$.

$$x = A \otimes x \oplus b \tag{3-22}$$

For a given A and b such linear equations can be solved under some assumptions. First the Kleene star operator $(.)^*$ is defined for matrix $A \in \mathbb{R}_{\max}^{n\times n}$ as in (3-23).

$$A^* \stackrel{\text{def}}{=} \bigoplus_{k=0}^{\infty} A^{\otimes k} = E \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus \dots \tag{3-23}$$

This A^* exists if all circuits in the communication graph of A have non-positive weight. By existing it is meant that none of the entries of A^* is equal to ∞. Assuming that the communication graph of $A \in \mathbb{R}_{\max}^{n\times n}$ only has non-positive circuit weights, the calculation given in (3-23) can be restricted to the calculation given in (3-24).

$$A^* = \bigoplus_{k=0}^{n-1} A^{\otimes k} \tag{3-24}$$

It follows that the solution of $x = A \otimes x \oplus b$ is equal to Equation (3-25).

$$x = A^* \otimes b \tag{3-25}$$

Statements made in this section are taken from [7] where also the proofs can be found.

3-2-2 Cycle Time Vector

For a given sequence $\{x(k) : k \in \mathbb{N}\}$ with $x(k) \in \mathbb{R}_{\max}^{n}$, the cycle time vector is defined. This cycle time vector is denoted by $\eta = (\eta_1, \eta_2, \dots, \eta_n)^\top$ where all η_j are defined as in (3-26).

$$\eta_j = \lim_{k\to\infty} \frac{x_j(k)}{k}, \quad \text{for } j \in \{1, \dots, n\} \tag{3-26}$$

When all η_j's have the same value, this value is called the asymptotic growth rate of the sequence. The mentioned sequence is related to the recurrence relation $x(k) = A \otimes x(k-1)$. Given a system matrix A and an initial state $x_0 = x(0)$, the resulting sequence is determined.

Take a recurrence relation of the form $x(k) = A \otimes x(k-1)$ with $A \in \mathbb{R}_{\max}^{n\times n}$ and A being regular (regular means that every row of A contains an element not equal to ε). Also assume that there exists an initial condition $x(0) = x_0 \in \mathbb{R}^n$ such that the limit in (3-27) exists.

$$\lim_{k\to\infty} \frac{A^{\otimes k} \otimes x_0}{k} \tag{3-27}$$

Then it is proven [7] that this limit, in fact the cycle time vector of A, is equal for any initial condition $x(0) \in \mathbb{R}^n$. Note that all elements of the initial condition $x(0)$ should be finite. Also note that the numerator of the fraction in (3-27) is equal to $x(k)$ with x_0 as initial value. In order to make this notation easier the k'th state with a given initial value x_0 is defined as in (3-28).

$$x(k; x_0) \stackrel{\text{def}}{=} A^{\otimes k} \otimes x_0 \tag{3-28}$$

3-2-3 Eigenvalues and Eigenvectors

Square matrices within max-plus algebra can have eigenvalues and eigenvectors. Let $A \in \mathbb{R}_{\max}^{n \times n}$, $\lambda \in \mathbb{R}_{\max}$ and $v \in \mathbb{R}_{\max}^{n}$ such that they satisfy Equation (3-29). Then λ is called an eigenvalue of A and v an eigenvector of A.

$$A \otimes v = \lambda \otimes v \tag{3-29}$$

Eigenvectors are always associated to a certain eigenvalue, therefore v is often called an eigenvector of A associated with eigenvalue λ.

The normalised matrix of A, denoted by A_λ, is defined for every entry as in Equation (3-30).

$$[A_\lambda]_{ij} = a_{ij} - \lambda \tag{3-30}$$

Let $A \in \mathbb{R}_{\max}^{n \times n}$ be a matrix with communication graph $\mathcal{G}(A)$ and set of circuits $\mathcal{C}(A)$. With the concepts explained in previous sections it is possible to find eigenvalues and eigenvectors of A. Define λ as the average weight of a critical circuit of the communication graph of A. Then if λ is finite, it holds that λ is an eigenvalue of A. Linked to this it is possible to come up with an eigenvector of A associated to λ. Certain columns of the Kleene star of the normalised matrix A_λ are an eigenvector of A associated to λ. In fact it is a column corresponding to an arbitrary node η of the critical graph $\mathcal{G}^{\mathrm{c}}(A)$.

So for matrix A an eigenvalue λ and eigenvector v can be determined as in Equations (3-31) and (3-32). In this notation the subscript $\cdot\eta$ denotes the η's column of that matrix.

$$\lambda = \max_{p \in \mathcal{C}(A)} \frac{|p|_{\mathrm{w}}}{|p|_1} \tag{3-31}$$

$$v = [A_\lambda^*]_{\cdot\eta}, \quad \text{with } \eta \in \mathcal{N}^{\mathrm{c}}(A) \tag{3-32}$$

3-2-4 Max-Plus Linear Systems

In the introduction of this chapter, DEDSs are described and conditions are given that must hold for such a system to be linear in the max-plus algebra. Systems that satisfy these conditions are MPL and can be described by models of the form given by Equations (3-33) and (3-34).

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \tag{3-33}$$

$$y(k) = C \otimes x(k) \tag{3-34}$$

Here $x(k) \in \mathbb{R}_{\max}^{n}$ contains the time instants at which the internal events of the system occur for the k'th time. Similarly $u(k) \in \mathbb{R}_{\max}^{m}$ contains the time instants at which the input events occur for the k'th time. Finally $y(k) \in \mathbb{R}_{\max}^{p}$ contains the time instants at which the output events occur for the k'th time. So the system has m input events, n internal events and p output events. Usually in models describing systems from industry, the system matrices $A \in \mathbb{R}_{\max}^{n \times n}$ and $B \in \mathbb{R}_{\max}^{n \times m}$ contain parameters that represent process times, transportation times, etc. [5]. Systems that do not have an external input can also be described as an MPL system. The term with the B-matrix of Equation (3-33) is left out in those situations.

Furthermore usually the interesting part of the system is the state and thus it is often the case that the output is equal to the state. In the case that there is no input and the output is equal to the state the MPL model is of the form given by Equations (3-35) and (3-36).

$$x(k) = A \otimes x(k-1) \tag{3-35}$$

$$y(k) = x(k) \tag{3-36}$$

Equation (3-33) describes an explicit model as everything on the right hand side of the equation is known at the time that $x(k)$ is calculated. It is often the case that the model is implicit which means that $x(k)$ is present on the left hand side as well as on the right hand side of the equation. With the assumption that the output is equal to the state (and thus is left out), the model then takes the form given in Equation (3-37).

$$x(k) = A_0 \otimes x(k) \oplus A_1 \otimes x(k-1) \oplus B \otimes u(k) \tag{3-37}$$

In Section 3-2-1 it is explained how this model can be solved where the b-term is equal to $A_1 \otimes x(k-1) \oplus B \otimes u(k)$ in this case. Consequently the solution for $x(k)$ is given as in (3-38).

$$x(k) = A_0^* \otimes (A_1 \otimes x(k-1) \oplus B \otimes u(k)), \quad A_0^* = \bigoplus_{j=0}^{n-1} A_0^{\otimes j} \tag{3-38}$$

This solution exists if A_0 satisfies the restrictions given in Section 3-2-1 as this ensures existence of A_0^*.

3-3 Stochastic Max-Plus Linear Systems

System parameters are usually not deterministic but somehow perturbed or variable. These uncertainties are preferably taken into account in a model that describes a system, however perturbations are unknown beforehand. Stochastic features are used in models to deal with these uncertainties. The main focus in this section is on models with stochastic parameters, the basics of this is explained in Section 3-3-1. Also some theory on the max-plus Lyapunov exponent is explained in Section 3-3-2. This max-plus Lyapunov exponent gives information on the cycle time vector of a stochastic MPL system that satisfies certain conditions.

3-3-1 Max-Plus Linear Models with Stochastic Parameters

MPL models with stochastic parameters are called Stochastic MPL models. This means that the entries of the A-matrices become stochastic variables. The recurrence relation then becomes as in Equation (3-39) in which it can be seen that matrix A is depending on k. This means that there is a sequence $\{A(k) : k \in \mathbb{N}\}$ with $A(k) \in \mathbb{R}_{\max}^{n \times n}$.

$$x(k) = A(k) \otimes x(k-1) \tag{3-39}$$

For a sequence of matrices, the order within a multiplication is important for the outcome. For this reason there is a definition and a notation for multiplication of the matrices of the sequence $\{A(k) : k \in \mathbb{N}\}$. This notation is defined in Equation (3-40) where it must hold that $m \geq l > 0$.

$$A[m,l] \stackrel{\text{def}}{=} \bigotimes_{k=l}^{m} A(k) \stackrel{\text{def}}{=} A(m) \otimes A(m-1) \otimes \cdots \otimes A(l+1) \otimes A(l) \tag{3-40}$$

With this definition and the knowledge of the initial value $x_0 = x(0)$, $x(k)$ can be expressed as $x(k, x_0) = A[k, 1] \otimes x_0$ where $k \geq 1$.

By making the weights of the arcs of the communication graph of $A(k)$ random, the statement can be made that the values of the A-matrices could sometimes be equal to $\varepsilon = -\infty$ and sometimes be equal to a real number. This could make the arc set of the communication graph of $A(k)$ different for two different values of k. This due to the reason that an entry $a_{ij} = \varepsilon$ results in no arc from node j to node i. Therefore there is the definition that $\{A(k) : k \in \mathbb{N}\}$ has fixed support if the set of arcs of the communication graph is non-random and does not depend on k. By means of a mathematical definition this is written as in Equation (3-41) for all $i, j \in \{1, 2, \ldots, n\}$.

$$\Big(\forall k \geq 0 : P\big([A(k)]_{ij} = \varepsilon\big) = 0\Big) \vee \Big(\forall k \geq 0 : P\big([A(k)]_{ij} = \varepsilon\big) = 1\Big) \tag{3-41}$$

A random matrix A is irreducible if it has fixed support and any sample of A is irreducible with probability one.

3-3-2 Max-Plus Lyapunov Exponent

Having an MPL model it is possible to make statements on the limit to infinity of the state. These statements have been adopted from [7].

Imagine a sequence of irreducible matrices $\{A(k) \in \mathbb{R}_{\max}^{n \times n} : k \in \mathbb{N}\}$ where all the matrices are independent and identically distributed. All finite elements of these matrices $A(k)$ are bounded from below by a finite number with probability one. Then the *max-plus Lyapunov exponent* λ is defined in Equation (3-42).

$$\lambda \stackrel{\text{def}}{=} \lim_{k\to\infty} \frac{1}{k} \left[A[k,0]\right]_{ij} = \lim_{k\to\infty} \frac{1}{k} \mathbb{E}\left[\left[A[k,0]\right]_{ij}\right] \tag{3-42}$$

Furthermore Equation (3-43) holds given that the initial condition x_0 is finite and integrable.

$$\lambda = \lim_{k\to\infty} \frac{x_j(k; x_0)}{k} = \lim_{k\to\infty} \frac{1}{k} \mathbb{E}\left[x_j(k; x_0)\right] \tag{3-43}$$

When the mentioned sequence of A-matrices does not have irreducible but reducible matrices, a statement can be made on the limit to infinity as well. First some aspects that are needed have to be explained. Denote the communication graph of a matrix $A(k)$ of the given sequence by $\mathcal{G} = \mathcal{G}(A(k))$ with node set $\mathcal{N}$ and arc set $\mathcal{D}$. The set $\pi^*(i)$ denotes the union of node $i \in \mathcal{N}$ and the set of all predecessors of node i. A predecessor $j \in \mathcal{N}$ of node i is a node such that $(j, i) \in \mathcal{D}$. The set of all predecessors of node i is denoted by $\pi^+(i)$. The mathematical notation for $\pi^*(i)$ is defined in (3-44).

$$\pi^*(i) \stackrel{\text{def}}{=} \{i\} \cup \pi^+(i) \tag{3-44}$$

The set $[i]$ denotes the union of nodes $j \in \mathcal{N}$ that communicate with node i. This means that there exists a path in $\mathcal{G}$ from j to i as well as from i to j, denoted by $i\mathcal{C}j$. This includes node i itself. The mathematical notation for this set is defined in (3-45).

$$[i] \overset{\text{def}}{=} \{j \in \mathcal{N} : i\mathcal{C}j\} \tag{3-45}$$

When restricting matrix $A(k)$ to the nodes of the set $[i]$, an irreducible matrix is obtained. For this matrix the max-plus Lyapunov exponent is denoted by $\lambda_{[i]}$. Now a statement can be made for the independent and identically distributed sequence $\{A(k) \in \mathbb{R}_{\max}^{n\times n} : k \in \mathbb{N}\}$ of reducible matrices with fixed support such that with probability one all finite elements are bounded from below by a finite number. For any arbitrary finite integrable initial value x_0, Equation (3-46) holds with probability one [4].

$$\lim_{k\to\infty} \frac{x_j(k;x_0)}{k} = \lim_{k\to\infty} \frac{1}{k}\mathbb{E}\left[x_j(k;x_0)\right] = \lambda_j, \quad \lambda_j = \bigoplus_{i\in\pi^*(j)} \lambda_{[i]}, \quad j \in \{1,\ldots,n\} \tag{3-46}$$

3-4 Switching Stochastic Max-Plus Linear Systems

Synchronisations or event order can change within a system. By using an MPL system it is not possible to model these characteristics. In order to be able to do this a Switching Max-Plus Linear (SMPL) system is introduced by [5]. Here it is stated that switching between modes makes it possible to change the structure of the system. The phenomenon “switching” is explained in Section 3-4-1, then the combination of stochastic parameters and switching is explained in Section 3-4-2.

3-4-1 Switching Max-Plus Linear Systems

With n_L modes in a system, a certain mode at event counter k is denoted by $l(k) \in \{1,\ldots,n_L\}$. This changes the MPL state-space model of Equation (3-33) into Equation (3-47) where $A^{(l(k))} \in \mathbb{R}_{\max}^{n\times n}$ and $B^{(l(k))} \in \mathbb{R}_{\max}^{n\times m}$ are the systems matrices for the corresponding mode $l(k)$.

$$x(k) = A^{(l(k))} \otimes x(k-1) \oplus B^{(l(k))} \otimes u(k) \tag{3-47}$$

The switching from one mode to another is determined by a so called switching mechanism. This switching can be defined deterministic or stochastic, however in this section only the deterministic situation is discussed. Two ways of explaining this deterministic switching mechanism are discussed in this section, first the formal mathematical way as stated in [5].

A switching variable $z(k)$ is the variable that determines the mode $l(k)$. The value of $z(k)$, and thus a potential mode change, can depend on the previous state $x(k-1)$, the previous mode $l(k-1)$, the input variable $u(k)$ and a control variable $v(k)$. Depending on the characteristics of mode switching within an SMPL system, the value and dimension of $z(k)$ is determined, using the just mentioned variables and a function Φ. This result in Definition (3-48).

$$z(k) = \Phi\left(x(k-1), l(k-1), u(k), v(k)\right) \in \mathbb{R}_{\max}^{n_z} \tag{3-48}$$

The image of $z(k)$ is denoted by the set $\mathscr{Z} \subseteq \mathbb{R}^{n_z}_{\max}$. This image is partitioned into n_L non-overlapping subsets $\mathscr{Z}^{(i)} \subset \mathscr{Z}, i \in \{1, \ldots, n_L\}$ in such a way that (3-49) holds.

$$\begin{aligned} &\forall k : z(k) \in \mathscr{Z}^{(i)}, i \in \{1, \ldots, n_L\} \\ &\bigcup_{i=1}^{n_L} \mathscr{Z}^{(i)} = \mathscr{Z} \end{aligned} \tag{3-49}$$

Now for a known $z(k)$ the mode at this event counter is obtained as in (3-50).

$$z(k) \in \mathscr{Z}^{(i)} \quad \Rightarrow \quad l(k) = i \tag{3-50}$$

In [8] an alternative way of writing the switching behaviour is shown which makes it easier to integrate random switching. In fact this way of writing is already stochastic, but as restrictions are added it still describes a deterministic switching manner. Again the switching can depend on the previous state $x(k-1)$, the previous mode $l(k-1)$, the input variable $u(k)$ and a control variable $v(k)$. Now the probability of switching to mode $l(k)$, given the determining variables, is given by Equation (3-51).

$$P\left[L(k) = l(k) | l(k-1), x(k-1), u(k), v(k)\right] \tag{3-51}$$

The expression $L(k)$ is the stochastic variable of which $l(k)$ is its value. Since P is a probability (3-52) must hold.

$$\begin{aligned} 0 \leq &P\left[L(k) = l(k) | l(k-1), x(k-1), u(k), v(k)\right] \leq 1 \\ \sum_{l(k)=1}^{n_L} &P\left[L(k) = l(k) | l(k-1), x(k-1), u(k), v(k)\right] = 1 \end{aligned} \tag{3-52}$$

To use this way of writing the switching mechanism for deterministic switching, there is an extra assumption. This means that with the knowledge of $l(k-1) = i \in \{1, \ldots, n_L\}$ and for given $x(k-1)$, $u(k)$ and $v(k)$ (3-53) must hold as well for all $l(k) \in \{1, \ldots, n_L\}$.

$$P\left[L(k) = l(k) | l(k-1) = i, x(k-1), u(k), v(k)\right] \in \{0, 1\} \tag{3-53}$$

This means that with the determining variables known, there is no doubt on the next mode, thus it is deterministic.

3-4-2 Combination of Switching and Stochastic Parameters

The explained options in switching and stochastic parameters can be combined within a model. The result is a Switching Stochastic Max-Plus Linear (SSMPL) model, the resulting state space representation of an SSMPL model is shown in Equation (3-54). This is also the sort of model that is used in order to model the automatic palletiser system.

$$x(k) = A(k)^{(l(k))} \otimes x(k-1) \oplus B(k)^{(l(k))} \otimes u(k) \tag{3-54}$$

The combination of switching and stochastics can also be used in another way by making the switching stochastic, this is called random switching [8]. Although random switching is not used in this thesis, it can be of interest for similar problems. However, the simulations that are performed with the model established with this project are executed for a certain order of cases. For this reason it is predetermined which modes are to be used for the cases and thus is the switching also predetermined and not random.

An example case study that is interesting to be investigated by a random switching model is making statements on how long it would take to process 1000 cases. Consequently the distinction can be made on whether the order of 1000 cases comes from a small supermarket or a large supermarket. As this information changes the stacks obtained by Load Forming Logic (LFL), the tray infeed would also be influenced. As a result the way in which actions happen in the system changes and consequently the random switching changes.

Chapter 4

Modelling the Automatic Palletiser

Modelling Vanderlande's Automatic Palletiser as a Switching Stochastic Max-Plus Linear (SSMPL) brings up a lot of choices that have to be made. The objective of the use of the model determines the form and aspects that are taken into account. The resulting model is validated on a reference simulation performed at Vanderlande. The simulation performed by Vanderlande is also used to determine timing variables in the SSMPL model. Choices and assumptions are made in the determination on what is and is not used from Vanderlande's reference simulation. Not only by system analysis but also by gaining insights from experts who have worked with the system, a lot of information is obtained about the system in order to get clear knowledge of the system. This chapter explains what is taken into account in the resulting SSMPL model and states the choices that are made.

4-1 Establishing the Model

The automatic palletiser cell as explained in Chapter 2 is modelled using MATLAB as an SSMPL model. In this model, the state of the system gives the time instants at which 33 internal events happen for a certain case. The situation on how a case is handled by the system differs per case, this depends on several characteristics. These different situations are called modes. This means that for every case, a certain mode is applicable. The theory on switching between such modes is stated in Section 3-4.

4-1-1 Internal Events

As the occurrences of events are linked to certain cases, the state elements are depending on an event counter k that denotes the k'th case handled by the system. The system contains two tray lifts that work parallel. There are 7 internal events that are linked to a tray lift, those events are all defined twice as there are two lifts. As a result, every event count contains 7 events that are not of interest for that specific state. By switching between modes it is possible to model the parallel event occurrences as explained in Section 4-1-2.

The system could be modelled with less than 33 events. But less details could then be drawn from the resulting data. All the events contained by the state vector are listed in Table A-1 in Appendix A-1. In this section some of those events are explained. In those explanation the focus is on the relevance for the model. Descriptions on how the system operates in detail are given in Chapter 2.

TUR and Palletiser Robot Stops

Both the Tray Unloading Robot (TUR) and the Palletiser Robot (PR) can make stops during their movement towards the case, or while carrying the case. Those stops are respectively denoted by (a) and (b).

When the TUR moves from the PickUp Table (PUT) towards one of the tray lifts, it can potentially stop. This potential stop is called TUR stop(a). It does this whenever the TUR may not enter the area within the Tray Pattern detection Object (TPO) camera view. State elements x_{11} and x_{12} are linked to this TUR stop. Event $x_{11}(k)$ is equal to the time instant that the TUR is at the position where it *would* stop while moving towards the k'th case. This means that if the TUR does not stop, $x_{11}(k)$ still has a value. Event $x_{12}(k)$ is equal tot the time instant that the k'th case starts moving towards the tray lift again. If the TUR does not stop during the movement towards the k'th case, the value of $x_{12}(k)$ is equal to $x_{11}(k)$.

When the PR moves from the PUT to the carrier, it can make a potential stop as well. This stop is called PR stop (b). The stop is modelled like TUR stop (a), the corresponding state elements are x_{30} and x_{31}. The reason why the PR cannot always enter the area of the carrier lift is due to the fact that STack check Object (STO) has to send out a signal that the stack is approved.

Both the TUR and the PR operate in the space around the PUT. As they cannot be in that space at the same time, they sometimes have to stop moving towards the PUT. Also these stops are modelled as explained for TUR stop (a) and PR stop (b). The corresponding state elements for TUR stop (b) are x_{18} and x_{19}. The corresponding state elements for PR stop (a) are x_{23} and x_{24}.

Case Placement on the Stack

The moments when a case is placed on the stack are important for calculations on capacity and analysis. For this reason, this event is modelled and given a specific state element, x_{33}. As the event is tracked for every case, the durations between cases being placed on the stack are available for analysis.

4-1-2 Modes

In order to describe different situations for the same system, different modes are defined. The model switches between these modes such that the correct event synchronisations are modelled for the corresponding case. The modelled internal events of the automatic palletiser system can be divided into two parts. The first part, called part I, contains the events that happen before the case is at the PUT. The second part, called part II, contains the events that happen after the case is at the PUT. For both parts, different modes hold which can be combined in different combinations for different cases. As the state of the cases going through the system are being modelled instead of the state of the system itself, the modes also describe certain characteristics concerning those cases. Table 4-1 lists the modes of the SSMPL model.

