

Lukas Fleischer

Eventbasierte Simulation eines Shuttle-Regal-Systems

Diplomarbeit

Wirtschaftsingenieurswesen - Maschinenbau

Technische Universität Graz
Fakultät für Maschinenbau und Wirtschaftswissenschaften

Institut für Technische Logistik
Vorstand: Univ.-Prof. Dr.-Ing. habil. Dirk Jodin

Betreuer: Univ.-Prof. Dr. Georg Kartnig

Wien, November 2013

This document is set in Palatino, compiled with pdfL^AT_EX₂ε and Biber.

The L^AT_EX template from Karl Voit is based on KOMA script and can be found online:
<https://github.com/novoid/LaTeX-KOMA-template>

Eidesstattliche Erklärung¹

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am _____
Datum

Unterschrift

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, _____
Date

Signature

¹Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Abstract

Diese Arbeit beschäftigt sich mit der Erstellung einer eventbasierten Simulation eines Shuttle-Regal-Systems. Diese Systeme zeichnen sich dadurch aus, dass sich in jeder Ebene ein Shuttle befindet, das die horizontalen Bewegungen durchführt; unabhängig davon übernimmt ein Liftsystem die vertikalen Bewegungen. Die beiden entkoppelten Systeme sind durch einen Puffer verbunden, der als Schnittstelle dient und an dem die Lagereinheiten von dem einen in den anderen Prozess übergeben werden. Durch eingehende Analyse dieses hochkomplexen Systems wird ein Prozessmodell gewonnen, das insbesondere Doppelspielzeiten realisiert. Unter Voraussetzung bestimmter vereinfachender Annahmen ergibt sich ein Simulationsmodell, dessen Funktionsweise anschließend ausführlich erläutert wird. Im Ergebnis werden die durch die Simulation generierten Daten mit Hilfe grafischer Darstellungen analysiert – das ermöglicht dann Rückschlüsse auf das Verhalten des Systems. Somit lässt sich nun ein Zusammenhang zwischen der Umschlagleistung und den geometrischen Parametern, Länge und Anzahl der Ebenen, feststellen; im Fokus steht hier insbesondere der Einfluss der Wartezeiten. Es ist evident, dass eine Optimierung der Umschlagleistung durch Minimierung der Wartezeiten erreicht werden kann.

The intent of this thesis is the creation of an event-based simulation of shuttle storage systems. Typical characteristics of this system are the shuttles located on every floor moving only in a horizontal direction. Vertical movement is achieved by independently moving elevators. These two processes are connected by a buffer-station, where storage units are transferred between them. After a thorough analysis of the complex system specialized in the generation of doublecycle times a process model is elaborated. This model is used as the foundation of a specific simulation based on numerous assumptions to simplify the complexity of the original system. A detailed description of the simulation-process is meant to deliver insight into the logic of the program. The presented results are based upon the data calculated within the simulation program itself. To achieve a better understanding of correlations of the system a wide variety of different plots are used. The conclusion reached by this thesis shows that geometric parameters such as the length and number of floors have a direct influence on the doublecycle times and therefore on the maximum throughput possible. Of particular interest seems the impact of occurring waiting times and their reduction.

Inhaltsverzeichnis

Abstract	iv
1 Einleitung	1
1.1 Überblick über das Shuttle-Regal-System	1
1.1.1 Neue Anforderungen - Neue Konzepte	1
1.1.2 Das Shuttle-Prinzip	2
1.1.3 Anwendungen in der Wirtschaft	3
1.2 Aufgabenstellung und Zielsetzung	4
1.3 Methode und Arbeitsschritte	5
2 Grundlagen	6
2.1 Begriffe und Darstellung	6
2.2 Software: MATLAB	7
2.3 Notation: BPMN 2.0	8
3 Erarbeitung eines Modells	11
3.1 Analyse der Prozesse	11
3.2 Modell der Prozessabläufe	14
3.3 Annahmen zum Aufbau der Simulation	17
4 Aufbau der Simulation	20
4.1 Simulationsmethodik	20
4.2 Optimierung der eventbasierten Struktur	21
4.3 Von Systemkomponenten zu Entitäten	21
4.4 Modell der eventbasierten Simulation	23
4.5 Erweiterung des Modells: Die „aktiven Shuttles“	24
4.6 Die Grenzen der Simulation	26
5 Funktionsweise der Simulation	27
5.1 Ablauf der Simulation	27
5.1.1 Die Parameter	28
5.1.2 Die Umgebung	29
5.1.3 Der Einlagerungszyklus	31
5.1.4 Der Zwischenzyklus	32
5.1.5 Der Auslagerungszyklus	32
5.2 Beschreibung der Funktionen	33
5.2.1 $[Calculate\ Shuttle\ Time]_f$	34
5.2.2 $[Calculate\ Lift\ Time]_f$	35

5.2.3	[<i>Check Shuttle Status</i>] _f	35
5.2.4	[<i>Define next Step</i>] _f	37
5.2.5	[<i>Calculate real Time</i>] _f	38
5.3	Exemplarische Berechnung der Doppelspielzeit	41
5.3.1	Start der Simulation	42
5.3.2	Berechnung des ersten Zyklus	43
5.3.3	Berechnung zur Laufzeit	46
5.4	Funktionen der Benutzeroberfläche	48
5.4.1	Die Simulation	48
5.4.2	Die Variation	49
5.4.3	Die grafische Ausgabe	51
6	Ergebnisse und Interpretation	53
6.1	Die Simulationsergebnisse	53
6.2	Die Definition der Parameter	54
6.3	Analyse der Zyklen	55
6.3.1	Die Systeme im Detail	55
6.3.2	Vergleich der Grundsysteme	59
6.3.3	Vergleich von modifizierten Grundsystemen	61
6.4	Analyse der Variationen	65
6.4.1	Variation der Länge	65
6.4.2	Variation der Shuttles	68
6.4.3	Variation der „aktiven Shuttles“	70
6.5	Interpretation der Ergebnisse	74
6.5.1	Entwicklung der Umschlagleistung	74
6.5.2	Zusammenhang der Umschlagleistung und der Lagerplätze	76
6.5.3	Kritische Annahmen und Optimierungspotential der Simulation	78
6.6	Ergebnisse verschiedener Simulationen im Vergleich	81
7	Zusammenfassung und Ausblick	85
	Literatur	88