The reason to distinguish these different modes is to take into account certain correlations or system characteristics. Two of those situations of part I are explained here:

- In general, the path made by Descrambling Logic Object (DLO) takes longer in a situation with more than one case in the tray. This is due to the restrictions for the TUR pushing the case away from the tray correctly in such a situation.
- The distinction is made on which lift is used for the tray handling the cases, these lifts are denoted by lift 7 and lift 25. This difference is made because the tray lifts work in parallel. When a case is handled in tray lift 7 the events in the state vector linked to tray lift 25 get the same value as in the previous state vector. In this manner, the similar but parallel events can be modelled in the SSMPL model. In order to model this correctly, modes are needed for the different lifts.

Table 4-1: Modes descriptions

Mode	Description
I.1	There is one case in the tray, this case is taken as first of that tray at lift 7.
I.2	There is one case in the tray, this case is not taken as first of that tray at lift 7.
I.3	There is one case in the tray, this case is taken as first of that tray at lift 25.
I.4	There is one case in the tray, this case is not taken as first of that tray at lift 25.
I.5	There are multiple cases in the tray, the taken case is the first case taken from that tray at lift 7.
I.6	There are multiple cases in the tray, the taken case is not the first case taken from that tray at lift 7.
I.7	There are multiple cases in the tray, the taken case is the first case taken from that tray at lift 25.
I.8	There are multiple cases in the tray, the taken case is not the first case taken from that tray at lift 25.
II.1	The handled case is the first placed case of a stack. Furthermore the lift takeover took place during the making of the previous stack.
II.2	The handled case is the first placed case of a stack. Furthermore the lift takeover took place after the making of the previous stack.
II.3	The handled case is placed directly after a slip sheet has been placed on the stack.

II.4	The handled case is placed directly after a lift takeover took place. Furthermore the case is not the first of a stack and not placed directly after a slip sheet has been placed on the stack.
II.5	The handled case is placed directly after the carrier lift has been lowered. Furthermore the other situations described in modes II.1 - II.4 did not occur.
II.6	The handled case is placed on the stack and none of the situations described in modes II.1 - II.5 did occur. The Carrier lift stayed at the same height.

4-1-3 Timing Variables

The modelled events happening in the automatic palletiser system are chosen conform Section 4-1-1. Between the occurrences of the modelled events, several actions happen within the system. As these actions take time, timing variables are modelled as links between the internal events. Every mode covers a certain situation of the system, and therefore every mode is modelled with its own "web" of events and timing variables. The choice of internal events and modes define the timing variables that have to be modelled. All the different timing variables are listed in Table A-2 in Appendix A-2. For distinct modes, it is known that actions take a different amount of time. For this reason there are timing variables that describe the same action but are defined apart as they are mode specific.

4-1-4 Synchronisations

The synchronisations of the model describe the dependencies within the system. The timing variables explained in Section 4-1-3 are also synchronisations. However, this section only explains the synchronisations that do not have a duration. Like in Figure 4-1, where the modelled subsequent events A and B are shown graphically. Because of the synchronisation without duration, event B immediately happens after event A happened and they happen at the same time.

Due to the structure differences of the system between the different modes, the synchronisations also differ. Table A-3 lists all the synchronisations of part I of the system schematically per mode. Some synchronisations link events of the same time step, others link two events of different time steps. All synchronisations of part I without duration that link events of different time steps occur when the event in that synchronisations does not occur in a certain mode. This is clarified by means of an example: In Table A-1 it is stated that the value

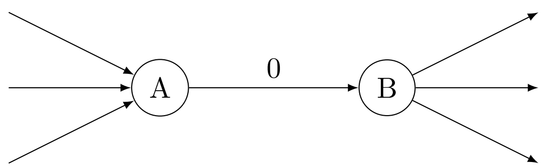


Figure 4-1: Two subsequent events synchronised inside a greater model

of event $x_1(k)$ is equal to the moment that the tray containing the k'th case arrives at the intermediate height where the TPO photo is taken. If mode I.2 with an arbitrary mode of part II holds for time step k, the equation that determines $x_1(k)$ is given by Equation (4-1). Mode I.2 is described in Table 4-1 as the situation when there is one case in the tray and this case is not taken as first from that tray at lift 7. As the case is not the first taken from that tray, the TPO photo is not made at the middle height of the tray lift but at the top height. This means that event x_1 never occurs in this situation and the value of this event is equal to the time instant that it did happen for the last time.

$$\begin{aligned} x_1(k) &= 0 \otimes x_1(k-1) \\ &= 0 + x_1(k-1) \\ &= x_1(k-1) \end{aligned} \tag{4-1}$$

Table A-4 lists the synchronisations of part II of the system schematically per mode. The synchronisations of events without durations are the same for every mode in part II. Unlike part I, part II contains event synchronisation between different time steps for non equal events. As an example event x_{19} is looked at into further detail. Table A-1 states that the value of event x_{19} is equal to the moment that the TUR leaves TUR stop (b) while it is pushing the case towards the PUT. For every mode the value of $x_{19}(k)$ is determined as in Equation (4-2). This means that the TUR pushing the k'th case leaves TUR stop (b) as soon as:

- The TUR arrived at TUR stop (b) with the k'th case,
- the PUT floor is at high position after the $k-1$'th case is taken from the PUT
- and the PR is far enough away from the PUT while carrying the $k-1$'th case.

$$\begin{aligned} x_{19}(k) &= \left(0 \otimes x_{18}(k)\right) \oplus \left(0 \otimes x_{29}(k-1)\right) \oplus \left(0 \otimes x_{32}(k-1)\right) \\ &= \max\left(x_{18}(k), x_{29}(k-1), x_{32}(k-1)\right) \end{aligned} \tag{4-2}$$

The result of putting together all the synchronisations and dependencies is a topological graph in which all the internal events are linked. The topological graph corresponding to mode I.1-II.1 is shown in Figure 4-2. The arrows in the graph represent the elements of the A_0 and A_1 matrices explained in Section 3-2-4. So this topological graph is not equal to the communication graph corresponding to the SSMPL model's A-matrix.

As a comparison, mode I.3-II.1 is also put in a topological graph and shown in Figure 4-3. This mode describes exactly the same situation as mode I.1-II.1 except the tray lift in which the case is handled. By comparing these graphs, the result of modelling similar and parallel events within the same model is visible.

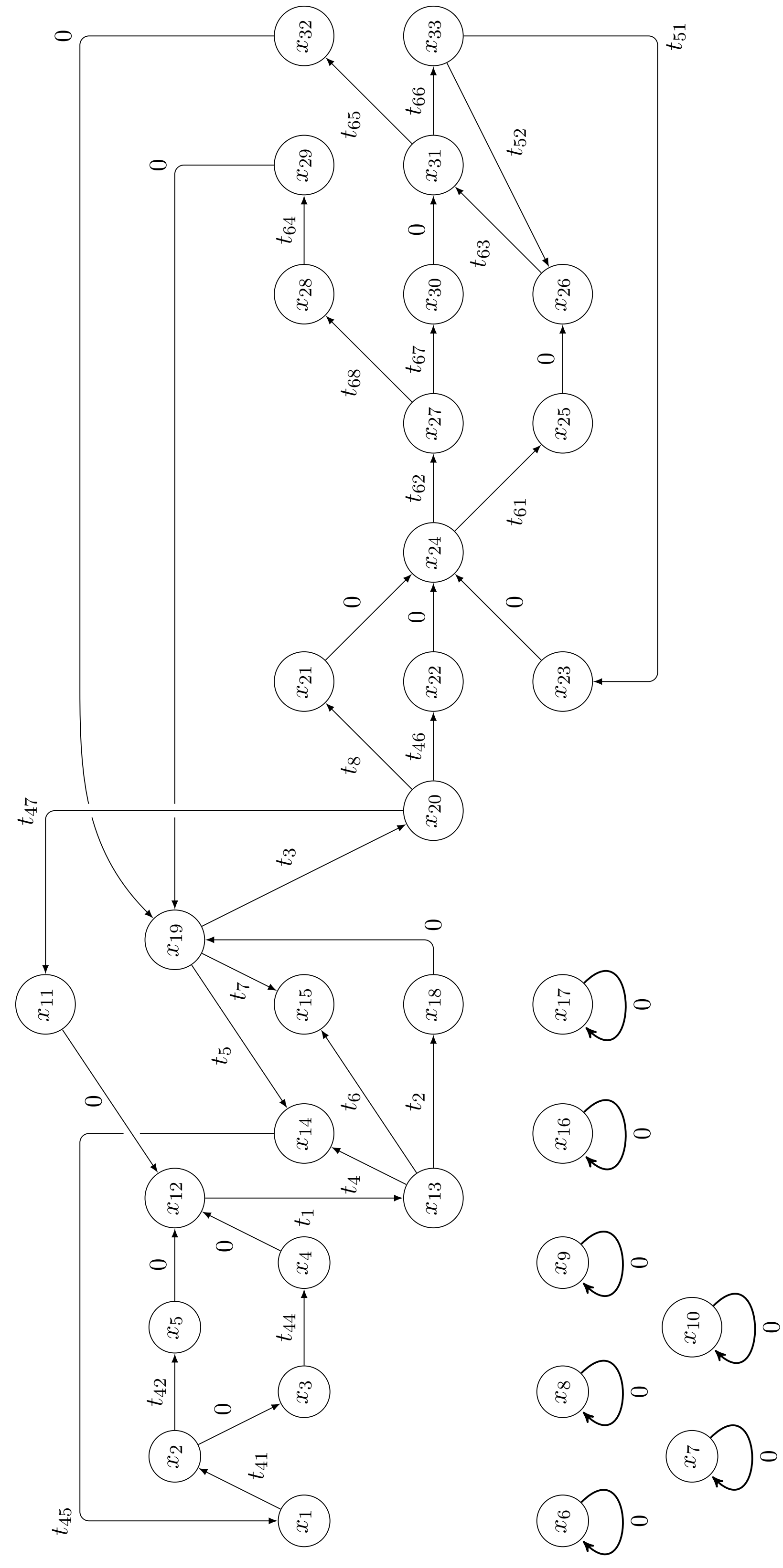


Figure 4-2: Topological graph representing the model in mode I.1-II.1

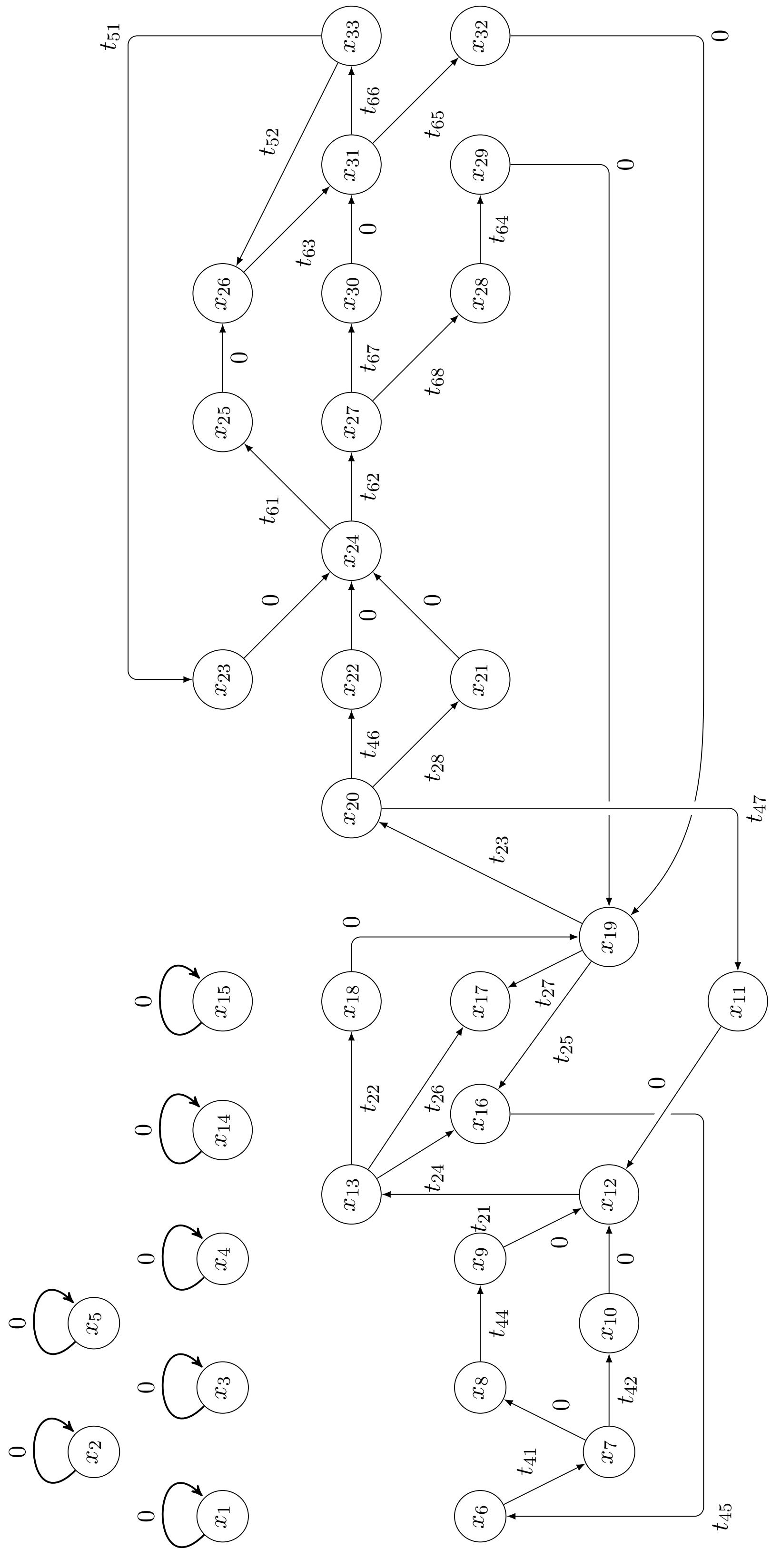


Figure 4-3: Topological graph representing the model in mode I.3-II.1

4-1-5 Assumptions

Several assumptions have been made during modelling the automatic palletiser. An explanation on making these assumptions is given in this section.

Infinite Tray Infeed

In reality the trays containing the cases are stored automatically in a storage warehouse. When a tray is needed, it is pulled out of the storage and brought to the correct palletiser. They tray then enters the system as infeed which is explained in Section 2-1. Therefore the trays are not always located at the automatic palletiser "on time", i.e. the palletiser sometimes has to wait for trays to continue its job. In order to make conclusions on capacity of the palletiser, the influences from outside the system are neglected. And thus the choice is made to model the system without the input of trays by modelling an infinite tray infeed which cannot run dry.

No Obstruction by Outfeed

The choice is made to not model possible obstructions below the carrier lift. This is done for the same reason that is given at the assumption of infinite tray infeed. This means for the model that the only dependencies for events to happen concerning finishing the stack of cases are at the PR side of the stack.

No Signal Messaging

A lot of communication between components of the system takes place. This messaging takes some time in the real system, depending on the signal they take 1 to 100 milliseconds. In the SSMPL model this timing is neglected. Besides, these messages do not necessarily contribute in the critical cycle of the system.

Durations Before and after Potential Stops

The SSMPL model takes into account that the TUR as well as the PR can stop on two locations. Whether the robots stop depends on several other events. In the model, the fact whether a robot stops or not pushing a certain case is determined *during* the simulation and not before. The action durations between internal events are however determined *before* the simulation takes place, as they are needed to simulate. Especially for the movement of the TUR from the tray to the PUT, this is a real disadvantage as some synchronisations of the TUR can happen before of after its potential stop in this movement.

The way of modelling such a stop is explained in this section for a deterministic case. Figure 4-4 shows four events in which a possible robot stop is modelled. The four events represent the following:

- event A: The robot starts the movement at the start position.

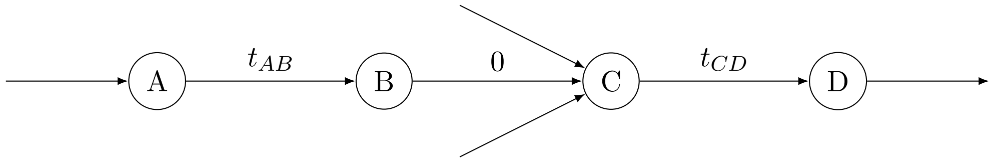


Figure 4-4: Example of modelling a robot stop.

- event B: The intermediate stop of the robot starts.
- event C: The intermediate stop of the robots ends.
- event D: The robot reaches the final destination of the movement.

Events C has some extra arrows pointing at the node as there are additional events that have to happen before the robot can move on after its potential stop. There are two possible situations that can occur, either the robot has to stop or not. In this example the objective is to model that the robot takes 4 seconds on moving from A to D without a stop. If the stop is necessary, the moving time takes 5 seconds, 2 seconds before the stop and 3 afterwards. In order to model the moment that event D happens correct, the values of the unknowns are taken equal to $t_{AB} = 1$ and $t_{CD} = 3$. In this way, a movement without a stop takes 4 seconds, and the duration of the second half after a stop takes 3 seconds. The only situations in which the model does not work correctly is when the events that release the robot from the stop happen in the first second of the intermediate stop. The movement would then take between 4 and 5 seconds instead of 5 seconds.

Timing Variables Not Based on Reference Simulation Loggings

Not all components included in the SSMPL model are logged by the reference simulation performed by Vanderlande. The components of which the timing variables are obtained elsewhere are TPO, DLO, STO, the tray lift, the PUT floor and the TUR unblocking the PUT. The variables of TPO, DLO and STO are based on tests with the actual application. The values of the tray lift, the PUT floor and the TUR unblocking the PUT are based on responses of experts or on internal documents.

Missing Correlations

Not all correlations present in the system are taken into account in the model. An example of a correlation that *is* taken into account is the correlation between the number of cases in the tray and the moving time of the TUR from the tray towards the PUT. When more than one cases are present in the tray, it becomes more difficult to determine a straight forward path from the tray towards the PUT. The consequence is that the average moving time of the TUR from the tray towards the PUT is longer when there are multiple cases in the tray. This correlation is modelled with the help of mode as explained in Section 4-1-2. The presumption is that more correlations are present in the system, however by modelling it some of them are not taken into account.

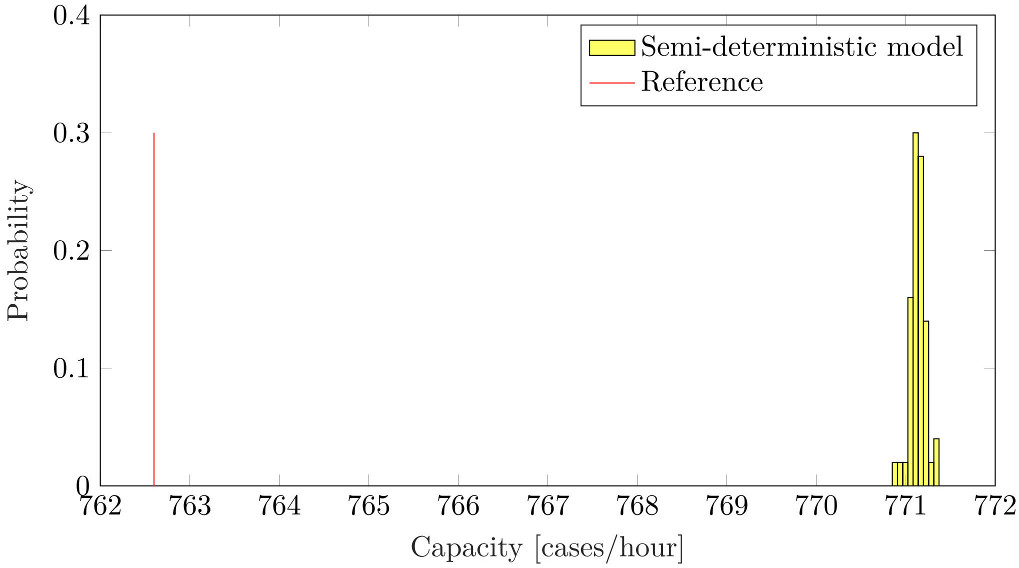


Figure 4-5: Histogram of 50 semi-deterministic model outcomes

4-1-6 Validation

Combining the information given in Sections 4-1-1 - 4-1-4, a Switching Max-Plus Linear (SMPL) model is the result. This model has to be validated in order to conclude on correctness. A reference value that is used for the validation is obtained from a simulation tool owned by Vanderlande called the PR Cycle Time Tool. Logging data obtained from the PR Cycle Time Tool is used to determine timing variables of the the SMPL model Due to this, the timing data are case-specific. However, not all timing variables specified in the SMPL model are logged in the PR Cycle Time Tool. The components of which timing variables are indeed not based on the loggings from the PR Cycle Time Tool are stated in Section 4-1-5. Simulation results obtained by using the SMPL model containing the case-specific timing data, are said to be obtained from the *semi-deterministic model.* The expectation is that the placing times of corresponding cases are the same in both the semi-deterministic model and the PR Cycle Time Tool.

In order to get a good indication of the resulting capacity from the semi-deterministic model, the simulation is done 50 times and the resulting capacities are plotted in a histogram. This histogram can be seen in Figure 4-5. The capacities of the histogram do not match the reference capacity obtained from the PR Cycle Time Tool. After investigation which is further explained in Section 4-1-7 the conclusion is drawn that the PR Cycle Time Tool makes a mistake in the waiting times of the PR when it enters the carrier lift area.

Due to the discrepancy in resulting capacity the simulation is performed with the lift and STO calculation times equal to 0. The lift and STO calculation timings are also excluded in the PR Cycle Time Tool, used for the reference result. The resulting histogram is shown in Figure 4-6. The difference between the modelled capacity and the reference is significantly small.

Despite the similarity in capacity shown in Figure 4-6, there could still be errors in the model. For that reason, the behaviour of several components is analysed. This analysis is performed

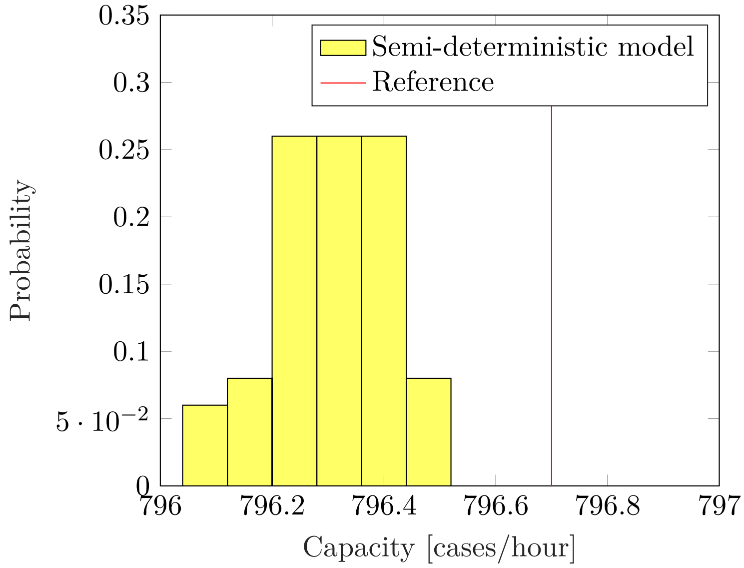


Figure 4-6: Histogram of 50 semi-deterministic model outcomes, the carrier lift and STO are excluded

with the use of a Gantt chart. A Gantt chart is a bar graph that that illustrates a schedule or, in this case, an execution of a simulation. The Gantt chart shown in Figure 4-7 concerns a time span of 15 seconds. The chart describes movements of the three components, the tray lifts, the TUR and the PR. Bars of same colors describe the same case. There are three noticeable occurrences in this 15 seconds:

- Case 100 and case 101 are taken from the same tray. This phenomenon results in no tray exchange bar for case 100. The same holds for cases 102 and 103.
- Case 100 is the first case of a new stack. Due to this, the PR has to wait relatively long before it can perform the action “PR place and return” as it has to wait for the carrier lift finishing the previous stack. This explains the relatively big gap of no action in the middle of the diagram.
- The TUR does not always make a stop on its way from the tray towards the PUT. The presence or absence of this stop determines whether the bars “TUR towards PUT” and “TUR place on PUT” are respectively both or not both drawn for that particular case.

Many dependencies present in the system can be found in the Gantt chart as well. As no errors are found by observing the Gantt chart, the correctness of the model becomes more plausible. Taking this and the capacity result in mind, the SMPL model is approved and is used for further analyses.

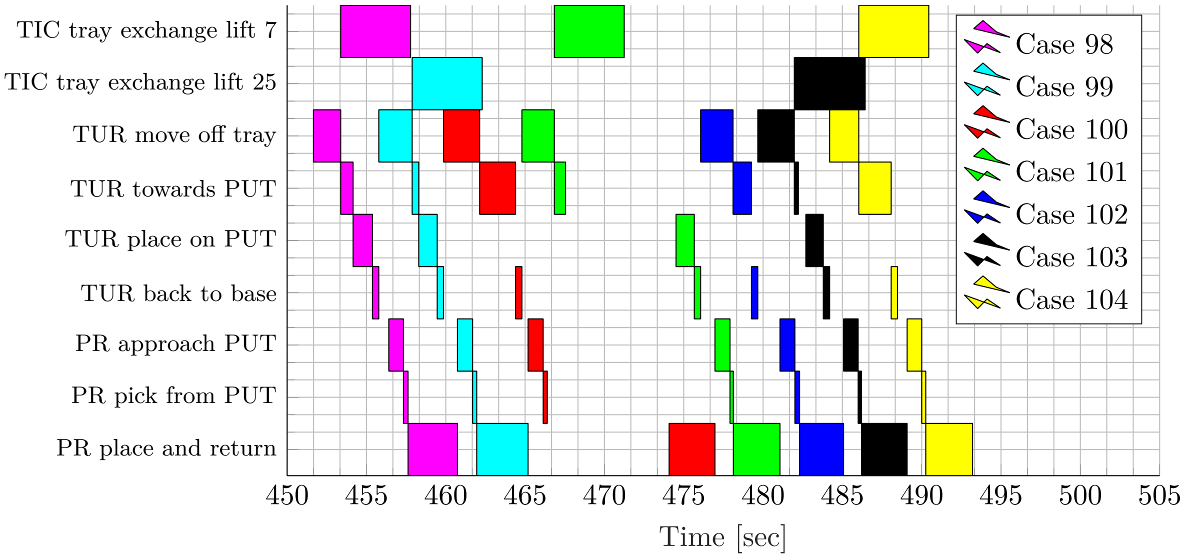


Figure 4-7: Gantt chart of cases 98 - 104 of a simulation of the semi-deterministic model. The carrier lift movements are not put in this Gantt chart which explains the gap.

4-1-7 Double Waiting Times

As mentioned in the previous section, a discrepancy between the SMPL model and the PR Cycle Time Tool causes significant difference in simulations of the total system, this section enlightens that error. The simulation done by Vanderlande has different stadia. First a detailed simulation is done using RobotStudio[1]. This simulation gives case-specific loggings of both the TUR and the PR. Those loggings are determined for a specific Load Forming Logic (LFL) recipe. This recipe describes how the cases should be stacked on the carrier. As the LFL recipe is predetermined, a simulation with the same recipe gives the same result. After the TUR and PR loggings are obtained, it is determined whether the carrier lift has to perform an action before a specific case is placed on the carrier. This in combination with the computation time of STO gives a potential extra waiting time for PR approaching the lift area. As the PR and the carrier lift can perform their actions in parallel, not every approach of the PR towards the lift area results in a stop.

The extra waiting time for the PR is added afterwards, which means that it does not influence the PR loggings obtained from RobotStudio. However, when the PR has to wait before it can enter the lift area it will approach the PUT later compared with a situation when the PR does not wait. The PR waiting times in the original loggings are too long as a consequence. The influence of this wrong approach is about 1.1% on the capacity of 762, 6 cases per hour determined by the PR Cycle Time Tool.

4-2 Stochastic Parameters

The SMPL model has been made, finished and validated. However, the model with the timing variables based on the RobotStudio simulation only represents one LFL recipe as explained

[1]RobotStudio is a simulation tool provided by the robot producer Asea Brown Boveri (ABB)

in Section 4-1-7. This would mean that if the SMPL model is to be used again for another LFL recipe, a new RobotStudio simulation has to be performed. In order to perform realistic simulations without using direct data from the reference simulation, an SSMPL model is made in this section. The stochastic parameters describe how the timing variables, used in different event counters, are distributed. First the stochastic distributions are made for most of the timing variables specified in the model. In order to conclude on the usability of the model, simulations performed with the SSMPL model are validated.

4-2-1 Timing Variable Distributions

All the timing variables explained in Section 4-1-3 need a certain value in the model to be simulated. In the SSMPL model these values are random according to a stochastic distribution. The validation of the semi-deterministic model is performed in Section 4-1-6 with case-specific values obtained from the PR Cycle Time Tool. Using the semi-deterministic model, a RobotStudio simulation is needed for every unique LFL recipe. Distributions of timing variables are based on histograms made using RobotStudio loggings. Three known distribution equations are fit optimally to the histograms. The advantage of using distribution described by functions is that they can be used analytically. As the fits do not seem to be good enough, the choice is made to exactly fit a piecewise linear distribution to the normalised histograms.

Existing Distributions

In order to use the results from the histograms, they are normalised first such that they have a surface equal to 1. In this way the shape of the histogram can be used as a Probability Density Function (PDF). The analysis performed in Chapter 6 makes use of analytical properties of the normal distribution. For such purposes there is an advantage in describing the distributions by known equations. Another advantage of known distributions is that many characteristics are known and proven. The three distributions are the modified PERT distribution, the Weibull distribution and the (adjusted) normal distribution. The choice for these distributions is explained in this section.

The modified PERT distribution is a specific version of the beta distribution in which the minimum, maximum and most likely value are specified. Due to the provided histogram, the minimum and maximum value of the timing variable duration are known. Furthermore most histograms show a most likely value by having a peak. Usually, the histograms of process times have their peak closer to the minimum value than to the maximum value. As a result the histogram has a triangular shape which makes the modified PERT distribution a good choice [9]. The modified PERT distribution is shaped to the histogram by adjusting the parameters according [10].

The Weibull distribution [11] is chosen because it is restricted to positive values. Furthermore a lot of this distribution is known. An additional advantage is that it is also implemented into MATLAB.

The normal distribution is the distribution that is used very often. Performing calculations with this distribution and drawing conclusions is relatively easy. As normal distribution can

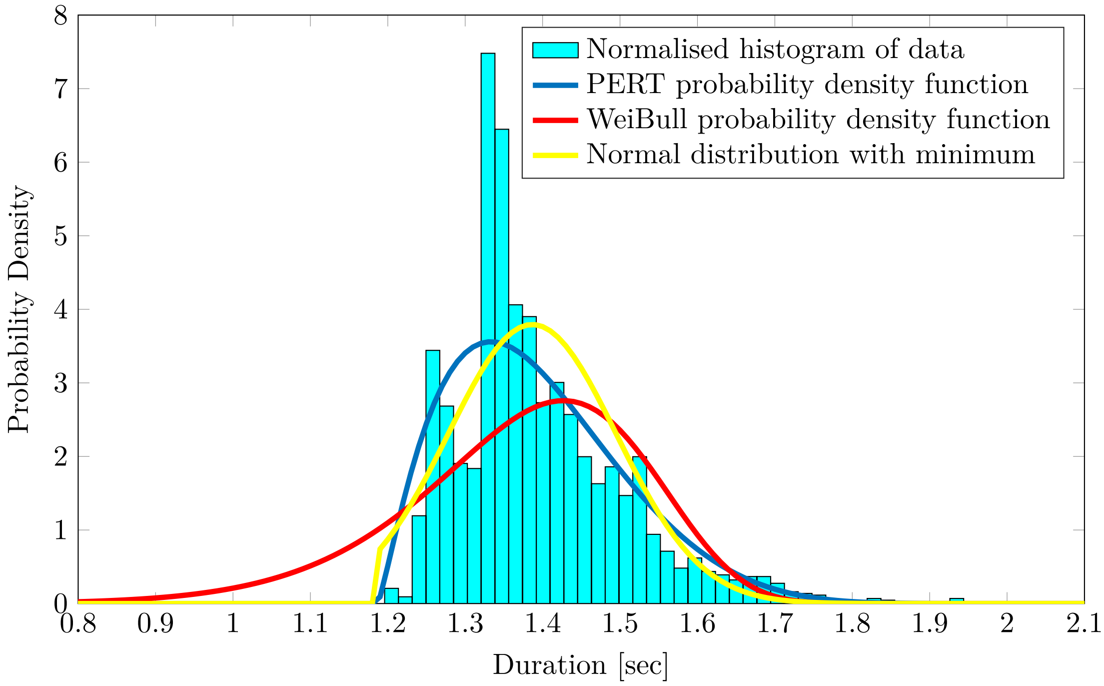


Figure 4-8: Histogram of a timing variable with three optimised distributions

contain negative values, the distribution is manually cut off at the minimum value of the histogram. Consequently the remaining part is scaled such that the surface of the PDF is equal to 1.

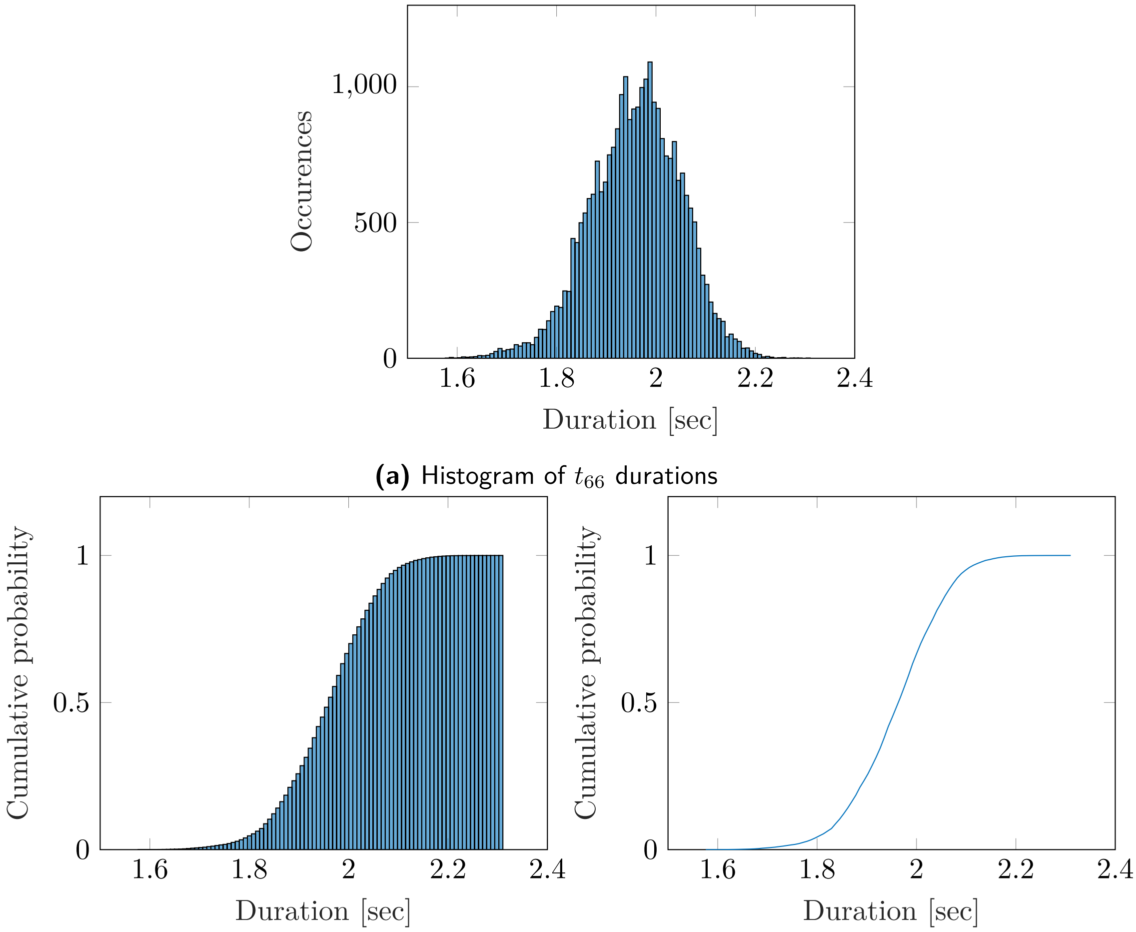
An example of a histogram on which the three distributions are fit is shown in Figure 4-8. For this histogram, the distributions all fit quite well. However during simulations of the SSMPL model it became clear that little adjustments on parameters of the distributions had significant influence on the resulting capacity. Therefore the choice is made to work with perfectly fitting piecewise linear distributions. This is not a problem for simulation purposes.

Piecewise Linear Distributions

The perfectly fitting piecewise linear distributions are fit exactly onto the normalised histogram. This means that taking a high number of samples out of the resulting distribution results in the same histogram. These distribution are obtained performing the following steps:

1. Create the histogram of the timing data.
2. Normalize the histogram such that its surface is equal to 1.
3. Determine the corresponding PDF and Cumulative Distribution Function (CDF).

The making of the piecewise linear distribution is shown here for timing variable t_{66}. This timing variable represents the movement of the PR from the position where it enters the STO camera vision area towards the position where it places the case on the stack. The histogram



(b) CDF Histogram of t_{66} durations

(c) CDF plot of t_{66} durations

Figure 4-9: Obtaining the CDF of t_{66}

of t_{66} values are shown in Figure 4-9a. The width of the histogram bins is optimised in order to obtain a similar shape compared to the corresponding PDF [12].

The resulting histogram is turned into a CDF version of the histogram shown in Figure 4-9b. The chosen CDF which is used for the simulations is shown in Figure 4-9c. When a value for t_{66} is needed in a simulation of the SSMPL model, it is obtained by selecting the x-value corresponding to a random value in the range $(0, 1)$. This is similar to taking a random value from the distribution corresponding to the original histogram in Figure 4-9a.

4-2-2 Validation

To validate the established SSMPL model, it is tested in several ways. First simulations are performed for the entire order set containing 27346 cases, the same order set as simulated with RobotStudio. The resulting model is compared to the values obtained by RobotStudio, as its correctness is trusted. For the validation of the entire order set, the timing variables are based on the entire set and the simulation is done on this set. Another validation is performed

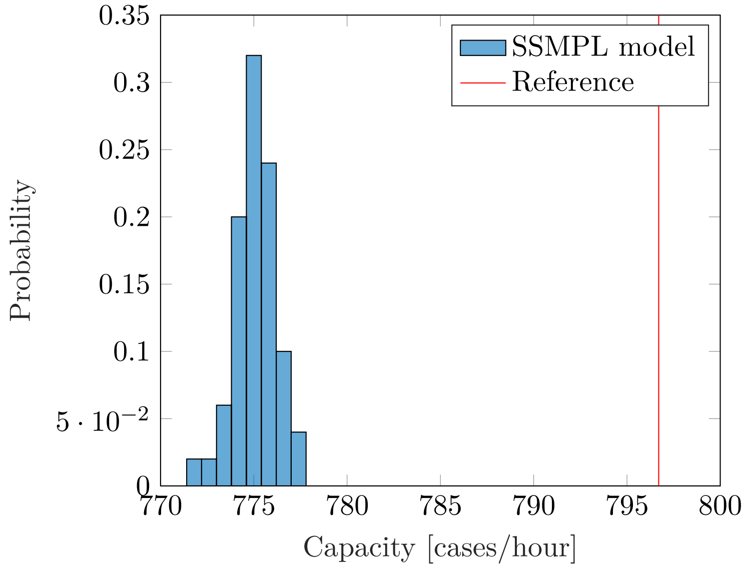


Figure 4-10: Histogram of 50 SSMPL model outcomes, the carrier lift and STO are excluded

in which the total order set is split into an identification part and a validation part. This tests whether establishing an SSMPL model is useful for using it on different order sets.

Validation for Total Reference Set

The validation of the SSMPL model for the total set of orders is done by comparing it to the outcome capacity of the PR Cycle Time Tool. First, this validation is done with the SSMPL model of which the carrier lift and STO are excluded. This is done because it was shown in Section 4-1-6 that the semi-deterministic model excluding carrier lift and STO approximates the reference good enough. The resulting histogram of the simulation is shown in Figure 4-10. The capacities resulting from the SSMPL model differ significantly from the reference capacity. The expectation is that the combination of maximum operators and random values is the reason for the SSMPL model results to have a lower capacity and thus a higher process time. Possible combinations of durations of timing variables within a maximum operator all become random due to the random timing variables. Somehow the different timing variables for a single case or consequent cases are depending on each other. This dependence disappears when timing variables become random. The chance that all random values are "low in their distribution" becomes smaller when more distributions are added to the maximum operator. Furthermore high outliers are always taken into account. Therefore the presumption is that not taking into account all the correlations in an SSMPL model gives a higher process time than the "real" process time. More research to this subject is desirable. The choice is made to use the SSMPL model for analyses despite the discrepancy as the difference compared to the semi-deterministic model stays around the same.

The SSMPL model is also validated for the whole system, including the carrier lift and STO. For this validation, 50 simulations are done with the SSMPL model and 50 simulation are done with the semi-deterministic model. The histograms are plotted in Figure 4-11, the difference of average capacity and capacity variation is visible. The capacities resulting from the SSMPL

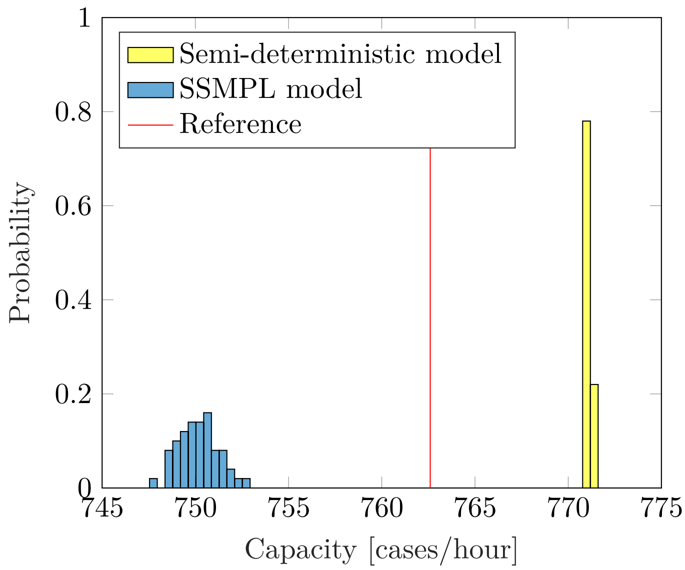


Figure 4-11: Histogram of 50 SSMPL and 50 semi-deterministic model outcomes modelling the total system

model differ significantly from the results obtained from the semi-deterministic model. The difference between the two is about the same compared to the difference of the SSMPL model and the reference when the carrier lift and STO were excluded, shown in Figure 4-10. So ex- or including the carrier lift and STO does not influence the lower capacity result of the SSMPL model.

Separate Identification and Validation Data

The SSMPL model uses stochastic distributions to determine the values of the timing variables. It is of high value that if the SSMPL model is applicable for other input files and not only for the reference set used for timing variable identification This means that the distribution is determined using one logging file provided by the PR Cycle Time Tool and that new simulations can be made with the SSMPL model for other LFL recipes. This is tested by dividing the total order set into two parts. One part is used for identification of the timing variable distributions, the other part is used to validate the obtained distribution. This way of testing is called cross-validation testing as states in [13].

In Figure 4-12 the result is shown of the simulation of order 11 to 40. For the identification of the distribution of the timing variables, orders 1 to 10 are used. The simulation results are likewise to the results shown in in the previous section. This indicates that it is possible to use this SSMPL model for new or other data as well.

This analysis is also done for the situation when the identification and verification set are of equal size. The result of that analysis can be seen in Figure 4-13. As the order set for validation in this simulation differs from the previous, the expected capacity is also different than in Figure 4-12. Also in this situation, the result is as desired. This indicates that it is also possible to simulate a dataset of equal size to the identification dataset.

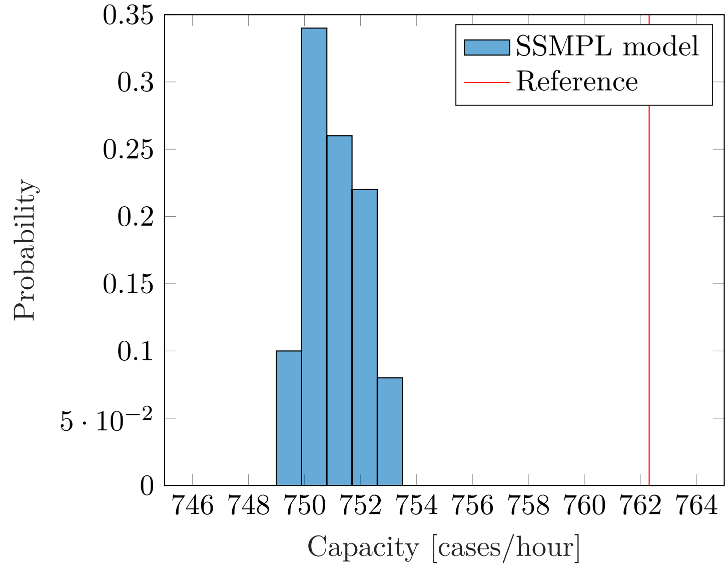


Figure 4-12: Histogram of 50 SSMPL model outcomes. Orders 1-10 are used for identification, orders 11-40 are used for verification.