Abbildungsverzeichnis

1.1	Blick in einen Regalgang	2
2.1	Basiselemente des BPMN	8
2.2	Spezielle Elemente des BPMN	9
3.1	Farblegende	11
3.2	Prozess: Einlagerung	12
3.3	Prozess: Auslagerung	12
3.4	Kombination der Prozesse	12
3.5	Aufbau des Shuttle Systems	14
3.6	Modellierung der Prozessabläufe	15
4.1	Entitäten mit Standardwerten	22
4.2	Ableitung der Simulationszyklen	23
4.3	Ableitung der Simulationszyklen	25
5.1	Funktionsabfolge: Simulationsüberblick	27
5.2	Funktionsabfolge: Umgebungserstellung	30
5.3	Funktionsabfolge: Einlagerungszyklus	31
5.4	Funktionsabfolge: Zwischenzyklus	32
5.5	Funktionsabfolge: Auslagerungszyklus	33
5.6	Funktionsweise: Shuttlezeit Berechnung	34
5.7	Funktionsweise: Shuttle Aktualisierung	36
5.8	Funktionsweise: Nächster Auftrag	38
5.9	Funktionsweise: Zeitbestimmung	39
5.10	Funktionsabfolge der Variation	49
5.11	Die Benutzeroberfläche GUI Main	50
5.12	Die Benutzeroberfläche GUI Plot 3D	52
6.1	System 1 – Zykluszeiten	56
6.2	System 1 – Zykluszeiten normiert	56
6.3	System 1 – Doppelspielzeiten	56
6.4	System 1 – Doppelspielzeiten normiert	56
6.5	System 2 – Zykluszeiten	57
6.6	System 2 – Zykluszeiten normiert	57
6.7	System 2 – Doppelspielzeiten	58
6.8	System 2 – Doppelspielzeiten normiert	58
6.9	System 3 – Zykluszeiten	59
6.10	System 3 – Zykluszeiten normiert	59

6.11	System 3 – Doppelspielzeiten	59
6.12	System 3 – Doppelspielzeiten normiert	59
6.13	Zusammensetzung der Zykluszeiten – absolut	60
6.14	Zusammensetzung der Zykluszeiten – relativ	60
6.15	Entstehung der Zykluszeiten – absolut	60
6.16	Entstehung der Zykluszeiten – relativ	60
6.17	System 1 – Doppelspielzeiten normiert $n_S = 40$	61
6.18	System 3 – Doppelspielzeiten normiert $n_S = 40$	61
6.19	Zusammensetzung der Zykluszeiten – absolut $n_S = 40$	62
6.20	Entstehung der Zykluszeiten – relativ $n_S = 40$	62
6.21	System 1 – Doppelspielzeiten – normiert – $n_{aS} = 5$	63
6.22	System 3 – Doppelspielzeiten – normiert – $n_{aS} = 5$	63
6.23	Zusammensetzung der Zykluszeiten – absolut – $n_{aS} = 5$	63
6.24	Entstehung der Zykluszeiten – relativ – $n_{aS} = 5$	63
6.25	Zusammensetzung der Zykluszeiten – absolut – $n_{\ddot{U}S} = 10$	64
6.26	Entstehung der Zykluszeiten – relativ – $n_{\ddot{U}S} = 10$	64
6.27	Zusammensetzung der Zykluszeiten – absolut – ohne Wh.	65
6.28	Entstehung der Zykluszeiten – relativ – ohne Wh.	65
6.29	Vergleich der Koordinatenerzeugung – $n_S = 20$; $n_{aS} = 10$	66
6.30	Vergleich der Koordinatenerzeugung – $n_S = 40$; $n_{aS} = 10$	66
6.31	Vergleich der Übergabestelle – $n_S = 20$; $n_{aS} = 10$	67
6.32	Vergleich der Übergabestelle – $n_S = 40$; $n_{aS} = 10$	67
6.33	Vergleich der „aktiven Shuttles“ – $n_S = 20$	67
6.34	Vergleich der „aktiven Shuttles“ – $n_S = 40$	67
6.35	Vergleich der Koordinatenerzeugung – $l = 40$ m; $n_{aS} = 10$	68
6.36	Vergleich der Koordinatenerzeugung – $l = 80$ m; $n_{aS} = 10$	68
6.37	Vergleich der Übergabestelle – $l = 40$ m; $n_{aS} = 10$	69
6.38	Vergleich der Übergabestelle – $l = 80$ m; $n_{aS} = 10$	69
6.39	Vergleich der „aktiven Shuttles“ – $l = 40$ m	70
6.40	Vergleich der „aktiven Shuttles“ – $l = 80$ m	70
6.41	Einfluss der Länge – $n_{aS} = 10$	70
6.42	Einfluss der Länge – $n_{aS} = 20$	70
6.43	Vergleich der Koordinatenerzeugung – $l = 30$ m; $n_S = 20$	71
6.44	Vergleich der Koordinatenerzeugung – $l = 80$ m; $n_S = 20$	71
6.45	Einfluss der Länge bei hohen „aktiven Shuttles“ – $n_S = 20$	72
6.46	Einfluss der Länge bei hohen „aktiven Shuttles“ – $n_S = 40$	72
6.47	Einfluss der Länge – $n_S = 20$	73
6.48	Einfluss der Länge – $n_S = 40$	73
6.49	Einfluss der Shuttles – $l = 40$ m	73
6.50	Einfluss der Shuttles – $l = 80$ m	73
6.51	Umschlagleistung – $n_{aS} = 10$	74
6.52	Umschlagleistung in Abhängigkeit der Länge und der Shuttles	75
6.53	Umschlagleistung in Abhängigkeit der Länge und der Shuttles – Position der Übergabestelle in der Mitte des Liftes	75
6.54	Umschlagleistungen in Abhängigkeit der Shuttles und der Lagerpositionen	77
6.55	Umschlagleistungen in Abhängigkeit der Länge und der Lagerpositionen .	77