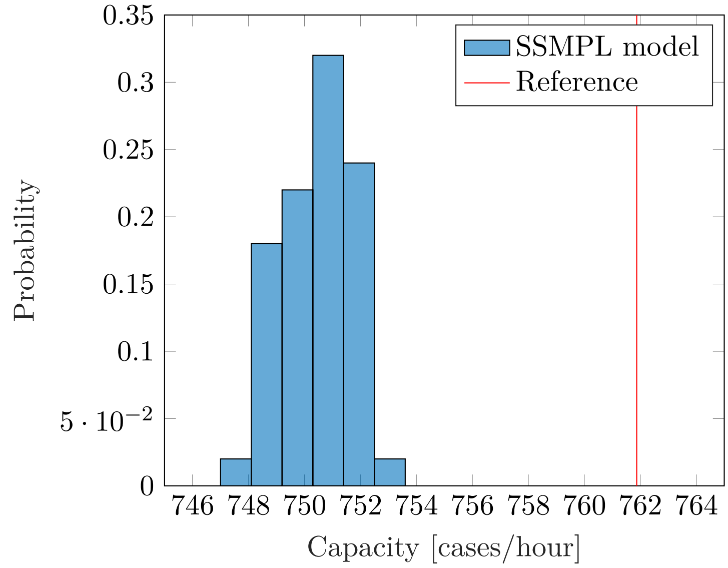


Figure 4-13: Histogram of 50 SSMPL model outcomes. Orders 1-20 are used for identification, orders 21-40 are used for verification.

Chapter 5

System Analyses

The established Switching Stochastic Max-Plus Linear (SSMPL) model of the Automated Case Picking (ACP) Automatic Palletiser System is approved for doing analysis. Conclusions on these analyses can be used in making decisions on for example system design.

First theory on the max-plus Lyapunov exponent is used to draw conclusions on asymptotic growth rate and corresponding capacity.

Secondly a bottleneck analysis is performed which gives insight on which component are determinant on the capacity of the system. The bottleneck analysis is split into three parts, based on critical circuits, on cycle times and on activity periods.

Next an analysis is performed on the sensitivity of the system. This means that changes are made to the system which influences the outcome of the system. The bottleneck analysis is used in order to get pre knowledge on which components of the system are likely to be of more influence than other parts.

As a result some thoughts are shared on what could be done to improve the system with focus on the resulting capacity of the system.

5-1 Lyapunov Exponent

The Lyapunov exponent is not defined for a switching recurrence relation. However it is still possible to give useful information on the established SSMPL model using the max-plus Lyapunov exponent. Having the recurrence relation stated in (3-39) and a corresponding sequence $\{A(k) : k \in \mathbb{N}\}$, the max-plus Lyapunov exponent explained in Section 3-3-2 is equal to the asymptotic growth rate of $x(k)$ [7]. The asymptotic growth rate which can be seen as the average cycle time of the system which is linked to capacity. In the analysis performed in this section, A-matrices corresponding to two modes are constructed with the expected values of the timing variables and are consequently combined.

The situation sketched here is a combination of ordered cases handled by the two lifts. The most occurring situation for both lifts is a case taken as first from the tray while multiple cases are located in that tray (modes I.5 and I.7). Of the second part, the most occurring mode is II.6 which describes the situation in which the carrier lift does not change height before the case is placed. The A-matrices of modes I.5-II.6 and I.7-II.6 are denoted in this example by respectively $A_{5,6}$ and $A_{7,6}$. Matrix $A_{5,6}$ describes a case that is handled by lift 7 and $A_{7,6}$ describes a case that is handled by lift 25. For this example, both matrices are constructed using the expected values of the timing variables as this is necessary for calculating the Lyapunov exponent of a stochastic model. The A-matrix that describes the combination of one case handled by lift 7 and a consequent case handled by lift 25 is obtained by max-plus multiplication as shown in Equation (5-1).

$$A = A_{7,6} \otimes A_{5,6} \tag{5-1}$$

The resulting A-matrix is shown in Equation (A-1) in Appendix A-4. This matrix is reducible as explained in Section 3-1-3. Furthermore the matrix has fixed support, which in practice means that the ε-terms are always on the same place within the matrix. The calculation of the max-plus Lyapunov exponent corresponding to the resulting A-matrix is performed with the use of MATLAB by approaching the limit to ∞ given in Equation (3-46). As a stop criterion for the approach, the maximum absolute value of the differences between two subsequent values of $\frac{x_j(k;x_0)}{k}$ have to be smaller than $1\mathrm{e}^{-10}$ for all j. The resulting approach of the Lyapunov exponent vector is taken equal to $\frac{x(298413;x_0)}{298413}$ as the stop criterion is reached after 298413 iterations. This vector λ is given in (A-2) in Appendix A-4.

All values of λ are equal to 9.585 when rounded off which is equal to the asymptotic growth rate of the state vector. This is translated to be the asymptotic cycle time of the placement of two cases, both handled by the different lifts. With this information, a conclusion can be drawn on the corresponding capacity. This is equal to $\frac{7200}{9.585} = 751.2$ cases per hour. In this calculation, 7200 is used the placement of two cases is described in one cycle time and two hours equals 7200 seconds. This analysis concludes that a capacity of 751.2 cases per hour is reached when only trays containing multiple cases of which only one case is taken are used. Furthermore the assumption is made that the carrier lift is not included in the model as only modes are chosen that describe situation in which the carrier lift does not move between cases. Including the carrier lift would result in a lower capacity. This means that the desired capacity of 800 cases per hour will most probably not be reached when cases are only taken as first from tray with multiple cases in it.

5-2 Bottleneck Analysis

The automatic palletiser is a complex system with many synchronisations. Due to these multiple synchronisations, drawing conclusions on the characteristics of the system becomes more difficult. Pointing out the bottleneck also becomes less obvious due to this complexity. The bottleneck in a synchronised system is defined in [14] as the resource that by throughput increasement causes an increase of the system's throughput. As the automatic palletiser is a dynamic system and not deterministic, the bottleneck is even more difficult to pinpoint. Not for every case, the systems acts in the same manner. This phenomena makes it possible to have different bottlenecks when focussing on certain time intervals at other moments. So several bottlenecks with a certain contribution are denoted instead of one “overall bottleneck”.

Three approaches are considered to draw conclusions on the bottleneck of the automatic palletiser. The first approach is based on calculating the critical circuit of the system. The second approach is by looking at the cycle times of the whole system and comparing these times per mode. The final approach is done by performing shifting bottleneck analysis based on [15].

5-2-1 Critical Circuit Analysis

The automatic palletiser consists of several robots that work together as a whole as explained in Section 2-2. All the different components have their own task which they do repeatedly for every case again. The slowest component determines the capacity, and thereby the capacity, of the overall system. The reason for a component being the slowest is not necessarily obvious due to the many dependencies and synchronisations present in the system. In this section, the critical circuit of the system is detected. The combination of several actions determine the critical circuit. The frequency of actions being present in the critical circuit is used to determine the rate of "bottleneckness" per action. This approach is based on [16] in which analysis is done on Gantt charts. First [16] gives a method on how flow diagrams, represented by topological graphs, can be obtained from Gantt charts. After that the critical path of this graph is determined as this answers the "what is the bottleneck" question, according to [16].

Theory

The result of a simulation with the established SSMPL model of the system is a state vector. This state vector contains the time instants at which the modelled internal events have occurred for the last time. The last state vector of the simulation is denoted by $x(l)$. This $x(l)$ is obtained by repeatedly multiplying the state vector by the corresponding A-matrix. The calculation in Equation (5-2) is performed to obtain $x(l)$.

$$x(l) = A(l) \otimes \cdots \otimes A(1) \otimes x(0) \tag{5-2}$$

Every max-plus matrix-vector multiplication can be graphically represented with a topological graph as shown in Section 3-1-3. The sequence of l matrices in Equation (5-2) can also be rewritten to one matrix as shown in Equation (3-40). This results in matrix $A[l, 1]$ which is obtained as in Equation (5-3). The topological graph corresponding to $A[l, 1]$ represents the total system evolution from the initial step to the final step.

$$A[l, 1] = A(l) \otimes \cdots \otimes A(1) \tag{5-3}$$

The critical circuit of a graph representing a system's A-matrix determines the periodic behaviour of powers of A [7]. The resulting $A[l, 1]$ matrix represents the simulation of the palletiser system for l cases. The values of $A[l, 1]$, in which l reaches is in the limit to ∞, can be used to draw conclusions on the average behaviour of the system. As the infinity cannot be reached in simulation, an approach to infinity can be established by taking l large. The larger l is taken, the better the approximation of the limit is.

For a large value of l, the computation of $A[l,1]$ becomes a complex and long calculation. Furthermore, the interpretation of the matrix's elements is difficult. The matrix $A[l,1]$ is the relation between the initial state $x(0)$ and the resulting state $x(l)$ as can be seen in Equation (5-4). This means that the communication graph related to $A[l,1]$ also gives the information of the evolution from initial state to the l'th state. The single arcs in this communication graph are difficult to interpret as they are a combination of many maxima operations.

$$x(l) = A[l,1] \otimes x(0) \tag{5-4}$$

The critical circuit of the resulting communication graph related to matrix $A[l,1]$ is denoted by $\mathcal{C}^c(A[l,1])$. The average weight of $\mathcal{C}^c(A[l,1])$ is denoted by λ_l and called an eigenvalue of $A[l,1]$. The value of λ_l is equal to the time it takes to get from the initial state to the final state. As λ_l is derived from $\mathcal{C}^c(A[l,1])$, the construction of the critical circuit is of importance. The critical circuit $\mathcal{C}^c(A[l,1])$ is a collection of one or more arcs in the communication graph of $A[l,1]$. The weight of these arcs are combinations of maxima and sums of other maxima and sums as $A[l,1]$ represents a max-plus multiplication sequence of l matrices.

From step 1 to step l, every intermediate step k also results in an intermediate sequence matrix $A[k,1]$. Every matrix $A(k)$ contains the values corresponding to the timing values that hold at time step k. This means that the sequence matrix $A[k,1]$ is built up by the timing variables that cause the maxima. For the last sequence matrix $A[l,1]$, all the elements correspond to a combination of timing variables as well. Timing variables are linked to different actions and those actions are consequently linked to different robots in the system. The timing variables that contribute the most to the critical circuit are of interest as these are the most determining timing variables of the system.

Problem Approach

The communication graph of the resulting A-matrix is implemented into MATLAB as a directed graph. Consequently the elementary circuits of the communication graph are determined using the algorithm explained in [17]. Elementary circuits are circuits in a graph without duplicate nodes as explained in Section 3-1-5. By calculating all the average weight of the elementary circuits, the critical circuits are determined which have the longest average weight. Usually, there is just one circuit as the timing variables differ a lot.

In order to determine which timing variables contributed to this critical circuit, the matrix multiplication is adjusted. In an additional data structure, the timing variables that cause the maximum are saved instead of only the resulting values of the max-plus multiplication. Only the timing variables that contribute in the maximum are saved. After the sequence of matrix multiplications, the timing variables that caused the maximum values in the resulting matrix are obtained. Consequently the critical circuit of the resulting matrix is determined, the corresponding timing variables are listed and sorted.

Results

The critical circuit analysis is performed on a simulation using the SSMPL model. The result of the analysis is a table containing all the timing variables of the model. The percentages of

Table 5-1: Percentages of contribution of the most determining clustered timing variables

Timing variables	Description (The time it takes to:)	Percentage of contribution	Average duration [sec]
t_3, t_{13}, t_{23} and t_{33}	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT	15.87%	1.23
t_1, t_{11}, t_{21} and t_{31}	move the TUR from TUR stop (a) towards the position where it starts pushing the case	13.92%	1.28
t_2, t_{12}, t_{22} and t_{32}	move the TUR from the position where it starts pushing the case towards TUR stop (b)	13.82%	1.61
t_{47}	move the TUR from the position where it placed the case on the PUT towards TUR stop (a)	13.75%	0.40
t_{66}	move the Palletiser Robot (PR) from the PR stop (b) towards the position where it placed the case on the stack	8.91%	1.96
t_{67}	move the PR from the position in the PUT towards PR stop (b), 100 mm above the PUT	8.53%	0.24
t_{62}	move the PR from the PR stop (a) towards the position in the PUT	8.53%	1.00
t_{51}	move the PR from the position where it placed the case on the stack towards PR stop (a)	6.70%	1.04

participation of all these timing variables in the construction of this critical circuit are shown in Table B-1 in Appendix B. As there are timing variables that describe the same movement but in different modes, the contribution is also determined while those timing variables are clustered. Two tables are made, one with all the separate timing variables and one with clustered timing variables. An example of this phenomenon is the movement of the Tray Unloading Robot (TUR) from the PickUp Table (PUT) towards a case in the tray. This movement is described in four different timing variables as the distinction is made between the two tray lifts and between one or more cases in a tray. The resulting table in which the corresponding movements are clustered is shown in Table B-2 in Appendix B.

The most determining clustered timing variables are shown in Table 5-1. The TUR actions contribute most often in the construction of the critical circuit. An important note in this analysis is that the percentages are made on *occurrences* of the specific timing variables. This means that durations of the different timing variables are not taken into account in these percentages. The average durations of the corresponding timing variables are also stated in Table 5-1.

5-2-2 Cycle Times

The resulting capacity of the automatic palletiser is related to on the case placements on the carrier. The durations between every case placement are called cycle times and they change as circumstances differ per case. The analysis performed in this section has its main focus on

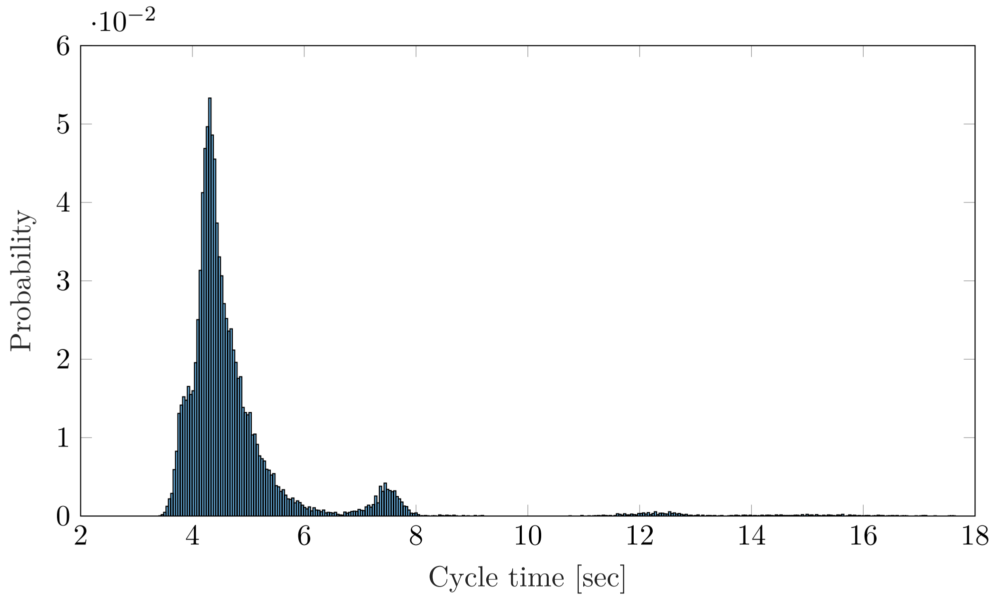


Figure 5-1: Histogram of cycle times of a simulation using the SSMPL model

these cycle times. As every case is linked to a specific mode, the corresponding cycle time is also linked to that mode. The average cycle time is calculated for every mode. These values are compared to each other in order to draw conclusions on influences of certain modes on the cycle time.

First Observations

The total distribution of all cycle times of a simulation using the SSMPL model is plotted in a histogram in Figure 5-1. The different peaks in the histogram indicate that the cycle times are not independently determined. The smaller peak in the histogram that contains cycle times between the 6.75 and 8 seconds are mostly linked to cases that belong to mode II.1 and II.4. Recall (Section 4-1-2) that cases that belong to mode II.4 are placed after a lift takeover that took place during the making of the stack. The cycle times that have a duration longer than 8 seconds are mostly linked to cases that belong to mode II.2 and II.5. Cases that belong to mode II.5 are placed after a slip sheet placement. By simple observation it becomes clear that cycle times depend on the mode.

Cycle Times per Mode

Cycle times per mode give an insight on which modes cause long or short cycle times. Table 5-2 is generated containing the percentual occurrence and the average cycle time of the different modes. The modes that concern cases that are not taken as first from a tray (II.2, II.4, II.6 and II.8) only occur in combination with the carrier lift doing nothing. These situations are not taken into account in the reference simulation, despite the fact that this is not a constraint of the system. The average cycle times show that the mode type of part II is very determinant in whether the cycle time is long or not. Modes II.2-II.3 describe situations with actions at

Table 5-2: Percentual occurrence and average cycle time per mode in a simulation using the SSMPL model

Mode	II.1: New stack	II.2: Lift takeover & new stack	II.3: Slip sheet	II.4: Lift takeover	II.5: Lift lowering	II.6: Nothing
I.1: One case, first, lift 7	0.53% 7.61s	0.07% 12.04s	0.14% 12.42s	0.66% 7.46s	4.33% 4.43s	13.08% 4.35s
I.2: One case, not first, lift 7	-	-	-	-	-	3.89% 4.42s
I.3: One case, first, lift 25	0.57% 7.66s	0.07% 11.87s	0.06% 12.14s	0.63% 7.47s	4.22% 4.49s	13.56% 4.41s
I.4: One case, not first, lift 25	-	-	-	-	-	3.78% 4.38s
I.5: Multiple cases, first, lift 7	0.68% 7.72s	0.09% 11.86s	0.22% 12.43s	0.57% 7.38s	5.26% 4.77s	19.60% 4.69s
I.6: Multiple cases, not first, lift 7	-	-	-	-	-	0.94% 4.75s
I.7: Multiple cases, first, lift 25	0.67% 7.57s	0.10% 12.14s	0.22% 12.49s	0.57% 7.48s	5.21% 4.72s	19.34% 4.71s
I.8: Multiple cases, not first. lift 25	-	-	-	-	-	0.90% 4.80s

the side of the carrier lift that take a relatively long time. Modes II.5-II.6 on the other hand describe mode with actions that take a relatively short amount of time. However, the percentual occurrences show that the "expensive modes" occur the least. The combination of percentual occurrence and average cycle time is an indication on how determinant a certain mode is for the total process time. In Table 5-2, less determinant modes are more green and the more determinant modes are more red.

By clustering modes more conclusions can be drawn. The division is made on whether the case is taken from the tray while it is the only case in the tray or not. The results of clustering modes I.1-I.4 and modes I.5-I.8 are shown in Table 5-3. The difference is significant and the percentual occurrence of both clusters are close to each other. It can be concluded that with the use of this analysis a good insight can be given into the influences of system changes over time.

Table 5-3: Resulting cycle times per cluster: one case or multiple cases in the tray

	One Case in Tray	Multiple Cases In Tray
Percentual Occurrence	45.61%	54.39%
Average Cycle Time [sec]	4.62	4.93
Cases per hour	778.73	729.76

5-2-3 Shifting Bottleneck Using Active Periods

Detection of bottlenecks can be done in different ways. In production systems for example, the focus can be on queue length, waiting times or machine activity. With the focus on activeness of the machines, one can look at the percentual activeness of the total simulation or the separate active periods can be taken into account. In [18] a method is proposed based on separate active periods. This method takes into account the active duration of the robots, because only when a robot is active it can be the direct cause for another robot to be inactive. By doing this, conclusions can be drawn on the average duration of these active periods.

Active Durations

The detection of the systems bottleneck by looking at the active periods has two approaches. The first approach proposed in [18] takes into account all the active periods of all the robots. After these periods are obtained, the average value of these active durations are calculated. The average durations of the active periods per robot are to be compared. This gives an indication on how much each robot contributes on being limiting for the overall system.

The robots of the automatic palletiser of which their activeness has been logged after simulation are the both tray lifts, the TUR, the PR and the carrier lift. This means that is is possible to see whether a robot is active and when a robot is not active. For the the first 60 seconds of the simulation, these active periods are plotted in Figure 5-2. In order to get an indication on how limiting each robot is, the average duration of the active periods is calculated for every robot and listed in Table 5-4. This indicates that the TUR is the bottleneck. Furthermore the PR would be limiting the most after that, although a lot less. Next, the both tray lifts are equally limiting, again less than the PR. In the next section a more advanced method is presented, the diagram from Figure 5-2 is used again for the implementation of this advanced method.

Table 5-4: Simulation results of SSMPL model

Robot	Average duration of active periods [sec]
Tray lift 7	5.05
Tray lift 25	5.07
The TUR	19.56
The PR	7.65
The carrier lift	2.14

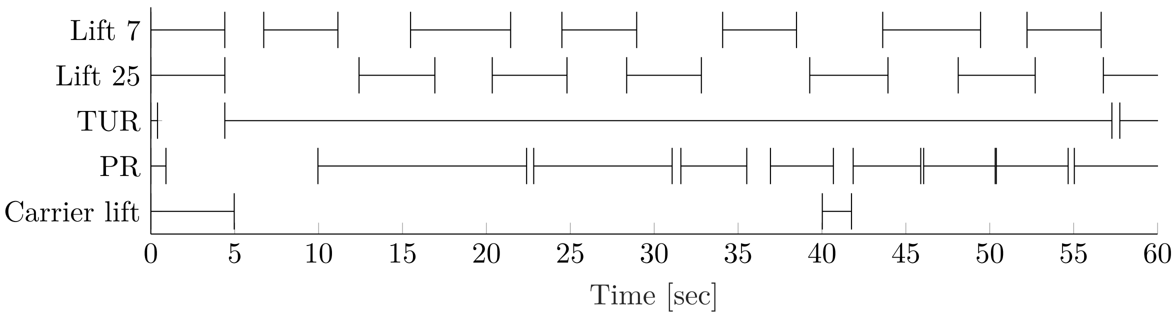


Figure 5-2: Activeness of the different robots of the first 60 seconds of a simulation of the SSMPL model

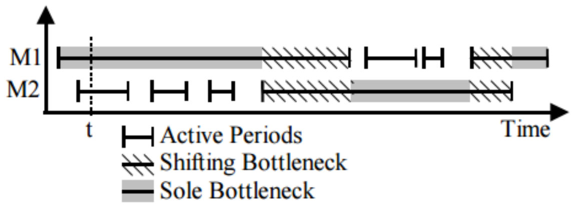


Figure 5-3: Shifting bottleneck example (taken from [15])

Momentary and Average Bottlenecks

An extension on the previous method [18] is proposed by the same authors in [15] using theories from [19]. By looking at the average durations of the active periods of the different robots it was already shown that there is not one overall bottleneck.

However, instead of looking at the averages, one can also point out the longest active durations per time instant. The machine with the longest active period at a single time instant is called the momentary bottleneck. Consequently, the duration of this active period is called the current bottleneck period. Only after the full simulation, this analysis can be performed as information on the durations of the entire active periods is needed. An illustration from literature is shown in Figure 5-3. In this example the momentary bottleneck is determined at time instant t. Both machines M1 and M2 are active at this time instant. As the active period of M1 is longer than the active period of M2, machine M1 is the momentary bottleneck at time instant t.

Instead of the momentary bottlenecks, usually there is interest in knowing the average bottlenecks over a period of time. For this reason the distinction is made between sole and shifting bottlenecks. This phenomenon is also shown in Figure 5-3. In order to classify the bottlenecks, the momentary bottlenecks are identified first. Consequently, the overlap of the all the momentary bottlenecks is identified as shifting bottlenecks. The other parts of the momentary bottlenecks are sole bottlenecks. In Figure 5-4 the bottleneck determination is applied to the first 60 seconds of a simulation using the SSMPL model. In this time span the TUR has far the most sole bottleneck contribution. Although the PR has a little sole bottleneck contribution, its shifting bottleneck contribution shows that its activity is also determining.

This analysis is done for the entire simulation of 27346 cases and the results are combined.

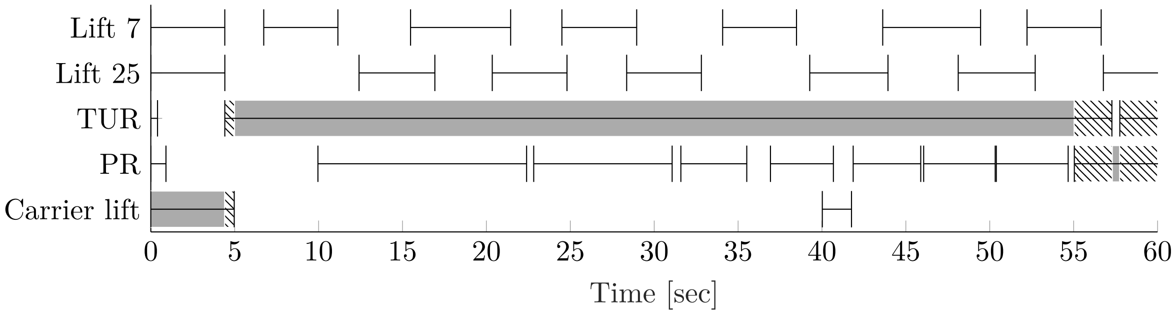


Figure 5-4: Sole and shifting bottlenecks of the first 60 seconds of a simulation of the SMPL model

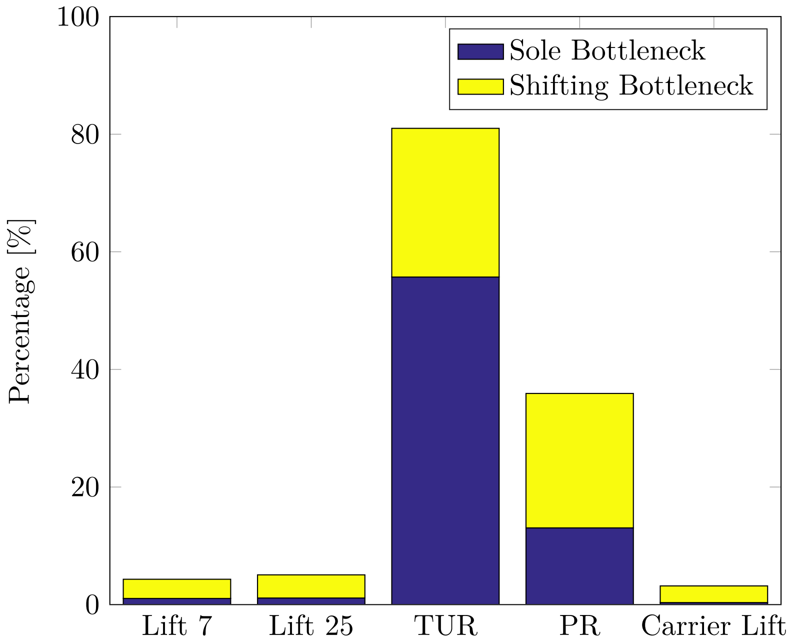


Figure 5-5: Average bottlenecks, Sole and shifting

The resulting bar graph in Figure 5-5 shows the bottleneck percentages of the total simulation time. As there can only be one sole bottleneck at every time instant, the sum of all the sole bottleneck percentages can be at most 1. Once only the sole bottleneck percentages are taken into account, the TUR jumps out as highest percentage even more.

Both the analysis on average active period and on shifting bottlenecks, the TUR activities are classified as most determining. Concluding, this analysis concludes that the TUR is the most limiting robot of the automatic palletiser.

5-3 Sensitivity Analysis

Sensitivity analysis is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input [20]. This analysis is performed on the SSMPL model of the automatic palletiser system. The output of the SSMPL model is the resulting capacity. The different sources of uncertainty in the model input are the timing variables per different component of the system.

In this section the timing variables of the TUR and the PR are manually changed by different proportions. Consequently the change in the resulting capacity is observed in order to conclude on these changes. Furthermore, system structure changes are modelled and simulated in order to conclude on the result of these changes.

5-3-1 Timing Variable Adjustment

For performing a simulation with the use of the SSMPL model, timing variables represent certain actions of certain components of the system. By adjusting the values of these timing variables it is possible to simulate the system with certain actions being faster or slower. Those adjustments are applied to the TUR and the PR. In order to simulate speed change for the TUR timing variables t_1 - t_{40} and t_{47} are adjusted. In order to simulate speed change for the PR timing variables t_{56}, t_{61}, t_{62} and t_{65} - t_{68} are adjusted. The timing variables are listed and explained in Appendix A-2.

The timing variables are made smaller and bigger to simulate the corresponding actions respectively faster and slower. These adjustments are performed with the factors 0.90, 0.95, 1.05 and 1.10. Tables 5-5 and 5-6 show the results of the adjustments. The generated timing variables of one simulation with the SSMPL model are used for this analysis. Both the TUR and the PR result in significant capacity changes as a result of adjusting the timing variables. Adjustments on TUR movements make the biggest difference on the capacity.

Table 5-5: Adjustments to the TUR timing variables

Adjustment factor	Resulting capacity	Difference to original	Percentual difference
0.90	779.97	+30.74	+4.10%
0.95	768.23	+19.00	+2.54%
1.00	**749.23**	-	-
1.05	725.14	−24.09	−3.22%
1.10	699.20	−50.03	−6.68%

Table 5-6: Adjustments to the PR timing variables

Adjustment factor	Resulting capacity	Difference to original	Percentual difference
0.90	763.16	+13.93	+1.86%
0.95	757.38	+8.15	+1.09%
1.00	**749.23**	-	-
1.05	737.97	−11.26	−1.50%
1.10	723.61	−25.62	−3.42%

TUR Adjustment Capacity Margin

By increasing the TUR speed more than in the precious section, the resulting capacity of the system increases as well. However, this increase of capacity has a limit as the rest of

Table 5-7: Absolute and percentual capacity change due to varying TUR timings

TUR timings ratio	Capacity [cases / hour]	Extra capacity [cases / hour]	profit ratio [extra capacity · TUR timings ratio]
1.00	749.23	0.00	0.00
0.95	768.23	19.00	18.06
0.90	779.97	30.74	27.67
0.85	785.84	36.61	31.12
0.80	788.77	39.54	31.63
0.75	790.58	41.35	31.02
0.70	791.98	42.75	29.93
0.65	793.23	44.00	28.61
0.60	794.43	45.20	27.12
0.55	795.56	46.33	25.49
0.50	796.65	47.42	23.71
0.20	799.29	50.06	10.01
0.10	799.30	50.07	5.01

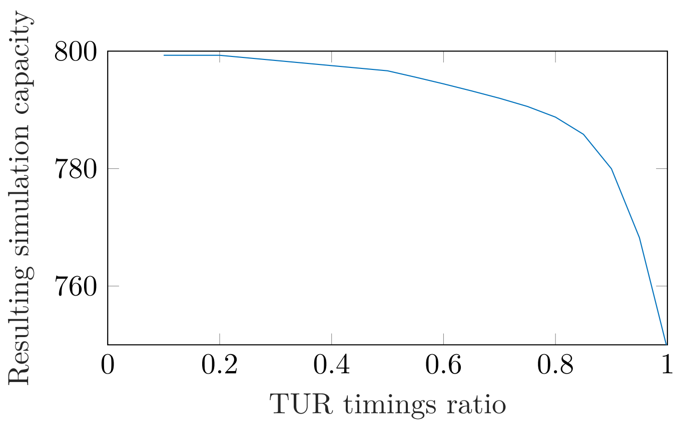


Figure 5-6: Capacity change due to varying TUR timings

the system still has its limitations. By adjusting the TUR timing variables, Table 5-7 and the graph in Figure 5-6 are made. The limiting capacity of the systems without the TUR is just below the 800 cases per hour. The capacity converges to this value as the TUR timings get smaller. This is phenomenom is described by the law of diminishing return [21] which states that you get less and less output when additional doses of an input are added while other inputs are fixed. In the context of the automatic palletiser the extra speed of the TUR would be the added input and the components other than the TUR the other inputs. An interesting value to look at is the rate of profit. To determine this value the TUR timing ratio is multiplied to the extra capacity. The highest rate of profit, approximately 31.35, is obtained at the TUR timings ratio of approximately 0.81.

5-3-2 System Structure Adjustment

By simple adjustments to the structure of the system, situations can be simulated that are different from reality. These simulation are of interest as possible changes to the system can improve the system, however the result is not always obvious. In this section, the impact of three system changes are investigated:

- **Removing the slip sheet placement.** Slip sheets are placed by the PR which is a system design decision. The PR is also responsible for placing the cases on the carrier. This makes placing a slip sheet by the PR a direct impact on the capacity.
- **Making the lift takeover height variable.** The lift on which the cases are placed performs a takeover to another lift at a certain height. As this height is fixed, it often happens between the placement of two arbitrary cases. The duration of the takeover is long enough to force the PR to wait for it. Due to this the PR cannot place the next case until the carrier lift finished the takeover and STack check Object (STO) performed its check.
- **Using only one tray lift.** A system design decision is to have two tray lifts that work alternately. By making a small change to the model, the influence of having one lift instead of two is investigated. Although the presumption is that the capacity will go down, the effect of this system change can be of interest. For example in application in which the high capacity of the system is not necessarily needed and there is still a wish to perform analysis on the systems' behaviour

Slip Sheet Placement

To get insight on the impact of removing slip sheet placements, simulations are performed with the model structure change. The simulation using the adjusted SSMPL model is performed 10 times. The resulting average capacity is equal to 757.3 cases per hour. This is an increase of 7.1 cases per hour compared to the original SSMPL model. This is a percentual increase of 0.95%.

Variable Lift Takeover Height

By making the lift takeover height variable, the logical choice is to make the height equal a height on which a case would be placed. By doing this, cases can be placed on the stack while the lift takeover takes place in parallel. This simulation is performed 10 times as well and results in an average capacity of 759.3 cases per hour. The extra capacity is equal to 9.1 cases per hour compared to the original SSMPL model. This is a percentual increase of 1.21%.

One Tray Lift

By changing the system from having two tray lifts into having one tray lift, the bottleneck shifts more towards the tray lift as the system becomes more reliable on it. Also the capacity drops down. By taking the average value of 10 simulation results the resulting capacity of the system with one lift is equal to 556.2 cases per hour. This is a loss of 194 cases per hour compared to the original SSMPL model. This is a percentual decrease of 25.85%.

5-4 Gaining Higher Capacity

An important objective of Vanderlande is to gain a higher capacity with the system. To determine on which component the focus should be in order to achieve this result, the performed analyses can be used.

The analysis on shifting bottlenecks, performed in Section 5-2-3, concluded on the TUR having the biggest sole bottleneck percentage. Furthermore, the critical circuits analysis showed that the most contribution to the critical circuit comes from actions performed by the TUR. In particular the movement of the TUR towards the PUT. This speaks in favour of improving the TUR movement from the tray towards the PUT. By adjusting the timing variables of the TUR, as shown in Table 5-5 in Section 5-3-1, the influence of speeding up the TUR is significant.

The analysis on the cycle times shows that the modes describing long events at the carrier lift have a longer cycle time. This results in a lower capacity, however those modes do not occur very often. Therefore the test performed in Section 5-3-2, in which these modes were removed, shows a capacity increase. These adjustment are making the lift takeover height variable and not placing slip sheets. The analysis on cycle times also showed the difference on cycle time by having one or multiple cases in a tray. Only having one case in the trays would result in a much higher capacity.

System design features give boundary conditions to choices. An important feature of Vanderlande's ACP system is the fact that multiple cases can be stored into one tray. A change of only allowing one case per tray would increase the capacity of the palletiser, but is not desirable. The presumption is that improving the path determined by Descrambling Logic Object (DLO) and looking at possible ways to increase the speed of moving from the tray to the PUT gives the biggest increase of capacity. An important note should be made that accuracy of the robots most likely decreases as their accelerations and speed increases. Furthermore, the mechanics of the TUR gives restriction on the possible paths that can be performed with certain acceleration and speed.

Many other changes to the system can be made in order to get a higher capacity. Some thoughts on possible changes with additional thought are stated here.

- **TUR stop determination**. Currently, the moment at which is determined whether the TUR will stop on its path towards the PUT is when the TUR approaches the case. This intermediate stop is needed when the PUT is blocked. However, the moment that the TUR enters the PUT area is later than the moment that the TUR approaches the case. This causes unnecessary stops that take more time than TUR movements without that stop. If the determination on whether the TUR has to stop is later, the presumption is that there will be less unnecessary stops and the capacity will increase.
- **Multiple PUTs.** As both the TUR and the PR have to wait in front of the PUT, the addition of an extra PUT could overcome this. Cases would be placed on both PUTs in alternating ways. The presumption is that this addition on the current system will cause less TUR stops towards the PUT.
- **Choice in PR paths.** The PR path from the PUT towards the carrier lift is predetermined and optimised by Load Forming Logic (LFL) and the order in which a case

sequence reaches the infeed. However, cases can also be placed differently on the PUT, turned 90 degrees. Whenever this would make the TUR path faster, the choice could be made to take a less optimised PR path. This approach could be taken broader such that the system could choice for longer PR paths at times that the carrier lift has to perform an action. In this manner, actions are performed more in parallel.

- **Optimization in number of cases in a tray.** The TUR path from tray to PUT is statistically longer if multiple cases are in a tray. In order to reduce the TUR time as much as possible, tray containing the exact number of cases needed has the preference. However, sometimes it is known beforehand that the TUR has to wait before it may enter the PUT. This happen at moments that the carrier lift has to perform an long lasting action. Control on the trays coming to the automatic palletiser could time the specific moments that the TUR takes a long path.

Chapter 6

Max-Plus Scaling Functions

Stochastic Max-Plus Linear (MPL) models often make use of stochastic Max-Plus Scaling (MPS) functions [22]. Modelling using MPS functions can give the same results as the Switching Stochastic Max-Plus Linear (SSMPL) model established in Chapter 4. Moreover, it can possibly do calculations faster and it can be used to approximate on the expected value of the resulting state. After implementation of the automatic palletiser in Section 6-2-4 the conclusion is drawn that this system is not suitable for the MPS function techniques. However, an addition is made on the known theory by giving an expression for a matrix multiplication of MPS functions, this is given in Section 6-1-2.

6-1 Theory

In this section explanation is given on MPS function theory and on tools used in the implementation of the automatic palletiser. This implementation is given in Section 6-2. The expected value of an MPS function is worked out, and a method to approximate this expected value is given.

6-1-1 Max-Plus Scaling Function

The MPS function $f : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}_{\max}$ is defined in Equation (6-1) as explained in [22].

$$f(w, e) = \max_j (\alpha_j + \beta_j w + \gamma_j e) \tag{6-1}$$

In this expression $\alpha \in \mathbb{R}_{\max}^{m \times 1}$, $\beta \in \mathbb{R}_{\max}^{m \times p}$ and $\gamma \in \mathbb{R}_{\max}^{m \times n}$ contain the scaling factors. The j in Equation (6-1) denotes the j'th row of that vector or matrix, so $j \in \{1, \ldots, m\}$. In MPL models, the vectors $w \in \mathcal{W} \subseteq \mathbb{R}_{\max}^{p \times 1}$ and $e \in \Omega \subseteq \mathbb{R}_{\max}^{n \times 1}$ respectively represent the non-stochastic system information and the uncertainty. Depending on the model, w usually contains the reference r, the state x and the input u and thus takes the form stated in (6-2).

$$w = \begin{bmatrix} r \\ x \\ u \end{bmatrix} \tag{6-2}$$

In the application of the MPL models, $f(w, e)$ can represent a state element. For this reason the example of $f(w, e)$ representing $x_2(k)$ is worked out. Here $x_2(k)$ is an element of $x(k)$ in the recurrence relation $x(k) = A \otimes x(k-1)$ with A as in (6-3). As this example does not have any reference or input variables, vector w only contains the state $x(k-1)$. The element $x_2(k)$ is equal to Equation (6-4).

$$A = \begin{bmatrix} \varepsilon & \tau_{21} & \varepsilon & \varepsilon \\ \varepsilon & \tau_{12} + \tau_{21} + e & \frac{1}{2}\tau_{34} & \varepsilon \\ \varepsilon & \tau_{12} + \frac{3}{2}\tau_{21} + e & \frac{1}{2}\tau_{21} + \frac{1}{2}\tau_{34} & \tau_{43} \\ \varepsilon & \tau_{12} + \frac{3}{2}\tau_{21} + \tau_{34} + e & \frac{1}{2}\tau_{21} + \frac{3}{2}\tau_{34} & \tau_{34} + \tau_{43} \end{bmatrix} \tag{6-3}$$

$$\begin{aligned} x_2(k) &= \Big(\varepsilon \otimes x_1(k-1)\Big) \oplus \Big((\tau_{12} + \tau_{21} + e) \otimes x_2(k-1)\Big) \oplus \ldots \\ &\quad \Big(\tfrac{1}{2}\tau_{34} \otimes x_3(k-1)\Big) \oplus \Big(\varepsilon \otimes x_4(k-1)\Big) \\ &= \max\Big(\varepsilon + x_1(k-1), \tau_{12} + \tau_{21} + e + x_2(k-1), \tfrac{1}{2}\tau_{34} + x_3(k-1), \varepsilon + x_4(k-1)\Big) \end{aligned} \tag{6-4}$$

Translation to the MPS function (6-1) makes the parameters equal to (6-5).

$$\alpha = \begin{bmatrix} \varepsilon \\ \tau_{12} + \tau_{21} \\ \frac{1}{2}\tau_{34} \\ \varepsilon \end{bmatrix}, \quad \beta = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \gamma = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{6-5}$$

6-1-2 Max-Plus Scaling Function in Matrix Multiplication

When simulating an MPL model, multiple A-matrices are max-plus multiplied in a sequence. This is also possible when a matrix in $\mathbb{R}_{\max}$ is expressed with MPS functions. However, the form of the scaling function is different to the form given in Equation (6-1).

Take matrix $A \in \mathbb{R}_{\max}^{I \times K}$ where every element a_{ik} is expressed as stated in Equation (6-6) with e_0 equal to $e_0 = 1$, A is then called an *MPS matrix.*

$$\begin{aligned} a_{ik} &= \max_l \left(\alpha_{ikl0} + \alpha_{ikl1} e_1 + \ldots + \alpha_{iklM} e_M\right) \\ &= \max_l \left(\sum_{m=0}^{M} \alpha_{iklm} e_m\right) \end{aligned} \tag{6-6}$$

The α-term and the γ-terms from Equation (6-1) are now all denoted by α-terms. Furthermore the β-terms which were present in Equation (6-1) are not in this equation as those terms are only of importance when the state, input or control variables are multiplied by the A-matrix.

In this situation the interest is only in the multiplication of several A-matrices. An explanation is given here with matrix A and matrix $B \in \mathbb{R}_{\max}^{K\times J}$ of which all elements b_{kj} are defined similar to the elements in matrix A as stated in (6-7).

$$b_{kj} = \max_{l'} \left(\sum_{m=0}^{M} \beta_{kjl'm} e_m \right) \tag{6-7}$$

There are several remarks that have to be made for clarification:

- The number of columns and the number of rows of respectively A and B are equal to K which makes the multiplication possible.
- For every a_{ik} and b_{kj} there are values L and L' such that $l \in \{1,\dots,L\}$ and $l' \in \{1,\dots,L'\}$. For every element a_{ik} the same value for L holds. In theory this does not have to be the case, this is however necessary for implementation in MATLAB with the use of 4-D doubles. The same holds for the value of L' for every element b_{kj}.
- By saying that all the error terms are put in one vector $\tilde{e} = \begin{bmatrix} 1 & e_1 & \cdots & e_M \end{bmatrix}^\top$, the same value for M can be used for every element a_{ik} and b_{kj}.

Having arbitrary matrices A and B as described, the derivation of obtaining $[A\otimes B]_{ij}$ is given in Equation (6-8). The product of A and B is denoted by H and the goal is to write the elements of H in the same manner as the elements of A and B.

$$\begin{aligned}
[H]_{ij} = [A \otimes B]_{ij} &= \max_k \left(\max_l \left(\sum_{m=0}^{M} \alpha_{iklm} e_m \right) + \max_{l'} \left(\sum_{m=0}^{M} \beta_{kjl'm} e_m \right) \right) \\
&= \max_k \left(\max_{l,l'} \left(\sum_{m=0}^{M} \alpha_{iklm} e_m + \sum_{m=0}^{M} \beta_{kjl'm} e_m \right) \right) \\
&= \max_k \left(\max_{l,l'} \left(\sum_{m=0}^{M} \left(\alpha_{iklm} + \beta_{kjl'm}\right) e_m \right) \right) \\
&= \max_{k,l,l'} \left(\sum_{m=0}^{M} \left(\alpha_{iklm} + \beta_{kjl'm}\right) e_m \right) \\
&= \max_n \left(\sum_{m=0}^{M} h_{ijnm} e_m \right)
\end{aligned} \tag{6-8}$$

In this final expression the $n \in \{1,\dots,K\cdot L\cdot L'\}$ combines the values k, l and l' and could be called the "l-term" for h_{ij} comparing it to the definitions of A and B in Equations (6-6) and (6-7). The way in which the element $[H]_{ij}$ is written is similar to Equation (6-6) which makes it possible to easily express sequences of MPS matrix multiplications.