6.56	ARENA: Doppelspielzeiten bei konstanter Länge	83
6.57	Vergleich der Simulationen: MATLAB	84
6.58	Vergleich der Simulationen: ARENA	84

1 Einleitung

In diesem Kapitel soll ein Überblick über die Entwicklung der Shuttle-Regal-Systeme, die wichtigsten Komponenten und deren Vor- und Nachteile gegeben werden. Danach gilt es die Aufgabenstellung dieser Arbeit zu erläutern und die Arbeitsschritte festzulegen.

1.1 Überblick über das Shuttle-Regal-System

1.1.1 Neue Anforderungen - Neue Konzepte

Nachdem in der Zeit rund um die Entwicklung der Hochregallager Anfang der 60er-Jahre der Fokus auf neue Lagertechniken und -systeme gerichtet war, rückten andere Themen der allgemeinen Logistikentwicklung in den Vordergrund. Verstärkte Aufmerksamkeit erlangten beispielsweise Konzepte wie das Supply Chain Management, die Verkehrslogistik und der immer größer werdende Einfluss rechnergesteuerter Systeme mit integrierter Datenverwaltung, die direkt mit Enterprise Resource Planning (ERP) Software zusammenarbeiten können; dabei gelang es, die Lagersysteme immer besser in die komplexen betrieblichen Abläufe zu integrieren. In jüngster Vergangenheit erlangten aber andere Aspekte Priorität: nämlich Schonung der Ressourcen im Sinne von „Green Logistics“ und Steigerung der Flexibilität sowie Effizienz der Systeme. Diese Entwicklung führt zu einer Reihe von neuen Anforderungen auch an die Intralogistik:

Während beim Thema der „Green Logistics“ vor allem die Senkung des Energiebedarfs und des damit verbundenen CO₂-Ausstoßes eine zentrale Rolle spielt und z. B. durch Energie-Rückgewinnungssysteme und effizientere Antriebstechnologie bewerkstelligt wird, erfordert der Markt andererseits immer kürzere Zeitspannen zur Erledigung einer bestimmten Kundenbestellung. Beachtet man dabei die Tendenz zu einem immer umfangreicher werdenden Artikelsortiment bei zugleich immer kleiner werdenden Bestellmengen (hier sei insbesondere auf die Anforderungen des E-Commerce verwiesen, der im Unterabschnitt 1.1.3 näher ausgeführt wird), so müssen neue Wege gefunden werden, um die Wirtschaftlichkeit der Unternehmen nicht zu gefährden. So wurden in den letzten Jahren, zusätzlich zur Weiterentwicklung bereits vorhandener Systeme wie Behälter- und Palettenlager (etwa durch Modifizierung der Antriebskonzepte und verbesserte IT-Integration), auch einige neue Konzepte im Bereich der Lagersysteme entwickelt [Hei11]:

- Satellitenlager
- Kanallager
- Shuttle-Lager

In der Folge soll nun das Hauptaugenmerk auf die Shuttle-Regal-Systeme gelegt werden.

1.1.2 Das Shuttle-Prinzip

Betrachtet man den Begriff „Shuttle“ in einer analytischen Sichtweise und löst ihn dadurch von marketing-technischen Einflüssen, so lässt sich ein einheitliches Prinzip erkennen: Einzelne Fahrzeuge befinden sich in den horizontalen Ebenen der Lagergasse (eines Regals) und können mit geeigneten Lastaufnahmemitteln Lagereinheiten aufnehmen oder abgeben. Unabhängig davon werden Lagereinheiten in vertikaler Richtung mit Hilfe von Hubgeräten (z. B. Lifte oder Vertikalförderer) transportiert. Um die beiden Prozesse zu verbinden, werden in vielen Fällen Pufferplätze angebracht, um die Übergabe zwischen den Komponenten völlig zu entkoppeln. Betrachtet man nun die einzelnen notwendigen Prozessschritte, so scheint diese Vorgehensweise keine Verbesserung im Vergleich zu einem klassischen Regalbediengerät (RGB), sondern sie verursacht vielmehr eine Verminderung der Leistung. Denn wo bei einem RGB Diagonalfahrten möglich und nur zwei Übergabezeiten notwendig sind, sind bei einem Shuttle-System nur eindimensionale Bewegungen möglich und es treten zusätzliche Transferzeiten auf. Der entscheidende Vorteil erschließt sich erst, wenn man das System als Ganzes betrachtet: Durch die Entkoppelung der Bewegungsrichtungen kann nun der Lift nach der Übergabe der Lagereinheit an ein Shuttle, das diese zu ihrem Lagerort bringt, schon die nächste Einheit in einer anderen Ebene aufladen. Dies führt zu einem hochdynamischen System, bei dem die Transporte in horizontaler Richtung quasi im Hintergrund stattfinden, während die tatsächlich nach außen wirksamen Transportzeiten nur durch die vertikalen Förderkomponenten und die dort wirksamen Übergabezeiten bestimmt werden [Hei11].



Abbildung 1.1: Blick in einen Regalgang¹

Abbildung 1.1 soll nun einen Blick in das System hinein ermöglichen, um den Aufbau besser nachvollziehen zu können. Zu sehen sind die einzelnen Shuttles in einem OSR Shuttle™ System der Firma KNAPP AG, die sich in den jeweiligen Ebenen in horizontaler Richtung bewegen.

¹URL: <http://www.knapp.com/cms/download.php?download=72&show=1> (besucht am 30. 10. 2013)

Die modulare Bauweise dieser Systeme führt zu einer deutlichen Reduktion der Massen der einzelnen Komponenten, wodurch Energieeinsparungen im Sinne der „Green Logistics“ realisiert werden, aber auch eine höhere Beschleunigung der Vertikalförderer ermöglicht wird.

Abschließend sollen die Vor- und Nachteile dieses Systems gegenübergestellt werden [Hep11]:

Vorteile

- Hohe Leistungsfähigkeit
- Einfache Skalierbarkeit
- Einfachere Wartung (durch Wartungsplattformen)

Nachteile

- Hohe Investitionskosten
- Komplexe Lagersteuerung
- Großer Engineering Aufwand

1.1.3 Anwendungen in der Wirtschaft

In letzter Zeit sind die Auswirkungen der Marktentwicklung – weg vom Einzelhandel in Richtung E-Commerce und Multichannel-Vertrieb – auch in den Anforderungen an moderne Lagersysteme zu beobachten. Da in diesem Sektor mit weiterem Wachstum gerechnet wird, müssen Unternehmen ihre strategische Ausrichtung und damit auch die technischen Anlagen diesem Umstand anpassen. Auf diesem Gebiet konkurrieren viele Anbieter um die Akquirierung neuer Kunden, die durch optimale Liefergeschwindigkeiten und zeitnahe Auftrags erledigung auch langfristig an eine Online-Verkaufsplattform gebunden werden sollen – denn nur ein zufriedener Kunde tätigt eine weitere Bestellung. Technisch gesehen handelt es sich hierbei um eine Vielzahl kleiner Bestellungen, bestehend aus einer Mischung von Auftragspositionen aus verschiedenen Produktparten; das macht einen hohen Warenbestand zur Aufrechterhaltung der Lieferfähigkeit notwendig [Kna12b].

Hier soll als repräsentatives Beispiel das Distributionszentrum von Boots UK dienen, das gemeinsam mit dem Intralogistiksystemanbieter Knapp realisiert wurde. Boots UK ist ein führendes britisches Handelsunternehmen für Gesundheits- und Schönheitsprodukte; der Sitz des Distributionszentrums befindet sich in Burton-on-Trent. Zusätzlich zu den bereits oben genannten Anforderungen, sollen an diesem Standort auch das Filialgeschäft und der internationale Versand abgewickelt werden – saisonale Spitzen müssen berücksichtigt werden [Kna12a].

Das erarbeitete Lösungskonzept stellt ein Shuttle-Lager des Typs OSR-Shuttle in den Mittelpunkt des Systems. Dieses, auf Artikel unterschiedlicher Beschaffenheit ausgelegte, halb-automatische Lager- und Kommissioniersystem ist vor allem für Artikel aus dem Segment der Langsam- und Mitteldreher geeignet. Dabei müssen vier verschiedene Packungsgrößen, sortenreine Lagerung und Gewichtsklassen von 15 kg und 25 kg berücksichtigt werden. Über die Auslegung des Lagersystems hinaus sind noch andere Bereiche der Intralogistik von eminenter Bedeutung: die Kommissionierung, die automatische Verpackung und etwaige Entsorgung von Verpackungsmaterial sowie die Lagerverwaltung mit Hilfe einer integrierten Software, um nur einige zu erwähnen [vgl. Kna12a; Kna12b].

Dieses Beispiel soll verdeutlichen, wie gut die einzelnen Prozesse und verwendeten Technologien aufeinander abgestimmt sein müssen, um der Forderung nach Flexibilität ohne Einschränkung der Wirtschaftlichkeit entsprechen zu können. Dass dabei dem verwendeten Lagersystem eine zentrale Rolle zuteil wird, rechtfertigt die intensive Beschäftigung mit dieser Thematik.

1.2 Aufgabenstellung und Zielsetzung

Nachdem ein Großteil der veröffentlichten Werte für Umschlagleistungen von Shuttle-Regal-Systemen von Unternehmen zur Verfügung gestellt werden, ohne weiter auf deren Berechnung einzugehen, entstand der Wunsch nach einem unabhängigen Ansatz, um diese Zeiten nachvollziehen zu können. Nach eingehender Beschäftigung mit dieser Thematik und dem Versuch, eine analytische Formel zur Abschätzung der Doppelspielzeit aufstellen zu können, mussten die Ziele dieser Arbeit begrenzt und wie folgt definiert werden:

Es soll ein Modell eines realen Shuttle-Regal-Systems erstellt werden, mit besonderem Augenmerk auf das zustande kommen von Doppelspielzyklen, um diese mit den Doppelspielzeiten herkömmlicher Regalbediengeräte (RGB) oder automatischen Kleinteillagern (AKL) vergleichen zu können. Es werden dabei nur Systeme betrachtet, bei denen jeder Ebene des Lagers ein eigenständiges Shuttle zugeordnet ist.

Dazu ist es nötig, eine geeignete Software auszuwählen (hier `MATLAB`) und mit den vom Programm zur Verfügung gestellten Mitteln aus dem Modell der Prozesse eine auf Funktionen basierende Simulation abzuleiten. Um das komplexe reale System abbilden zu können, müssen in einer ersten Annäherung auch stark vereinfachende Annahmen zugelassen werden. Außerdem soll die Simulation unabhängig von bestehenden grafischen Simulationsprogrammen oder Simulationsprozessoren entwickelt werden.