6-1-3 Expected Value and Approximation

The expectation of state elements could be of interest as they can represent timing instant of internal system events. As the single state elements can be written as an MPS function as shown in Section 6-1-1, there is also a need in expressing the expectation of an MPS function. This is explained along the definitions of [23]. The MPS function given in (6-9) represents a single state element at a single event counter where $e(k)$ has probability density function $p(\cdot)$.

$$v(k) = \max_{j=1,\dots,m} (\alpha_j + \beta_j w(k) + \gamma_j e(k)) \tag{6-9}$$

Now the set Ω which describes the error terms is divided into m non-overlapping subsets Ω_j depending on the value of $e(k)$. The division is made in such a way that for given arbitrary $w(k) \in \mathcal{W}$ Equation (6-10) holds. The expectation of the MPS function is then equal to Equation (6-11).

$$\begin{aligned} & e(k) \in \Omega_j \quad \Rightarrow \quad v(k) = \alpha_j + \beta_j w(k) + \gamma_j e(k) \\ & \bigcup_{j=1}^{m} \Omega_j = \Omega \end{aligned} \tag{6-10}$$

$$\begin{aligned} \mathbb{E}[v(k)] &= \int_{\Omega} \cdots \int \max_j (\alpha_j + \beta_j w(k) + \gamma_j e(k)) p(e(k)) \mathrm{d}e_1(k) \cdots \mathrm{d}e_n(k) \\ &= \sum_{j=1}^{m} \int_{\Omega_j} \cdots \int (\alpha_j + \beta_j w(k) + \gamma_j e(k)) p(e(k)) \mathrm{d}e_1(k) \cdots \mathrm{d}e_n(k) \end{aligned} \tag{6-11}$$

Note that an MPS function as in Equation (6-9) is a maximum function of a number of affine expressions. Indeed, for every j the expression is affine as it is a sum of products of constants. In [24] an approximation method is introduced that is based on higher order moments, this work is used as guidance for the following explanation. The method is explained for an arbitrary vector $x \in \mathbb{R}^n$ of which the maximum is taken of the n elements of the vector. It holds that every element of x represents an affine expression. An upper bound for the expectation stated in (6-12) is the final result.

$$\mathbb{E}[\max(x_1, \dots, x_n)] \tag{6-12}$$

For the given vector x and $p \geq 1$, the p-norm and the ∞-norm are respectively defined as stated in (6-13) [25].

$$\begin{aligned} \|x\|_p &= (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}} \\ \|x\|_\infty &= \max(|x_1|, \dots, |x_n|) \end{aligned} \tag{6-13}$$

The p-norm and the ∞-norm of a vector meet the inequality stated in Inequality (6-14).

$$\|x\|_\infty \leq \|x\|_p \tag{6-14}$$

The same inequality holds for the expectation of these vectors, with the norms written out this result in Inequality (6-15).

$$\mathbb{E}[\max(|x_1|, \dots, |x_n|)] \leq \mathbb{E}\left[(|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}\right] \tag{6-15}$$

In [26] Jensen's inequality is stated which is used to give a relation between the expectation of certain functions. Let $\varphi(x)$ be a concave function defined by $\varphi : X \to Y$ and let $x \in X$ with probability one, Inequality (6-16) is the result.

$$\mathbb{E}\left[\varphi(x)\right] \leq \varphi\left(\mathbb{E}\left[x\right]\right) \tag{6-16}$$

With all the mathematical relations defined, some inequalities are stated that construct the first upper bound of the expectation in (6-12). The elements $x_j \in \mathbb{R}$ of x are considered random and $p > 1$. Inequality (6-17) is a straightforward one and is handled as trivial.

$$\mathbb{E}\left[\max\left(x_1, \ldots, x_n\right)\right] \leq \mathbb{E}\left[\max\left(|x_1|, \ldots, |x_n|\right)\right] \tag{6-17}$$

Secondly Inequality (6-15) is used to derive Inequality (6-18).

$$\mathbb{E}\left[\max\left(|x_1|, \ldots, |x_n|\right)\right] \leq \mathbb{E}\left[\left(|x_1|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}\right] \tag{6-18}$$

For $p > 1$ and $x > 0$ the function $\varphi(x) = x^{\frac{1}{p}}$ is concave which result with the use of Inequality (6-16) in Inequality (6-19).

$$\mathbb{E}\left[\left(|x_1|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}\right] \leq \mathbb{E}\left[\left(|x_1|^p + \cdots + |x_n|^p\right)\right]^{\frac{1}{p}} \tag{6-19}$$

Finally with the assumption that the elements of x are independent the expectation of the sum is rewritten as the sum of the expectations as shown in Equation (6-20).

$$\mathbb{E}\left[\left(|x_1|^p + \cdots + |x_n|^p\right)\right]^{\frac{1}{p}} = \left(\sum_{j=1}^{n} \mathbb{E}\left[|x_j|^p\right]\right)^{\frac{1}{p}} \tag{6-20}$$

Now the combination of (6-17)-(6-20) and the given conditions and assumptions give the upper bound stated in Inequality (6-21).

$$\mathbb{E}\left[\max\left(x_1, \ldots, x_n\right)\right] \leq \left(\sum_{j=1}^{n} \mathbb{E}\left[|x_j|^p\right]\right)^{\frac{1}{p}} \tag{6-21}$$

As in applications the state of MPL systems usually indicate time instants of event occurrences, the elements x_j are positive. In that case [24] remarks that a different upper bound than in (6-21) can be constructed with a smaller error. In that situation S is defined with the condition that $S \leq x_j$ for all j and p is chosen to be a positive and even integer. Then the approximation of $\mathbb{E}\left[\max\left(x_1, \ldots, x_n\right)\right]$ is given by the upper bound given in Equation (6-22).

$$\mathfrak{U}\left(\mathbb{E}\left[\max\left(x_1, \ldots, x_n\right)\right]\right) = \left(\sum_{j=1}^{n} \mathbb{E}\left[(x_j - S)^p\right]\right)^{\frac{1}{p}} + S \tag{6-22}$$

6-2 Implementation

The theory explained in Sections 6-1-1 and 6-1-2 is applied to an example in this section. In this example, two types of reduction are applied in order to reduce the complexity. After finishing the example, the same procedure is applied to a simplified model of the Automated Case Picking (ACP) automatic palletiser system.

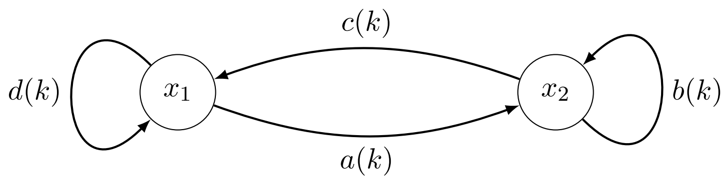


Figure 6-1: Topological graph of example system

6-2-1 Example

As an example we look at the following simple situation given by the system which is presented graphically in Figure 6-1. The mathematical equations that correspond to this are given in Equation (6-23) with the parameters specified in (6-24). In this example the power is taken of the A-matrix corresponding to this system. The A-matrix that is derived from this system is equal to Equation (6-25).

$$\begin{cases} x_1(k) = \max\left(x_1(k-1) + d(k), x_2(k-1) + c(k)\right) \\ x_2(k) = \max\left(x_1(k) + a(k), x_2(k-1) + b(k)\right) \end{cases}$$
$$\Leftrightarrow$$
$$\begin{cases} x_1(k) = \max\left(x_1(k-1) + d(k), x_2(k-1) + c(k)\right) \\ x_2(k) = \max\left(x_1(k-1) + a(k) + d(k), x_2(k-1) + a(k) + c(k), x_2(k-1) + b(k)\right) \end{cases} \tag{6-23}$$

$$\begin{cases} a(k) = a_0 + a_1 e_1(k) \\ b(k) = b_0 + b_1 e_2(k) \\ c(k) = c_0 + c_1 e_3(k) \\ d(k) = d_0 + d_1 e_4(k) \end{cases} \tag{6-24}$$

$$\begin{aligned} A(k) &= \begin{bmatrix} d(k) & c(k) \\ a(k) + d(k) & \max\left(a(k) + c(k), b(k)\right) \end{bmatrix} \\ &= \begin{bmatrix} d_0 + d_1 e_4(k) & c_0 + c_1 e_3(k) \\ a_0 + d_0 + a_1 e_1(k) + d_1 e_4(k) & \max\left(a_0 + c_0 + a_1 e_1(k) + c_1 e_3(k), b_0 + b_1 e_2(k)\right) \end{bmatrix} \end{aligned} \tag{6-25}$$

The resulting dimension of the MPS A-matrix depends on the following 3 things:

- The number of independent error terms that are needed within the multiplication.
- The maximum number of terms in a maximum operator within the A-matrix.
- The dimension of the A-matrix.

In this example A to the power 2 is calculated and for every $A(k)$ there is need for 4 error terms. Therefore the length of vector $\tilde{e}$ is equal to 9, 2 times 4 error terms and an additional 1 at top of the vector. As a result the inner matrices of MPS(A) have 9 columns.

The maximum number of terms in a maximum operator in Equation (6-25) is 2. This makes the size of the inner matrices equal to 2×9.

The dimension of the A-matrix is equal to 2×2. The resulting dimension of the MPS A-matrix is equal to $2 \times 9 \times 2 \times 2$.

Having an MPS matrix A, the notation for the matrix in which the inner matrices are written out is MPS(A). The MPS $A(k)$-matrix corresponding to $A(k)$ given in Equation (6-25) is equal to Equation (6-26) with $\tilde{e}$ equal to (6-27).

$$\mathrm{MPS}(A(k)) = \begin{bmatrix} \begin{bmatrix} d_0 & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} c_0 & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} \\ \begin{bmatrix} a_0 + d_0 & a_1 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} a_0 + c_0 & a_1 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \\ b_0 & 0 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} \end{bmatrix} \tag{6-26}$$

$$\tilde{e} = \begin{bmatrix} 1 \\ e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \\ e_1(k+1) \\ e_2(k+1) \\ e_3(k+1) \\ e_4(k+1) \end{bmatrix} \tag{6-27}$$

Consequently, the MPS $A(k+1)$-matrix has a similar structure and is shown in Equation (6-28). However the scaling parameters of the error terms are now multiplied to other terms of $\tilde{e}$.

$$\mathrm{MPS}(A(k+1)) = \begin{bmatrix} \begin{bmatrix} d_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} c_0 & 0 & 0 & 0 & 0 & 0 & 0 & c_1 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} \\ \begin{bmatrix} a_0 + d_0 & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} a_0 + c_0 & 0 & 0 & 0 & 0 & a_1 & 0 & c_1 & 0 \\ b_0 & 0 & 0 & 0 & 0 & 0 & b_1 & 0 & 0 \end{bmatrix}\tilde{e} \end{bmatrix} \tag{6-28}$$

Now the MPS matrix of the product of $A(k+1) \otimes A(k)$ is calculated. As shown in Equation (6-8) the resulting inner matrices within the MPS matrix contain all the combinations of the matrices that are being multiplied. For this example the result of the multiplication is stated in Equation (6-29).

$$
\mathrm{MPS}(A(k+1)\otimes A(k)) = \begin{bmatrix} \begin{bmatrix} 2d_0 & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_0+c_0+d_0 & a_1 & 0 & 0 & d_1 & 0 & 0 & c_1 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & c_1 & 0 \\ \varepsilon & a_1 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} c_0+d_0 & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_0+2c_0 & a_1 & 0 & c_1 & 0 & 0 & 0 & c_1 & 0 \\ b_0+c_0 & 0 & b_1 & 0 & 0 & 0 & 0 & c_1 & 0 \\ \varepsilon & a_1 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & b_1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} a_0+2d_0 & 0 & 0 & 0 & d_1 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2a_0+c_0+d_0 & a_1 & 0 & 0 & d_1 & a_1 & 0 & c_1 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+d_0 & a_1 & 0 & 0 & d_1 & 0 & b_1 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & b_1 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} a_0+c_0+d_0 & 0 & 0 & c_1 & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2a_0+2c_0 & a_1 & 0 & c_1 & 0 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+c_0 & 0 & b_1 & 0 & 0 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+c_0 & a_1 & 0 & c_1 & 0 & 0 & b_1 & 0 & 0 \\ 2b_0 & 0 & b_1 & 0 & 0 & 0 & b_1 & 0 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-29}
$$

6-2-2 Simple Reduction

The MPS matrix in Equation (6-29) is used to obtain the answer to $A(k+1)\otimes A(k)$. For every element within the overall matrix, the maximum is taken from the different rows of the corresponding inner matrix. It is easy to see that the rows that have an $\varepsilon(=-\infty)$ as first element never give the maximum. Therefore a reduction can be made on the result. Only the number of rows of the inner matrices can be reduced as all columns are needed for possible maxima.

The resulting matrix as in Equation (6-29) can be transformed by reordering the rows of the inner matrices. This change does not influence the value of the A-matrix as the order within a maximum operator does not matter for the answer. The reordering puts all the rows of which the first element is equal to ε to the lower rows. After that, the least number of ε-rows in the inner matrices is determined. For (6-29) this is equal to 3, that number of rows are deleted from the inner matrices.

After the described reduction the resulting MPS matrix of the example takes the form given in Equation (6-30).

$$
\mathrm{MPS}(A(k+1)\otimes A(k)) \overset{\text{reduced}}{=} \begin{bmatrix} \begin{bmatrix} 2d_0 & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & d_1 \\ a_0+c_0+d_0 & a_1 & 0 & 0 & d_1 & 0 & 0 & c_1 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} c_0+d_0 & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & d_1 \\ a_0+2c_0 & a_1 & 0 & c_1 & 0 & 0 & 0 & c_1 & 0 \\ b_0+c_0 & 0 & b_1 & 0 & 0 & 0 & 0 & c_1 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & c_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} a_0+2d_0 & 0 & 0 & 0 & d_1 & a_1 & 0 & 0 & d_1 \\ 2a_0+c_0+d_0 & a_1 & 0 & 0 & d_1 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+d_0 & a_1 & 0 & 0 & d_1 & 0 & b_1 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & 0 & 0 & d_1 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} a_0+c_0+d_0 & 0 & 0 & c_1 & 0 & a_1 & 0 & 0 & d_1 \\ 2a_0+2c_0 & a_1 & 0 & c_1 & 0 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+c_0 & 0 & b_1 & 0 & 0 & a_1 & 0 & c_1 & 0 \\ a_0+b_0+c_0 & a_1 & 0 & c_1 & 0 & 0 & b_1 & 0 & 0 \\ 2b_0 & 0 & b_1 & 0 & 0 & 0 & b_1 & 0 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-30}
$$

6-2-3 Sorted Reduction

Unimportant information is deleted with the simple reduction explained in Section 6-2-2. However, only one row of the inner matrices of an MPS matrix contains the information that is needed. This is the row that, given the errors, causes the maximum. Beforehand, the row that gives this maximum is not known. There is a likelihood per row of giving the maximum nonetheless.

This likelihood is not easy to determine. First of all, the rows are corresponding to the sum of many random variables. In theory the restriction can be given that these error terms are identically distributed, however this is usually not the case in practise. In [27] random walks are discussed. Random walks can be compared to sums of random variables. It is stated there that, as expected, the standard deviation of the sum of random variables grows as random values are added. This means that summing many random variables gives more uncertainty on the resulting value. Consequently the likelihood of a row to give the maximum is more uncertain.

Furthermore, the different rows within an inner matrix are not independent. This is because of the fact that error terms affect more than one row. Because of this dependency it is not possible to compare confidence intervals of the resulting distributions. Although it is mathematically possible to determine a likelihood for every row to be the maximum, this takes a lot of calculation time. Another approach could be calculating the resulting distribution of the resulting maximum caused by on the different rows, as discussed in [28]. However this is not considered in this study as the focus is on obtaining the real solution to the matrix multiplication.

In this study there has only been reduction on the “ground variable”. This is the first element of a row in the inner matrices. As the error terms are scaled and added to these ground terms, the ground terms are likely to be near the expected value. In the case of normally distributed errors with mean equal to zero, the ground terms are equal to the expected value.

This way of reduction is shown here applied on the example of Equation (6-30). For this example the unknowns are chosen as stated in (6-31).

$$\begin{aligned} a_0 &= 3, & a_1 &= 1 \\ b_0 &= 6, & b_1 &= 2 \\ c_0 &= 9, & c_1 &= 3 \\ d_0 &= 12, & d_1 &= 4 \end{aligned} \tag{6-31}$$

When these values are used in the example, the matrix stated in (6-32) is the result

$$\mathrm{MPS}(A(k+1) \otimes A(k)) \overset{\text{reduced}}{=} \begin{bmatrix} \begin{bmatrix} 24 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 24 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ \varepsilon & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 21 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \\ 21 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 \\ 15 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ \varepsilon & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} 27 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 \\ 27 & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 \\ 21 & 1 & 0 & 0 & 4 & 0 & 2 & 0 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \\ \varepsilon & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 24 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 \\ 24 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 \\ 18 & 0 & 2 & 0 & 0 & 1 & 0 & 3 & 0 \\ 18 & 1 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\ 12 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-32}$$

For this example the reduction number is equal to 2 which means that after each iteration only 2 rows are kept for the next calculation. This result in Equation (6-33).

$$\mathrm{MPS}(A(k+1)\otimes A(k)) \overset{\text{reduced}}{=} \begin{bmatrix} \begin{bmatrix} 24 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 24 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 21 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \\ 21 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} 27 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 \\ 27 & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 24 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 \\ 24 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-33}$$

The next iteration is performed by max-plus multiplication with $A(k+2)$. The MPS matrix of $A(k+2)$ is given by (6-34) in which $\tilde{e}$ is extended to (6-35).

$$\mathrm{MPS}(A(k+2)) = \begin{bmatrix} \begin{bmatrix} d_0 & 0 & \cdots & 0 & 0 & 0 & 0 & d_1 \\ \varepsilon & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} c_0 & 0 & \cdots & 0 & 0 & 0 & c_1 & 0 \\ \varepsilon & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} a_0 + d_0 & 0 & \cdots & 0 & a_1 & 0 & 0 & d_1 \\ \varepsilon & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} a_0 + c_0 & 0 & \cdots & 0 & a_1 & 0 & c_1 & 0 \\ b_0 & 0 & \cdots & 0 & 0 & b_1 & 0 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-34}$$

$$\tilde{e} = \begin{bmatrix} 1 \\ e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \\ e_1(k+1) \\ e_2(k+1) \\ e_3(k+1) \\ e_4(k+1) \\ e_1(k+2) \\ e_2(k+2) \\ e_3(k+2) \\ e_4(k+2) \end{bmatrix} \tag{6-35}$$

Due to the extra step, the error vector $\tilde{e}$ is larger now which makes the inner matrices of the MPS matrix larger as well. All the inner matrices get 4 extra columns.

The MPS $A(k+2)$-matrix is multiplied to the reduced MPS matrix of the product of $A(k+1)$ and $A(k)$. The resulting reduced MPS matrix is given by (6-36).

$$\mathrm{MPS}(A(k+2)\otimes\ldots\otimes A(k)) \overset{\text{reduced}}{=} \begin{bmatrix} \begin{bmatrix} 24+d_0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & d_1 \\ 24+d_0 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & d_1 \\ 24+\varepsilon & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 24+\varepsilon & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 27+c_0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 & 0 & 0 & c_1 & 0 \\ 27+c_0 & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 & 0 & 0 & c_1 & 0 \\ 27+\varepsilon & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 27+\varepsilon & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 21+d_0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & d_1 \\ 21+d_0 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & d_1 \\ 21+\varepsilon & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 21+\varepsilon & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 24+c_0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 & 0 & 0 & c_1 & 0 \\ 24+c_0 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 & 0 & 0 & c_1 & 0 \\ 24+\varepsilon & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 24+\varepsilon & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tilde{e} \\ \begin{bmatrix} 24+a_0+d_0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & a_0 & 0 & 0 & d_1 \\ 24+a_0+d_0 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & a_0 & 0 & 0 & d_1 \\ 24+\varepsilon & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 24+\varepsilon & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 27+a_0+c_0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 & a_0 & 0 & c_1 & 0 \\ 27+a_0+c_0 & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 & a_0 & 0 & c_1 & 0 \\ 27+b_0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 & 0 & b_0 & 0 & 0 \\ 27+b_0 & 1 & 0 & 0 & 4 & 1 & 0 & 3 & 0 & 0 & b_0 & 0 & 0 \end{bmatrix} \tilde{e} & \begin{bmatrix} 21+a_0+d_0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & a_0 & 0 & 0 & d_1 \\ 21+a_0+d_0 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & a_0 & 0 & 0 & d_1 \\ 21+\varepsilon & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 21+\varepsilon & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 24+a_0+c_0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 & a_0 & 0 & c_1 & 0 \\ 24+a_0+c_0 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 & a_0 & 0 & c_1 & 0 \\ 24+b_0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 & 0 & b_0 & 0 & 0 \\ 24+b_0 & 1 & 0 & 3 & 0 & 1 & 0 & 3 & 0 & 0 & b_0 & 0 & 0 \end{bmatrix} \tilde{e} \end{bmatrix} \tag{6-36}$$

The obtained MPS matrix is not fully reduced yet. Both explained reductions are used. First the unnecessary rows with $-\infty$ as first element are deleted. Then the parameters are filled in with the chosen values and rows are deleted after being sorted. This results in the reduced MPS matrix given by (6-37).

$$\mathrm{MPS}(A(k+2)\otimes\ldots\otimes A(k)) \overset{\text{reduced}}{=} \begin{bmatrix} \begin{bmatrix} 36 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 36 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}\tilde{e} & \begin{bmatrix} 33 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 \\ 33 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}\tilde{e} \\ \begin{bmatrix} 39 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 \\ 39 & 1 & 0 & 0 & 4 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 \end{bmatrix}\tilde{e} & \begin{bmatrix} 36 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 4 & 1 & 0 & 0 & 4 \\ 36 & 1 & 0 & 3 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 0 & 4 \end{bmatrix}\tilde{e} \end{bmatrix} \tag{6-37}$$

In the second reduction step, there had to be chosen between rows with the same ground value. For two normal distribution with the same mean value, the chance of one value being bigger than the other is 50%. For this reason, the upper two rows of these rows are kept in the reduced matrix as there is no preference.

6-2-4 Palletiser Implementation

There is a big chance that the rows that would give the maximum in the example of Section 6-2-3 was deleted. However taking just two rows to keep after each iteration is very little, it was just for the sake of the example. Now the automatic palletiser is implemented into this MPS function calculation. As there were only two events in the previous example, also only two events are taken into account in this implementation:

1. $x_1(k)$ is the moment that the Tray Unloading Robot (TUR) gets the trigger that it may approach the tray lift.
2. $x_2(k)$ is the moment that the Palletiser Robot (PR) gets the trigger that it may approach the PickUp Table (PUT).