Das zu erstellende Programm soll auch zu einem späteren Zeitpunkt in der Lage sein, anderen Anforderungen zu entsprechen – somit ist ein modularer Aufbau der Simulation unbedingt notwendig. Daraus resultiert eine Zerlegung des komplexen Gesamtsystems in einfache Funktionen, die variabel angepasst werden können, ohne die Funktionalität des Gesamtsystems zu beeinträchtigen.

Die Arbeit bezieht sich ausschließlich auf das Lagersystem selbst; die Peripherie wie etwa Kommissioniersysteme und Förderstrecken fließen nicht in die Betrachtung mit ein, sondern die Systemgrenzen werden über den Ein- und Austritt der Lagereinheiten definiert. Um den realen Ablauf des Shuttle-Regal-Systems besser nachvollziehen zu können, war es notwendig, einen kompetenten Partner aus der Wirtschaft zu gewinnen. Die Firma Knapp, die jahrelange Erfahrungen mit Shuttle-Systemen vorweisen kann, erklärte sich freundlicherweise bereit, in dieser Sache ihre Unterstützung zu gewähren. Durch diese Partnerschaft konnten wertvolle Daten über die Funktionsweise der komplexen Vorgänge gesammelt und dadurch eine noch besser auf den realen Prozess angepasste Simulation erstellt werden.

1.3 Methode und Arbeitsschritte

Um die später verwendete Systematik zu verstehen, wird diese vorab in Kapitel 2 besprochen. Hier wird einerseits auf spezielle Darstellungsarten, die in der Arbeit einheitlich angewendet werden, andererseits auf die verwendete Simulationssoftware MATLAB und die Prozessdarstellung mit Hilfe der BPMN (BUSINESS PROCESS MODEL AND NOTATION), einer vor allem in der Geschäftsprozessmodellierung verwendeten Notation, eingegangen. Kapitel 3 beschäftigt sich mit dem Aufbau eines Modelles, welches sowohl den realen Ablauf hinreichend genau abbildet als auch der Forderung nach Doppelspielzyklen gerecht wird. Nach einer Beschreibung des gewählten Systems und einer Analyse der Prozesse folgt eine Ausführung über die Vereinfachungen, die getroffen werden müssen, um eine funktionstüchtige Simulation zu erhalten.

Aus den dadurch gewonnenen Erkenntnissen werden in Kapitel 4 die relevanten Entitäten mit ihren Wertebereichen definiert. Ausgehend vom Prozessmodell wird nun ein angepasstes Modell für die eventbasierte Simulation erarbeitet und eine Variable eingeführt, die die Gleichzeitigkeit der Vorgänge widerspiegelt: die „aktiven Shuttles“.

In Kapitel 5 wird dann ausführlich auf die eigentliche Ausführung der Simulation eingegangen. Dazu werden die notwendigen Parameter festgelegt und das Grundgerüst der Simulation eines Zyklus erarbeitet. Danach werden alle auftretenden Funktionen einzeln im Detail besprochen, um ihre Logik zu verdeutlichen. Sodann wird mit zwei praktischen Beispielen die Berechnung der Zykluszeiten erläutert, um das konkrete Verhalten des Systems sowohl am Ausgangspunkt einer Simulation als auch in einem zufällig bestimmten Systemzeitpunkt beobachten zu können. Abschließend wird ein kurzer Überblick über die Verwendung und Optionen der grafischen Benutzeroberfläche gegeben, um einen leichten Einstieg in das Simulationsprogramm zu gewähren.

Kapitel 6 beschäftigt sich mit der Analyse der Daten, die mit Hilfe der Simulation gewonnen wurden. Hier wird insbesondere die Zusammensetzung der Doppelspielzeiten in einem Simulationsdurchlauf bei mehreren Systemkonfigurationen betrachtet. Danach werden die Auswirkungen der Parameter Länge, Shuttles und „aktive Shuttles“ auf die mittlere Doppelspielzeit analysiert. Ergänzend dazu wird auch der Einfluss der Anzahl der Lagerpositionen und der geometrischen Randbedingungen auf die Umschlagleistung aufgezeigt. Letztlich beschäftigt sich das Kapitel auch mit einer kritischen Betrachtung der der Simulation zu Grunde liegenden Arbeitshypothese und verweist auf mögliche Lösungsansätze. Abschließend wird ein Vergleich mit den Ergebnissen einer parallelen Masterarbeit gezogen, um die hier erarbeiteten Resultate zu verifizieren. Schließlich werden im letzten Kapitel die Ergebnisse der Arbeit zusammenfassend betrachtet und es wird ein Ausblick auf weitere mögliche Forschungsfragen eröffnet.

2 Grundlagen

Bevor mit den Erläuterungen zur tatsächlichen Prozesserstellung und Simulation begonnen werden kann, sollen in diesem Abschnitt einige Anmerkungen zur durchgängigen Formatierung, der verwendeten Software und der Notation der Prozessmodelle gemacht werden.

2.1 Begriffe und Darstellung

Das erstellte Simulationsprogramm, dessen Entstehungsprozess und Auswertung Inhalt dieser Arbeit sind, wurde über einen längeren Zeitraum konzipiert und programmiert. Um den Code auch über die regionalen sprachlichen Grenzen hinaus nachvollziehbar und verständlich zu machen, wurde dieser im Sinne der Internationalität komplett in Englisch gehalten. Dies steht auch im Einklang mit der Programmiersprache in der verwendeten Software MATLAB. Aus diesem Grund wurden auch die implementierte grafische Benutzeroberfläche sowie alle Eingaben und Ausgaben in englischer Sprache verfasst.

Da diese Arbeit auf dem oben beschriebenen Programm basiert und zur Erläuterung der logischen Abläufe dienen soll, sind auch alle Abbildungen und Legenden in Englisch gehalten, die Erklärungen im Text sind jedoch in deutscher Sprache verfasst. Dabei wird die Kenntnis der englischen Begriffe, die benötigt werden, um den Ausführungen folgen zu können, als bekannt vorausgesetzt. Die einzige Ausnahme davon bilden die Namen der aufgerufenen Funktionen, diese werden nicht übersetzt, sondern für eine bessere Erkennbarkeit im Textfluss nach folgendem Schema hervorgehoben:

[Funktionsname]_f

Ebenso wird die Änderung eines vordefinierten Wertes einer Entität folgendermaßen formatiert:

[Statusänderung]

Schlussendlich wird auch der selbst eingeführte Parameter der aktiven Shuttles mit einer eigenen Formatierung versehen, um Verwechslungen mit dem geometrischen Parameter: Anzahl der Shuttles, vorzubeugen. Dieser wird wie folgt gekennzeichnet:

„aktive Shuttles“

Um komplexe Entscheidungsprozesse leichter nachvollziehbar zu machen, wird in den Kapiteln 3, 4 und 5 ein einheitliches Farbschema verwendet, welches in Abbildung 3.1 zu sehen ist. In Kapitel 6 wird eine Vielzahl verschiedener Typen von grafischen Darstellungen verwendet, welche jeweils beim ersten Auftreten mit einer Legende versehen werden. Tritt derselbe Darstellungstyp ein weiteres Mal bei der Betrachtung eines anderen Datensatzes auf, so wird das zuvor definierte Farbschema beibehalten.

2.2 Software: MATLAB

Bei Matlab[®] handelt es sich um eine Software, mit der es möglich ist, komplexe mathematische und technische Probleme zu berechnen und zu simulieren. Dazu lassen sich einerseits die im Paket vorhandenen mathematischen Operatoren und Funktionen benützen, andererseits ermöglicht es MATLAB aber auch, eigene Funktionen zu programmieren. Die Programmiersprache bedient sich hauptsächlich mathematischer Notationen und erlaubt dabei auch Kontroll- und Datenstrukturen, die der Programmiersprache C ähneln. MATLAB wird von der Firma „The MathWorks, Inc.“ vertrieben. [vgl. Ste12, S. 15]

Für diese Arbeit wurde MATLAB in der Version R2012b herangezogen, vor allem geschuldet der Tatsache, dass sich die grafische Darstellungen von Ergebnissen aus großen Datenmengen sehr gut realisieren lässt. Da für die Berechnung der Werte keine komplexen mathematischen Funktionen notwendig sind, wird auf die vielfältigen gebotenen Möglichkeiten nicht weiter eingegangen. Zum besseren Verständnis über den Aufbau der Simulation soll aber ein kurzer Überblick über die von MATLAB verwendeten Datenstrukturen gegeben werden.

Die Bezeichnung MATLAB leitet sich aus der Abkürzung der beiden Begriffe MATRIX-LABORATORY ab, daher lässt sich auch leicht erklären, dass die **Matrix** der Standardtyp für die Erfassung komplexer Datenstrukturen ist. Die Indizierung folgt den in der Mathematik üblichen Regeln. Der Inhalt eines Feldes einer Matrix kann entweder eine Zahl (standardmäßig des Typs `double`) oder eine Zeichenkette sein. Hier sind eindeutig Zahlenwerte zu bevorzugen, da Zeichenketten in einer Zeile immer die gleiche Anzahl an Zeichen haben müssen.

Um Daten zu erzeugen, die in verschiedenen Zellen unterschiedliche Datentypen enthalten sollen, muss auf einen anderen Datenstrukturtyp zurückgegriffen werden: den **Cell-Array**. Dieser wird wie eine Matrix indiziert, kann aber nun in jedem Feld entweder eine Zahl, eine Matrix, eine beliebig lange Zeichenkette oder einen weiteren Cell-Array beinhalten. Diese Flexibilität wird in der Simulation vor allem dazu genutzt, den Programmablauf übersichtlicher zu gestalten. So wird der Status nicht nur mit `0` oder `1`, sondern intuitiv mit `empty` oder `ready` bezeichnet. Zusätzlich dazu lassen sich alle Entitäten jeweils zu einem Cell-Array zusammenfassen, was ihre Aufgabenbereiche noch deutlicher unterstreicht.

Um schließlich die Ergebnisse übersichtlich zu gestalten, wird der Datentyp **Struktur (struct)** benützt. Dieser ermöglicht es, Feldern einen Namen zuzuordnen und darin Daten zu speichern. Damit ist es möglich, die Resultate vieler verschiedener Simulationen geordnet nach ihren Parametern abzuspeichern.

Die Kenntnis über grundlegende Vorgehensweisen wie `if-then`-Bedingungen oder `for`-Schleifen wird als bekannt vorausgesetzt. Für die Erläuterung weiterer programmiertechnischer Feinheiten sowie die Implementierung der grafischen Benutzeroberfläche (GUI) wird einerseits auf die ausführliche Dokumentation von MATLAB verwiesen, andererseits bietet auch das praktisch orientierte Buch von STEIN [Ste12] eine gute Übersicht über die Funktionsweisen.

2.3 Notation: BPMN 2.0

Um die komplexen Vorgänge der Realität darzustellen und ein Funktionsmodell aufzustellen, reichen herkömmliche Flussdiagramme nicht mehr aus bzw. sind bei einer hohen Komplexität nicht mehr leicht nachzuvollziehen. Daher wird in dieser Arbeit eine Darstellungsart aus der Modellierung von Geschäftsprozessen benützt: das BPMN.