The reason for choosing those events is that those moments are important and determining synchronisations in the system.

The variables a, b, c, d are chosen to be normally distributed with values given in (6-38).

$$\begin{array}{lcll} a(k) \sim \mathcal{N}(3.99, 0.45^2) & \Rightarrow & a_0 = 3.99, & a_1 = 0.45 \\ b(k) \sim \mathcal{N}(4.42, 0.36^2) & \Rightarrow & b_0 = 4.42, & b_1 = 0.36 \\ c(k) \sim \mathcal{N}(0.33, 0.13^2) & \Rightarrow & c_0 = 0.33, & c_1 = 0.13 \\ d(k) \sim \mathcal{N}(2.17, 0.1^2) & \Rightarrow & d_0 = 2.17, & d_1 = 0.1 \end{array} \tag{6-38}$$

The MPS A-matrix corresponding to this example is given by Equation (6-39).

$$\mathrm{MPS}(A(k)) = \begin{bmatrix} \begin{bmatrix} 2.17 & 0 & 0 & 0 & 0.1 \\ \varepsilon & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} 0.33 & 0 & 0 & 0.13 & 0 \\ \varepsilon & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} \\ \begin{bmatrix} 6.16 & 0.45 & 0 & 0 & 0.1 \\ \varepsilon & 0 & 0 & 0 & 0 \end{bmatrix}\tilde{e} & \begin{bmatrix} 4.32 & 0.45 & 0 & 0.13 & 0 \\ 4.42 & 0 & 0.36 & 0 & 0 \end{bmatrix}\tilde{e} \end{bmatrix} \tag{6-39}$$

With this matrix, analysis is done on accuracy and speed of calculating a max-plus product sequence of these A-matrices. In this analysis two parameters are taken variable. The first variable parameter is the number of A-matrices that are multiplied (number of iterations).

Table 6-1: Results and durations of calculations of the MPS function implementation to the palletiser example

Number of iterations i	Number of rows after reduction r	$\delta(A, i, r)$	Calculation time [sec]	Approximation time 1 [sec]	Approximation time 2 [sec]
10	100	0.0122	0.0042	0.0326	0.0011
10	1000	0.0001	0.0042	0.1376	0.0010
10	10000	0	0.0042	0.2309	0.0093
10	100000	0	0.0042	0.2179	0.0133
50	100	0.0417	0.0013	0.3946	0.0003
50	1000	0.0387	0.0013	4.3092	0.0048
50	10000	0.0352	0.0013	42.1532	0.0584
50	100000	0.032	0.0013	589.7985	0.5201
100	100	0.0452	0.0108	1.7716	0.0006
100	1000	0.0429	0.0108	17.0920	0.0098
100	10000	0.0424	0.0108	160.3838	0.098
100	100000	0.0403	0.0108	2633.7	0.9795

The second variable parameter is the number of inner matrix rows that are kept after the reductions.

Take the example where 100 iterations are taken and the reduction reduces the number of rows to 1000. The multiplication that is being approximated then is equal to is $A[k+99,k] = A(k+99) \otimes A(k+98) \otimes \cdots \otimes A(k+1) \otimes A(k)$. The approximation with 1000 rows after reduction is denoted by $A_{\text{approx}(1000)}[k+99,k]$. The dimensions of the MPS matrices corresponding to these 100 A-matrices are of the dimension $1000 \times 401 \times 2 \times 2$. The accuracy δ of this situation is calculated by dividing the difference between the real solution $A[k+99,k]$ and the approximation $A_{\text{approx}(1000)}[k+99,k]$ by the real value $A[k+99,k]$ as shown in Equation (6-40). In order to do this, the sum of all the elements in the matrix is taken, in this way the division is possible. The sum of all elements on a matrix is denoted by $\rho(A)$.

$$\delta(A, 100, 1000) = \frac{\rho\big(A[k+99,k] - A_{\text{approx}(1000)}[k+99,k]\big)}{\rho\big(A[k+99,k]\big)} \tag{6-40}$$

For the given values of the system and for different number of iterations and reduction rows, these calculations are performed using MATLAB and put in Table 6-1. Also the time it takes to do the calculations is tracked. First of the all the real matrix multiplication is timed which is called the calculations time. The approximation time is split into two parts. The first, approximation time 1, concerns the calculation of the MPS matrix that corresponds to the result of the multiplication sequence. The second, approximation time 2, concerns the calculation of the final approximation of the resulting matrix by calculating the maxima of the different element for a certain perturbation vector $\tilde{e}$. The reason for splitting these two approximation times is because of the fact that the calculation of the approximation of the multiplication sequence has to be done just once when performing simulations. When this is done, calculations for different situations can be done by only calculating the result of a certain perturbation vector combined with the resulting MPS matrix.

As random variables are used, the resulting values change when performing the experiment again. However, every performed similar experiment gives values that are comparable to the values in Table 6-1.

6-3 Complexity and Accuracy

When p A-matrices are multiplied in a sequence, the multiplication looks like $A(k+p-1) \otimes A(k+p-2) \otimes \ldots \otimes A(k+1) \otimes A(k)$. The number of columns of the inner matrices of the resulting MPS matrix is depending on the number of error terms used and the number of matrices that are in the multiplication sequence. In the example in Section 6-2-1 there are 4 error terms and p matrices in the multiplication, the resulting number of columns of the inner matrices is $4p+1$. If n_e is the number of error terms needed for an A-matrix, the resulting number of columns in equal to $n_e p + 1$.

The number of rows of the resulting inner matrices is depending on the dimension of the A-matrix and on the value of L (for implementation into MATLAB this is taken equal to the maximum value of L). In the example in Section 6-2-1 the dimensions of the A matrix is 2×2, so K is equal to 2. The value of L is also equal to 2. The number of rows of the resulting matrices is then equal to $K^{p-1}L^p$.

The dimension of the MPS matrix of the result of the multiplication $A(k+p-1) \otimes A(k+p-2) \otimes \ldots \otimes A(k+1) \otimes A(k)$ is equal to $(K^{p-1}L^p) \times (n_e p+1) \times 2 \times 2$. The assumption that L is equal for all elements is partly determinative here.

As the number of rows of the inner matrices grows rapidly in applications, rows are deleted according to a sorting procedure. However, by deleting rows, the possible maximum could be deleted, depending on the risk that is taken at deletion. Due to this the accuracy of the resulting A-matrix is worse than by performing the matrix multiplication sequence.

For the implementation performed in Section 6-2-4 the δ-term shows that the correct A-matrix is only obtained when few matrices are multiplied and few reduction is applied. For realistic simulation with fast calculation times many matrices need to be multiplied and significant reduction should be applied. The presumption of the moderate added value of using the MPS matrix multiplication in the automatic palletiser implementation is that the determining cycle times in the model are too close to each other. Systems that contain a component that is dominant in the determination of the cycle time of the system are more suitable for this method.

Chapter 7

Conclusions and Recommendations

This final chapter concludes the master thesis. Answers are provided for the main research questions and the two sub-questions. The main research question is:

Is it possible to model and analyse the ACP automatic palletiser with the use of max-plus algebra?

To answer this question, two sub-questions were formulated:

1. *Can Vanderlande's Automated Case Picking (ACP) automatic palletiser be modelled correctly using a Switching Stochastic Max-Plus Linear (SSMPL) model?*
2. *After establishing an SSMPL model, what can be said on capacity, bottleneck and sensitivity of the ACP automatic palletiser?*

Furthermore, all other conclusions that are presented throughout the thesis are stated here. Finally, some recommendations are given for future research, that is similar to or succeeding this thesis.

7-1 Conclusions

The answer to the first sub-question is yes, but the following additional statements have to be kept in mind:

- The resulting model is made under the assumptions that:
  - There is infinite tray infeed
  - There are no obstruction at the outfeed
  - The signal messaging does not take time
- The obtained capacity calculated by the SSMPL model is structural lower than the reference capacity. This difference is approximately 1.7%.

The established model is used to answer the second sub-question. As already concluded in the previous sub-question, the capacity obtained with the model is slightly lower than the reference capacity. Despite this difference, the corrected reference capacity is taken as true capacity. The correction on the reference capacity is due to a mistake made on the determination of the waiting times of the Palletiser Robot (PR). The result is a system capacity of approximately 771 cases per hour, while the reference capacity stated approximately 763 cases per hour. The Tray Unloading Robot (TUR) is pointed out as most determining bottleneck of the system. Due to the dynamic behaviour of the system, the bottleneck role is divided among the different component of the system. The degree of “bottleneckness” is reflected in the system’s sensitiveness. This results in the biggest effect of adjusting TUR timing variables followed by the PR timing variables.

More (detailed) conclusions are drawn on the subjects modelling, analyses and on Max-Plus Scaling (MPS).

7-1-1 Modelling

Working towards the resulting SSMPL model, several conclusions are drawn that are listed here.

- The structure of the Switching Max-Plus Linear (SMPL) model constructed in Section 4-1 is correct. Validation of the SMPL model to the reference simulation performed at Vanderlande resulted in a difference of approximately1.2%. After analysis, the difference in capacity was due to a slight mistake in the capacity determination. Validation of the SMPL model with exclusion of the part where the mistake is made, results in a difference in capacity of approximately 0.05%.
- Modelling with the use of an SMPL model is concluded to be constructive and relatively easy. This conclusion is made as the established model gave good results without having to perform difficult or complex calculations. By only mapping all the synchronisations, modes and durations, simulation can be performed.
- SSMPL models are very sensitive for how timing variables that link internal events are distributed. This conclusions is drawn on basis of tests with the three distributions, the modified PERT distribution, the Weibull distribution and the normal distribution with a minimum. Parameter adjustments to the distributions had significant influence on the resulting capacity. For this reason, the conclusion is that these distribution are not suitable to use for simulation purposes of the automatic palletiser with the use of an SSMPL model.
- The capacity obtained by the resulting SSMPL model is consequent lower than the reference capacity, approximately 1.7%. This is linked to the higher process time in which the SSMPL model results. The presumed cause for this phenomenon is the combination of stochastic parameters and not taking into account al the correlations present in the system.
- When the datasets used for identification and verification are different, the SSMPL works similar to the situation in which the same datasets are used. This is concluded

with the use of a cross-validation test. The advantage of this using separate identification and verification sets is that different Load Forming Logic (LFL) recipes can be simulated with a model based on one PR Cycle Time Tool simulation.

7-1-2 Analyses

The SSMPL model of the automatic palletiser is used to do analyses on bottleneck and sensitivity issues. Several conclusions are drawn based on these analyses and listed here.

- The performed critical circuit analysis concludes that the TUR movement towards the PickUp Table (PUT) is the most determining action for the total process time of the simulation. This means that this specific timing variable is part of the maximum value inside a maximum operator most often. The linked conclusion can be made that this means that other components than the TUR have to wait for the TUR moving towards the PUT most often.

- The analysis done on the cycle times concludes that placing a slip sheet causes the longest average cycle time. This means that placing a slip sheet has the largest effect on the capacity. Another conclusion drawn in this analysis is that the cycle times concerning cases taken from a tray containing multiple cases are longer than single cases.

- Bottleneck analysis on basis of active durations performed concludes on the TUR being the dominant bottleneck. Both the sole and shifting bottleneck percentages are highest for the TUR with respectively percentages of approximately 57% and 24%. The PR is classified as the second bottleneck of the total system.

- The sensitivity of the resulting capacity as a result of adjusting the timing variables is analysed. Adjustments to timing variables related to the TUR created the differences in capacity. By adjusting the TUR timing variables to very low values, the capacity raised to a limit just below 800 cases per hour. The highest rate of profit is reached a TUR timings ratio of approximately 0.81

- Capacity changes caused structure changes of the system are analysed. Removing the possibility of placing slip sheets results in a capacity increase of 7.1 cases per hour (0.9%). Making the lift takeover height variable results in a capacity increase of 9.1 cases per hour (1.2%). Changing the number of tray lifts from two to one results in a capacity decrease of 194 cases per hour (25.9%).

7-1-3 Max-Plus Scaling

In order to make use of MPS functions in an SSMPL model, an easy way to write the MPS matrix multiplication is desirable. This is worked out and the conclusion is drawn that matrices of correct size and of the form given in Equation 7-1 can be multiplied. The result of the max-plus multiplication can be written in the same notation.

$$
\begin{aligned}
A &\in \mathbb{R}_{\max}^{I \times K} \\
[A]_{ik} &= a_{ik} \\
&= \max_{l} \left(\alpha_{ikl0} + \alpha_{ikl1} e_1 + \ldots + \alpha_{iklM} e_M \right) \\
&= \max_{l} \left(\sum_{m=0}^{M} \alpha_{iklm} e_m \right)
\end{aligned}
\tag{7-1}
$$

Modelling with the use of MPS functions did not work for the automatic palletiser system. The main reason for that is that the difference in cycle times of the TUR and the PR is not large enough. This results in the need of high computational complexity in order to get correct results.

7-2 Recommendations

Recommendations are put forward in this section as more research can gain interesting results. Furthermore recommendations are stated concerning Vanderlande's automatic palletiser system.

Correlations in Switching Stochastic Max-Plus Linear Systems

As presumed in Section 4-2-2, the lack of correlations probably results in a higher process time than the reference process time. More research on this is desirable in order to conclude on this. Other research can be done on structural differences in process time due to the lack of correlations. When this difference is proven to be equal for large enough simulations, the difference in modelling results and reality is not a problem. As an SSMPL system can be seen as a large network of connections, one can express it as many combined maximum operations containing random variables. Research on random variables combined with maximum operators in large systems can thus be added.

Max-Plus Scaling Functions for Switching Stochastic Max-Plus Linear Models

Modelling with the use of MPS function can give serious advantages. The automatic palletiser system was not suitable for being modelled like this due to fast expanding computational complexity. However, other systems, or other implementation can possibly make modelling with the use of MPS functions applicable. More research on computation complexity related to MPS matrix multiplication is desirable. Conclusions on this research can determine which conditions are needed to make a usable SSMPL model with the use of MPS functions.

Random Switching for the Automatic Palletiser

Random switching is way of modelling that is not researched in this thesis. However, it is recommended to do additional research on this as it can answer some interesting questions on the system. An example of analysis that can be done with a Random Switching Max-Plus Linear (RSMPL) model is on upper bounds of process times. Due to the maximum operators in combination with stochastic scenario changes, confidence intervals can be given on worst case scenario's. Not only the automatic palletiser but potentially other systems are suitable for this modelling. Other systems than the palletiser that describe parallel actions that are synchronised can be analysed like this.

Capacity of the Automatic Palletiser

For improving the capacity of the automatic palletiser it is recommended to put the main focus on the TUR. The analyses performed in this thesis point out that the TUR movement from the tray towards the PUT is dominant in the determination of most cycle times. However, accuracy should be kept in mind by making the movement of the TUR faster. Therefore it is recommended to improve the circumstance of the moving TUR instead instead of making the robot faster.

Other changes to the system that could improve the capacity are:

- Making the lift takeover height variable.
- Placing slip sheets by another robot than the PR.
- Determine whether the TUR has to stop in its way towards the PUT later than currently done.
- Changing the system by making two PUTs.
- Making the PR paths determining on the order in which trays come into the system. Also the number of cases in a tray can be taken into account in this situation.
- Making number of cases in the tray that is ordered from the warehouse depending on the LFL recipe.

Appendix A

Modelling Documentation

A-1 Events

Table A-1: List of event descriptions

State element (Event)	Description
$x_1(k)$ / $x_6(k)$	The moment that the tray containing the k'th case arrives at the height where the Tray Pattern detection Object (TPO) photo is taken.
$x_2(k)$ / $x_7(k)$	The moment that the TPO photo is made for the k'th case.
$x_3(k)$ / $x_8(k)$	The moment that the tray containing the k'th case leaves the height where the TPO photo is taken.
$x_4(k)$ / $x_9(k)$	The moment that the tray containing the k'th case arrives at the upper height.
$x_5(k)$ / $x_{10}(k)$	The moment that TPO and Descrambling Logic Object (DLO) finished their calculations for the k'th case.
$x_{11}(k)$	The moment that the Tray Unloading Robot (TUR) arrives at the TUR stop (a) while going to the k'th case located at the tray lift.
$x_{12}(k)$	The moment that the TUR leaves the TUR stop (a) while going to the k'th case located at the tray lift. When the TUR does not have to stop, this is equal to $x_{11}(k)$.
$x_{13}(k)$	The moment that the TUR starts pushing the k'th case.
$x_{14}(k)$ / $x_{16}(k)$	The moment that the TUR pushed the k'th case just off the tray.

$x_{15}(k)$ / $x_{17}(k)$	The moment that the TUR moved just out of the TPO camera vision while pushing the k'th case.
$x_{18}(k)$	The moment that the TUR arrives at the TUR stop (b) while pushing the k'th case towards the PickUp Table (PUT).
$x_{19}(k)$	The moment that the TUR leaves the TUR stop (b) while it is pushing the k'th case towards the PUT. When the TUR does not have to stop, this is equal to $x_{18}(k)$.
$x_{20}(k)$	The moment that the TUR placed the k'th case on the PUT.
$x_{21}(k)$	The moment that the TUR is at the position where it is not blocking the Palletiser Robot (PR) from the PUT area any more after placing the k'th case on the PUT.
$x_{22}(k)$	The moment that the PUT floor arrives at the low position after the k'th case is placed on the PUT.
$x_{23}(k)$	The moment that the PR arrives at the PR stop (a) while going to the k'th case.
$x_{24}(k)$	The moment that the PR leaves the PR stop (a) while going to the k'th case. If the PR does not stop, this is equal to $x_{23}(k)$.
$x_{25}(k)$	The moment that the PR moved just out of the STack check Object (STO) camera vision while going to the k'th case.
$x_{26}(k)$	The moment that the STO starts making the photo and calculations of the k'th case.
$x_{27}(k)$	The moment that the PR is at the position where it is in the PUT before it picks up the k'th case.
$x_{28}(k)$	The moment that the PR is 35 above the PUT after it picked up the k'th case.
$x_{29}(k)$	The moment that the PUT floor arrives at the high position after the k'th case is picked up from the PUT.
$x_{30}(k)$	The moment that the PR arrives at the PR stop (b), 100 mm above the PUT, after picking up the k'th case.
$x_{31}(k)$	The moment that the PR leaves the PR stop (b) after picking up the k'th case.
$x_{32}(k)$	The moment that the PR is at the position where it unblocks the TUR from the PUT while carrying the k'th case.
$x_{33}(k)$	The moment that the k'th case is placed on the stack.

A-2 Timing Variables

Table A-2: List of timing variable with descriptions

Timing variable	Description	From	To
t_1 / t_{11}	The time it takes to move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 7.	$x_{12}(k)$	$x_{13}(k)$
t_2	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the TUR stop (b) when there is one case in the tray.	$x_{13}(k)$	$x_{18}(k)$
t_3	The time it takes to move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there is one case in the tray. Furthermore the case that is handled in this timing variable came from tray lift 7.	$x_{19}(k)$	$x_{20}(k)$
t_4	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray before TUR stop (b).	$x_{13}(k)$	$x_{14}(k)$
t_5	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 7) towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray after TUR stop (b).	$x_{19}(k)$	$x_{14}(k)$
t_6	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR is just out of the TPO camera vision when there is one case in the tray. Furthermore the TUR should be out of sight before TUR stop (b).	$x_{13}(k)$	$x_{15}(k)$

t_7	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 7) towards the position where the TUR is just out of the TPO camera vision when there is one case in the tray. Furthermore the TUR should be out of sight after TUR stop (b).	$x_{19}(k)$	$x_{15}(k)$
t_8 / t_{18}	The time it takes to move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 7.	$x_{20}(k)$	$x_{21}(k)$
t_{12}	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the TUR stop (b) when there are two or more cases in the tray.	$x_{13}(k)$	$x_{18}(k)$
t_{13}	The time it takes to move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there are two or more cases in the tray. Furthermore the case that is handled in this timing variable came from tray lift 7.	$x_{19}(k)$	$x_{20}(k)$
t_{14}	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray before TUR stop (b).	$x_{13}(k)$	$x_{14}(k)$
t_{15}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 7) towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray after TUR stop (b).	$x_{19}(k)$	$x_{14}(k)$
t_{16}	The time it takes to move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight before TUR stop (b).	$x_{13}(k)$	$x_{15}(k)$

t_{17}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 7) towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight after TUR stop (b).	$x_{19}(k)$	$x_{15}(k)$
t_{21} / t_{31}	The time it takes to move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 25.	$x_{12}(k)$	$x_{13}(k)$
t_{22}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the TUR stop (b) when there is one case in the tray.	$x_{13}(k)$	$x_{18}(k)$
t_{23}	The time it takes to move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there is one case in the tray. Furthermore the case that is handled in this timing variable came from tray lift 25.	$x_{19}(k)$	$x_{20}(k)$
t_{24}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray before TUR stop (b).	$x_{13}(k)$	$x_{16}(k)$
t_{25}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray after TUR stop (b).	$x_{19}(k)$	$x_{16}(k)$
t_{26}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR is just out of the TPO camera vision when there is one case in the tray. Furthermore the TUR should be out of sight before TUR stop (b).	$x_{13}(k)$	$x_{17}(k)$

t_{27}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR is just out of the TPO camera vision when there is one case in the tray. Furthermore the TUR should be out of sight after TUR stop (b).	$x_{19}(k)$	$x_{17}(k)$
t_{28} / t_{38}	The time it takes to move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 25.	$x_{20}(k)$	$x_{21}(k)$
t_{32}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the TUR stop (b) when there are two or more cases in the tray.	$x_{13}(k)$	$x_{18}(k)$
t_{33}	The time it takes to move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there are two or more cases in the tray. Furthermore the case that is handled in this timing variable came from tray lift 25.	$x_{19}(k)$	$x_{20}(k)$
t_{34}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray before TUR stop (b).	$x_{13}(k)$	$x_{16}(k)$
t_{35}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray after TUR stop (b).	$x_{19}(k)$	$x_{16}(k)$
t_{36}	The time it takes to move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight before TUR stop (b).	$x_{13}(k)$	$x_{17}(k)$

t_{37}	The time it takes to move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight after TUR stop (b).	$x_{19}(k)$	$x_{17}(k)$
t_{41}	The time it takes to make the TPO photo of a case.	1. $x_1(k)$ 2. $x_6(k)$ 3. $x_{15}(k-1)$ 4. $x_{17}(k-1)$	1. $x_2(k)$ 2. $x_7(k)$ 3. $x_2(k)$ 4. $x_7(k)$
t_{42}	The time it takes to make the TPO and DLO calculations of a case, when the case is the first case taken from the tray.	1. $x_2(k)$ 2. $x_7(k)$	1. $x_5(k)$ 2. $x_{10}(k)$
t_{43}	The time it takes to make the TPO and DLO calculations of a case, when the case is not the first case taken from the tray.	1. $x_2(k)$ 2. $x_7(k)$	1. $x_5(k)$ 2. $x_{10}(k)$
t_{44}	The time it takes to move a tray lift from the height on which the TPO photo is taken towards the upper height.	1. $x_3(k)$ 2. $x_8(k)$	1. $x_4(k)$ 2. $x_9(k)$
t_{45}	The time it takes to move a tray lift from the upper height towards the lower height, to exchange trays and to move the tray lift from the lower height towards the height on the TPO photo is taken.	1. $x_{14}(k-1)$ 2. $x_{16}(k-1)$	1. $x_1(k)$ 2. $x_6(k)$
t_{46}	The time it takes to move the PUT floor from the high position to the low position.	$x_{20}(k)$	$x_{22}(k)$
t_{47}	The time it takes to move the TUR from the position where it placed the case on the PUT towards the TUR stop (a).	$x_{20}(k-1)$	$x_{11}(k)$
t_{51}	The time it takes to move the PR from the position where it placed the case on the stack towards the PR stop (a).	$x_{33}(k-1)$	$x_{23}(k)$

t_{52}	The time it takes to move the PR away from the lift shaft added to the time it takes to lower the finished stack and extend platform 1 such that a case can be placed on it.	$x_{33}(k-1)$	$x_{26}(k)$
t_{53}	The time it takes to move the PR away from the lift shaft added to the time it takes to change platform of the finished stack, to lower that stack, to lift platform 1 up again and to extend platform 1 such that a case can be placed on it.	$x_{33}(k-1)$	$x_{26}(k)$
t_{54}	The time it takes to move the PR away from the lift shaft added to the time it takes to lower the shaft to the lift takeover height, perform a lift takeover, and lower the shaft to the new placement height.	$x_{33}(k-1)$	$x_{26}(k)$
t_{55}	The time it takes to move the PR away from the lift shaft added to the time it takes to lower the carrier lift to the position on which the case is placed.	$x_{33}(k-1)$	$x_{26}(k)$
t_{56}	The time it takes to get and place a slip sheet from the position where the PR placed the case added to the time it takes to move the PR from the position where it placed the case on the stack towards the PR stop (a).	$x_{33}(k-1)$	$x_{23}(k)$
t_{61}	The time it takes to move the PR from the PR stop (a) towards the position where the PR is just out of the STO camera vision.	$x_{24}(k)$	$x_{25}(k)$
t_{62}	The time it takes to move the PR from the PR stop (a) towards the position in the PUT.	$x_{24}(k)$	$x_{27}(k)$
t_{63}	The time it takes to make the STO photo and calculations.	$x_{26}(k)$	$x_{31}(k)$
t_{64}	The time it takes to move the PUT floor from the low position to the high position.	$x_{28}(k)$	$x_{29}(k)$
t_{65}	The time it takes to move the PR from the PR stop (b) towards the position where it unblocks the TUR from the PUT.	$x_{31}(k)$	$x_{32}(k)$
t_{66}	The time it takes to move the PR from the PR stop (b) towards the position where it placed the case on the stack.	$x_{31}(k)$	$x_{33}(k)$

t_{67}	The time it takes to move the PR from the position in the PUT towards the PR stop (b), 100 mm above the PUT.	$x_{27}(k)$	$x_{30}(k)$
t_{68}	The time it takes to move the PR from the position in the PUT towards 35 mm above the PUT.	$x_{27}(k)$	$x_{28}(k)$

A-3 Synchronisations

Table A-3: Synchronisations of events in part I of the system per mode

Event A	Event B	I.1	I.2	I.3	I.4	I.5	I.6	I.7	I.8
$x_2(k)$	$x_3(k)$	x	x			x	x		
$x_4(k)$	$x_{12}(k)$	x				x			
$x_5(k)$	$x_{12}(k)$	x	x			x	x		
$x_7(k)$	$x_8(k)$			x	x			x	x
$x_9(k)$	$x_{12}(k)$			x				x	
$x_{10}(k)$	$x_{12}(k)$			x	x			x	x
$x_{11}(k)$	$x_{12}(k)$	x	x	x	x	x	x	x	x
$x_{18}(k)$	$x_{19}(k)$	x	x	x	x	x	x	x	x
$x_1(k-1)$	$x_1(k)$		x	x	x		x	x	x
$x_2(k-1)$	$x_2(k)$			x	x			x	x
$x_3(k-1)$	$x_3(k)$			x	x			x	x
$x_4(k-1)$	$x_4(k)$		x	x	x		x	x	x
$x_5(k-1)$	$x_5(k)$			x	x			x	x
$x_6(k-1)$	$x_6(k)$	x	x		x	x	x		x
$x_7(k-1)$	$x_7(k)$	x	x			x	x		
$x_8(k-1)$	$x_8(k)$	x	x			x	x		
$x_9(k-1)$	$x_9(k)$	x	x		x	x	x		x
$x_{10}(k-1)$	$x_{10}(k)$	x	x			x	x		
$x_{14}(k-1)$	$x_{14}(k)$			x	x			x	x
$x_{15}(k-1)$	$x_{15}(k)$			x	x			x	x
$x_{16}(k-1)$	$x_{16}(k)$	x	x			x	x		
$x_{17}(k-1)$	$x_{17}(k)$	x	x			x	x		

Table A-4: Synchronisations of events in part II of the system per mode

Event A	Event B	II.1	II.2	II.3	II.4	II.5	II.6
$x_{21}(k)$	$x_{24}(k)$	x	x	x	x	x	x
$x_{22}(k)$	$x_{24}(k)$	x	x	x	x	x	x
$x_{23}(k)$	$x_{24}(k)$	x	x	x	x	x	x
$x_{25}(k)$	$x_{26}(k)$	x	x	x	x	x	x
$x_{30}(k)$	$x_{31}(k)$	x	x	x	x	x	x
$x_{29}(k-1)$	$x_{19}(k)$	x	x	x	x	x	x
$x_{32}(k-1)$	$x_{19}(k)$	x	x	x	x	x	x

A-4 Validation

$$A = \begin{bmatrix}
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.504 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.7003 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.7003 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.5299 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.3459 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.504 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.7003 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3.7003 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.5299 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.3459 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 9.3897 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5.2598 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1.6668 & \varepsilon & \varepsilon & 1.6668 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 9.3897 & \varepsilon & 4.5299 & \varepsilon & \varepsilon & \varepsilon & 5.2598 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1.6668 & \varepsilon & \varepsilon & 1.6668 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 10.6808 & \varepsilon & 5.8209 & \varepsilon & \varepsilon & \varepsilon & 6.5509 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2.9578 & \varepsilon & \varepsilon & 2.9578 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.7611 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2.6312 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 8.4134 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.2835 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0.6905 & \varepsilon & \varepsilon & 0.6905 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13.9301 & \varepsilon & 9.0703 & \varepsilon & \varepsilon & \varepsilon & 9.8002 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.2071 & \varepsilon & \varepsilon & 6.2071 & 4.276 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 12.7879 & \varepsilon & 7.9281 & \varepsilon & \varepsilon & \varepsilon & 8.658 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5.0649 & \varepsilon & \varepsilon & 5.0649 & 3.1338 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 12.441 & \varepsilon & 7.5812 & \varepsilon & \varepsilon & \varepsilon & 8.3111 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.7181 & \varepsilon & \varepsilon & 4.7181 & \varepsilon \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 12.441 & \varepsilon & 7.5812 & \varepsilon & \varepsilon & \varepsilon & 8.3111 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4.7181 & \varepsilon & \varepsilon & 4.7181 & 2.7869 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13.7149 & \varepsilon & 8.8551 & \varepsilon & \varepsilon & \varepsilon & 9.585 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5.992 & \varepsilon & \varepsilon & 5.992 & 4.0608 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13.994 & \varepsilon & 9.1342 & \varepsilon & \varepsilon & \varepsilon & 9.8641 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.2711 & \varepsilon & \varepsilon & 6.2711 & 4.3399 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14.1149 & \varepsilon & 9.2551 & \varepsilon & \varepsilon & \varepsilon & 9.985 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.392 & \varepsilon & \varepsilon & 6.392 & 4.4608 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 13.6194 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 9.4895 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 5.8964 & \varepsilon & \varepsilon & 5.8964 & 5.2675 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14.1149 & \varepsilon & 9.2551 & \varepsilon & \varepsilon & \varepsilon & 9.985 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.392 & \varepsilon & \varepsilon & 6.392 & 5.2675 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14.693 & \varepsilon & 9.8331 & \varepsilon & \varepsilon & \varepsilon & 10.5631 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.97 & \varepsilon & \varepsilon & 6.97 & 5.8456 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 14.693 & \varepsilon & 9.8331 & \varepsilon & \varepsilon & \varepsilon & 10.5631 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 6.97 & \varepsilon & \varepsilon & 6.97 & 5.8456 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.1114 & \varepsilon & 10.2516 & \varepsilon & \varepsilon & \varepsilon & 10.9815 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7.3884 & \varepsilon & \varepsilon & 7.3884 & 6.264 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.1942 & \varepsilon & 10.3344 & \varepsilon & \varepsilon & \varepsilon & 11.0643 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7.4713 & \varepsilon & \varepsilon & 7.4713 & 6.3468 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.5942 & \varepsilon & 10.7344 & \varepsilon & \varepsilon & \varepsilon & 11.4643 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7.8713 & \varepsilon & \varepsilon & 7.8713 & 6.7468 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.3481 & \varepsilon & 10.4883 & \varepsilon & \varepsilon & \varepsilon & 11.2182 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7.6252 & \varepsilon & \varepsilon & 7.6252 & 6.5007 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.3481 & \varepsilon & 10.4883 & \varepsilon & \varepsilon & \varepsilon & 11.2182 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7.6252 & \varepsilon & \varepsilon & 7.6252 & 6.5007 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 15.864 & \varepsilon & 11.0042 & \varepsilon & \varepsilon & \varepsilon & 11.7341 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 8.1411 & \varepsilon & \varepsilon & 8.1411 & 7.0166 \\
\varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 17.3067 & \varepsilon & 12.4469 & \varepsilon & \varepsilon & \varepsilon & 13.1768 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 9.5838 & \varepsilon & \varepsilon & 9.5838 & 8.4593
\end{bmatrix} \tag{A-1}$$

$$\lambda = \lim_{k \to \infty} \frac{x(k; x_0)}{k} = \begin{bmatrix} 9.585 \\ 9.585 \end{bmatrix} \tag{A-2}$$

Appendix B

Critical Circuit Analysis

Table B-1: Contribution to the critical circuit per timing variable

Timing Variable	Description (The time it takes to:)	Contribution in critical circuit
47	move the Tray Unloading Robot (TUR) from the position where it placed the case on the PickUp Table (PUT) towards TUR stop (a)	0.1375
66	move the Palletiser Robot (PR) from the PR stop (b) towards the position where it placed the case on the stack	0.0891
67	move the PR from the position in the PUT towards PR stop (b), 100 mm above the PUT	0.0853
62	move the PR from the PR stop (a) towards the position in the PUT	0.0853
51	move the PR from the position where it placed the case on the stack towards PR stop (a)	0.0670
13	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there are two or more cases in the tray. Furthermore the case that is handled in this timing variable came from tray lift 7	0.0509
11	move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 7	0.0476
12	move the TUR from the position where it starts pushing the case at lift 7 towards TUR stop (b) when there are two or more cases in the tray	0.0473

33	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there are two or more cases in the tray. Furthermore the case that is handled in this timing variable came from tray lift 25	0.0467
31	move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 25	0.0427
32	move the TUR from the position where it starts pushing the case at lift 25 towards TUR stop (b) when there are two or more cases in the tray	0.0425
23	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there is one case in the tray. Furthermore the case that is handled in this timing variable came from tray lift 25	0.0319
3	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT when there is one case in the tray. Furthermore the case that is handled in this timing variable came from tray lift 7	0.0292
21	move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 25	0.0269
22	move the TUR from the position where it starts pushing the case at lift 25 towards TUR stop (b) when there is one case in the tray	0.0266
63	make the STack check Object (STO) photo and calculations	0.0250
1	move the TUR from TUR stop (a) towards the position where it starts pushing the case at lift 7	0.0220
2	move the TUR from the position where it starts pushing the case at lift 7 towards TUR stop (b) when there is one case in the tray	0.0218
65	move the PR from PR stop (b) towards the position where it unblocks the TUR from the PUT	0.0213
46	move the PUT floor from the high position to the low position	0.0189
55	move the PR away from the lift shaft added to the time it takes to lower the carrier lift to the position on which the case is placed	0.0078
52	move the PR away from the lift shaft added to the time it takes to lower the finished stack and extend platform 1 such that a case can be placed on it	0.0060

54	move the PR away from the lift shaft added to the time it takes to lower the shaft to the lift takeover height, perform a lift takeover, and lower the shaft to the new placement height	0.0059
61	move the PR from PR stop (a) towards the position where the PR is just out of the STO camera vision	0.0045
41	make the Tray Pattern detection Object (TPO) photo of a case	0.0016
56	get and place a slip sheet from the position where the PR placed the case added to the time it takes to move the PR from the position where it placed the case on the stack towards PR stop (a)	0.0015
18	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 7	0.0011
53	move the PR away from the lift shaft added to the time it takes to change platform of the finished stack, to lower that stack, to lift platform 1 up again and to extend platform 1 such that a case can be placed on it	0.0008
45	move a tray lift from the upper height towards the lower height, to exchange trays and to move the tray lift from the lower height towards the height on the TPO photo is taken	0.0008
43	make the TPO and Descrambling Logic Object (DLO) calculations of a case, when the case is not the first case taken from the tray	0.0008
37	move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight after TUR stop (b)	0.0006
38	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 25	0.0005
8	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 7	0.0005
44	move a tray lift from the height on which the TPO photo is taken towards the upper height	0.0004

42	make the TPO and DLO calculations of a case, when the case is the first case taken from the tray	0.0004
24	move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray before TUR stop (b)	0.0003
28	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 25	0.0003
4	move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR pushed the case just off the tray when there is one case in the tray. Furthermore the case must be pushed off the tray before TUR stop (b)	0.0002
14	the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray before TUR stop (b)	0.0002
16	move the TUR from the position where it starts pushing the case at lift 7 towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight before TUR stop (b)	0.0001
34	move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray before TUR stop (b)	0.0001
35	move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray after TUR stop (b)	0.0001
36	move the TUR from the position where it starts pushing the case at lift 25 towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight before TUR stop (b)	0.00001

17	move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 7) towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight after TUR stop (b)	0.00001

Table B-2: Contribution to the critical circuit per clustered timing variables

Timing Variable	Description (The time it takes to:)	Contribution in critical circuit
3, 13, 23 and 33	move the TUR from TUR stop (b) towards the position where the TUR placed the case on the PUT	0.1587
1, 11, 21 and 31	move the TUR from TUR stop (a) towards the position where it starts pushing the case	0.1392
2, 12, 22 and 32	move the TUR from the position where it starts pushing the case towards TUR stop (b)	0.1382
47	move the TUR from the position where it placed the case on the PUT towards TUR stop (a)	0.1375
66	move the PR from the PR stop (b) towards the position where it placed the case on the stack	0.0891
67	move the PR from the position in the PUT towards PR stop (b), 100 mm above the PUT	0.0853
62	move the PR from the PR stop (a) towards the position in the PUT	0.0853
51	move the PR from the position where it placed the case on the stack towards PR stop (a)	0.0670
63	make the STO photo and calculations	0.0250
65	move the PR from PR stop (b) towards the position where it unblocks the TUR from the PUT	0.0213
52, 53, 54 and 55	move the PR away from the lift shaft added to the time it takes lower the carrier lift and perform a potential extra lift action	0.0205
46	move the PUT floor from the high position to the low position	0.0189

61	move the PR from PR stop (a) towards the position where the PR is just out of the STO camera vision	0.0045
8, 18, 28 and 38	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more	0.0025
41	make the TPO photo of a case	0.0016
56	get and place a slip sheet from the position where the PR placed the case added to the time it takes to move the PR from the position where it placed the case on the stack towards PR stop (a)	0.0015
45	move a tray lift from the upper height towards the lower height, to exchange trays and to move the tray lift from the lower height towards the height on the TPO photo is taken	0.0008
43	make the TPO and DLO calculations of a case, when the case is not the first case taken from the tray	0.0008
4, 14, 24 and 34	move the TUR from the position where it starts pushing the case towards the position where the TUR pushed the case just off the tray. Furthermore the case must be pushed off the tray before TUR stop (b)	0.0008
17 and 37	move the TUR from the position where it starts pushing the case at TUR stop (b) towards the position where the TUR is just out of the TPO camera vision when there are two or more cases in the tray. Furthermore the TUR should be out of sight after TUR stop (b)	0.00061
38	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 25	0.0005
8	move the TUR from the position where it placed the case on the PUT towards the position where it does not block the PR from the PUT any more while it moves towards tray lift 7	0.0005
44	move a tray lift from the height on which the TPO photo is taken towards the upper height	0.0004
42	make the TPO and DLO calculations of a case, when the case is the first case taken from the tray	0.0004

16 and 36	move the TUR from the position where it starts pushing the case towards the position where the TUR is just out of the TPO camera vision. Furthermore the TUR should be out of sight before TUR stop (b)	0.00011
35	move the TUR from the position where it starts pushing the case at TUR stop (b) (coming from lift 25) towards the position where the TUR pushed the case just off the tray when there are two or more cases in the tray. Furthermore the case must be pushed off the tray after TUR stop (b)	0.0001

Bibliography

[1] Vanderlande, "Company profile." https://www.vanderlande.com/about-vanderlande/company-profile, 2017. Accessed: 2017-06-07.

[2] Vanderlande, "Acp description." https://www.vanderlande.com/warehouse-automation/innovative-systems/order-fulfilment/automated-case-picking, 2017. Accessed: 2017-06-07.

[3] J. J. Kostelijk, *System/Subsystem Design Description - Auto Case Palletiser*. Vanderlande Industries Holding B.V., 2017. Internal Document.

[4] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and linearity: an algebra for discrete event systems*. John Wiley & Sons Ltd, 1992.

[5] T. J. van den Boom and B. De Schutter, "Modelling and control of discrete event systems using switching max-plus-linear systems," *Control engineering practice*, vol. 14, no. 10, pp. 1199–1211, 2006.

[6] Used picture from internal site https://devcolla.vanderlande.com/display/ACPS/Functional+Module+Case+Palletiser. Accessed: 2017-05-24.

[7] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006.

[8] T. J. van den Boom and B. De Schutter, "Modeling and control of switching max-plus-linear systems with random and deterministic switching," *Discrete Event Dynamic Systems*, vol. 22, no. 3, pp. 293–332, 2012.

[9] D. Vose, *Risk analysis: a quantitative guide*. John Wiley & Sons, 2008.

[10] S. Adyanthaya, Z. Zhang, M. Geilen, J. Voeten, T. Basten, and R. Schiffelers, "Robustness analysis of multiprocessor schedules," in *Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014 International Conference on*, pp. 9–17, IEEE, 2014.

[11] W. Weibull *et al.*, "A statistical distribution function of wide applicability," *Journal of applied mechanics*, vol. 18, no. 3, pp. 293–297, 1951.

[12] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, 1981.

[13] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach.* Cambridge university press, 2007.

[14] Y. Yang, M. Geilen, T. Basten, S. Stuijk, and H. Corporaal, "Automated bottleneck-driven design-space exploration of media processing systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1041–1046, European Design and Automation Association, 2010.

[15] C. Roser, M. Nakano, and M. Tanaka, "Productivity improvement: shifting bottleneck detection," in *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, pp. 1079–1086, Winter Simulation Conference, 2002.

[16] M. Hendriks, J. Verriet, T. Basten, B. Theelen, M. Brassé, and L. Somers, "Analyzing execution traces: critical-path analysis and distance analysis," *International Journal on Software Tools for Technology Transfer*, pp. 1–24, 2016.

[17] J. C. Tiernan, "An efficient search algorithm to find the elementary circuits of a graph," *Communications of the ACM*, vol. 13, no. 12, pp. 722–726, 1970.

[18] C. Roser, M. Nakano, and M. Tanaka, "A practical bottleneck detection method," in *Proceedings of the 33nd conference on Winter simulation*, pp. 949–953, IEEE Computer Society, 2001.

[19] S. R. Lawrence and A. H. Buss, "Shifting production bottlenecks: causes, cures, and conundrums," *Production and operations management*, vol. 3, no. 1, pp. 21–37, 1994.

[20] A. Saltelli, "Sensitivity analysis for importance assessment," *Risk Analysis*, vol. 22, no. 3, pp. 579–590, 2002.

[21] P. A. Samuelson and W. Nordhaus, *Economics (16th edn).* 1998.

[22] T. J. van den Boom and B. De Schutter, "Analytic expressions in stochastic max-plus-linear algebra," in *53rd IEEE Conference on Decision and Control*, pp. 1608–1613, IEEE, 2014.

[23] T. Van den Boom and B. De Schutter, "Model predictive control for perturbed max-plus-linear systems: A stochastic approach," *International Journal of Control*, vol. 77, no. 3, pp. 302–309, 2004.

[24] S. S. Farahani, T. van den Boom, H. van der Weide, and B. De Schutter, "An approximation method for computing the expected value of max-affine expressions," *European Journal of Control*, vol. 27, pp. 17–27, 2016.

[25] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[26] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[27] G. H. Weiss, “Random walks,” *Encyclopedia of Statistical Sciences*, 1983.

[28] R. B. Arellano-Valle and M. G. Genton, “On the exact distribution of the maximum of absolutely continuous dependent random variables,” *Statistics & Probability Letters*, vol. 78, no. 1, pp. 27–35, 2008.

Glossary

List of Acronyms

ABB	Asea Brown Boveri
ACP	Automated Case Picking
CDF	Cumulative Distribution Function
DEDS	Discrete Event Dynamic System
DLO	Descrambling Logic Object
LFL	Load Forming Logic
MPL	Max-Plus Linear
MPS	Max-Plus Scaling
PDF	Probability Density Function
PR	Palletiser Robot
PUT	PickUp Table
RSMPL	Random Switching Max-Plus Linear
SMPL	Switching Max-Plus Linear
SSMPL	Switching Stochastic Max-Plus Linear
STO	STack check Object
TPO	Tray Pattern detection Object
TUR	Tray Unloading Robot