BPMN stand in der ersten Version von 2004 für „Business Process Modeling Notation“ und wurde von der Business Process Management Initiative (BPMI) veröffentlicht. Sie hatte es sich zum Ziel gesetzt, eine standardisierte, grafische Prozessnotation zu definieren und einem großen Publikum zur Verfügung zu stellen. Mit der Übernahme der BPMI durch die Object Management Group (OMG) 2005 begann sich der Standard weltweit durchzusetzen. Die aktuelle Fassung BPMN 2.0 wurde 2011 veröffentlicht, wobei sich die Bedeutung von BPMN zu „Business Process Model and Notation“ änderte, um der formalen Definition eines Metamodells Rechnung zu tragen. Die Spezifikation von BPMN 2.0 wird kostenlos in englischer Sprache zur Verfügung gestellt [OMG11]. Darin befinden sich alle BPMN-Symbole mit ihren Bedeutungen und Anwendungsregeln [vgl. FR12, S. 8-10].

In der Folge soll ein kurzer Überblick über die Grundfunktionen und die generelle Funktionsweise gegeben werden; zusätzlich dazu werden alle in dieser Arbeit auftretenden Symbole als Übersicht dargestellt und ihre Funktion erläutert. In Abbildung 2.1 sind die Basiselemente des BPMN abgebildet:

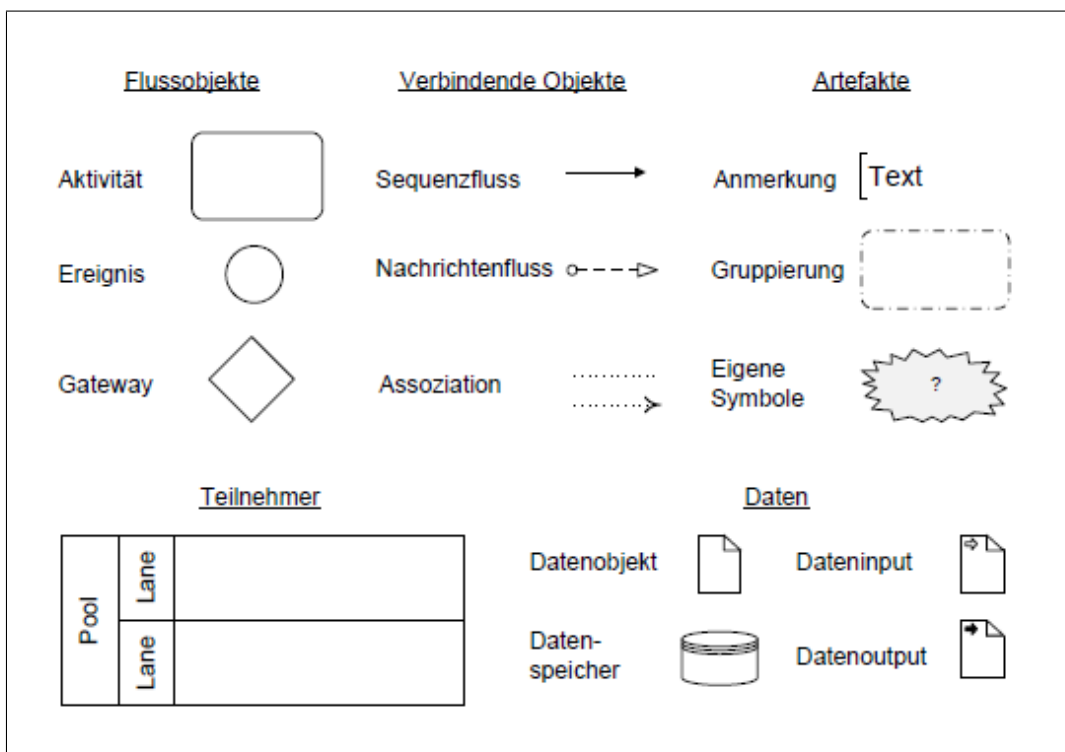


Abbildung 2.1: Basiselemente des BPMN¹

¹[FR12, S. 21]

Das Modell beschreibt Vorgänge, die ablaufen (*Aktivitäten*), unter der Bedingung (*Gateway*), dass gewisse Dinge passieren (*Ereignis*). Die logische Abfolge in einer Instanz (*Pool*) wird durch den Sequenzfluss dargestellt, zwischen zwei Instanzen wird über den Nachrichtenfluss kommuniziert. Ebenfalls lassen sich damit die Erzeugung, Verarbeitung und Ablage von Daten darstellen [vgl. FR12, S. 21-22].

Nachdem nun die prinzipielle Systematik definiert wurde, werden in Abbildung 2.2 die speziellen verwendeten Symbole gezeigt:

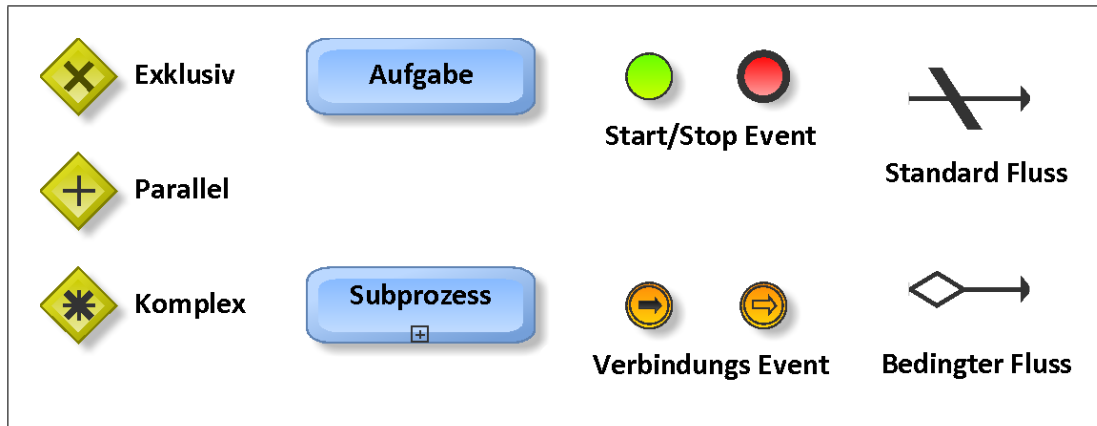


Abbildung 2.2: Spezielle Elemente des BPMN

Bei den Gateways werden folgende verwendet:

- **Exklusiv:** Dieses Gateway steht für eine Oder-Entscheidung, es kann also nur einer der folgenden Fälle eintreten.
- **Parallel:** Dieses Gateway symbolisiert eine Und-Entscheidung, bei einem Split (Aufspaltung) werden also alle Prozesse gestartet, bei einem Merge (Zusammenführung) wird der Prozess erst weiter geführt, wenn alle eingehenden Prozesse beendet wurden.
- **Komplex:** Dieses Gateway stellt eine komplexe Verbindung mehrerer Entscheidungen dar und wird in dieser Arbeit für die Definition einer komplizierten Schleife verwendet.

Bei den Aktivitäten wird zusätzlich zur normalen Aufgabe noch der Subprozess eingeführt. Dieser wird in der Folge dazu benützt, um Funktionen in Übersichtsdarstellungen zu kennzeichnen, deren Logik explizit an einem anderen Punkt der Arbeit besprochen wird. Folgende Events treten auf:

- **Start/Stop:** Jeder Prozess beginnt mit einem Startevent und endet mit einem Stopevent. Jedem Zweig des Modells muss ein eigenes Ende zugewiesen werden.
- **Verbindungen (Links):** Diese Events dienen dazu, einen Prozess zu unterbrechen (Throw) und an einer definierten anderen Stelle (Catch) weiterzuführen. Diese Darstellungsweise wird bei Funktionen mit einer komplexen Schleifenstruktur genützt.

Neben dem normalen Sequenzfluss werden noch zwei weitere Typen von Flüssen eingesetzt:

- **Bedingter Fluss:** Die Darstellung einer Entscheidung lässt sich auch platzsparend ohne Gateways und Aktivitäten darstellen. Hier wird die Bedingung, die zutreffen muss, damit dieser Sequenzfluss eintritt, über der Flusslinie positioniert.
- **Standard Fluss:** Diese Darstellung erlaubt es, einen Fluss festzulegen, der alle Fälle bedient, welche nicht explizit durch einen Bedingten Fluss festgelegt wurden.

Die hier abgebildeten Prozesse nutzen nur einen Bruchteil der von BPMN zur Verfügung gestellten Methoden und Modellierungswerkzeugen aus. Für eine genaue Beschreibung dieser Tools wird auf die Literatur verwiesen, für eine übersichtliche Darstellung der gesamten Notation wird [BPM11] empfohlen.

Zur Erstellung der in dieser Arbeit abgebildeten BPMN Modelle wurde die freie Software ARIS Express der Firma „Software AG“ verwendet.

3 Erarbeitung eines Modells

In diesem Kapitel werden die Abläufe und Vorgänge in realen Systemen beschrieben und analysiert. Mit den in Abschnitt 1.2 definierten Anforderungen an die Simulation und den aus der Analyse gewonnenen Informationen werden wichtige Annahmen über das System abgeleitet und deren Auswirkungen besprochen. Abschließend wird ein Modell erarbeitet, welches allen angestrebten Eigenschaften und Annahmen genügt. Dieses Modell wird in der Folge als Grundlage für die Simulation verwendet.

Um eine bessere Übersicht in weiteren Abbildungen zu ermöglichen, wird das in Abbildung 3.1 dargestellte Farbschema einheitlich verwendet.

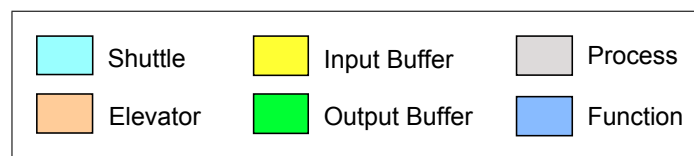


Abbildung 3.1: Farblegende

Die hellblaue Farbe wird bei der Darstellung von Vorgängen des Shuttles angewendet. Dies gilt in gleicher Weise für die Farbe Orange und die den Lift betreffenden Prozesse. Die Einlagerungs- bzw. Auslagerungspuffer werden in gelb respektive grün dargestellt. Werden allgemeine Prozesskomponenten benötigt, so werden diese in grau angezeigt, allgemeine Funktionen, die erst in der Simulation auftreten, sind blau.

3.1 Analyse der Prozesse

Betrachtet man die Vorgänge, die in einem Shuttle System ablaufen, so ergeben sich drei unterschiedliche Prozesse:

- Einlagerung
- Auslagerung
- Umlagerung

Als erstes soll die Einlagerung betrachtet werden. Abbildung 3.2 zeigt die Systemkomponenten, die eine Lagereinheit durchlaufen muss, um eingelagert zu werden. Dieser Prozess wirkt auf den ersten Blick trivial:

Eine Lagereinheit wird von außen an den Lift übergeben und in der ihr zugewiesenen Ebene im Einlagerungspuffer abgelegt. Von dort aus wird sie von einem leeren Shuttle aufgenommen und zu ihrem Lagerplatz gebracht. Mit der Entladung des Shuttles endet der Prozess und die Lagereinheit wartet nunmehr auf ihre nächste Auslagerung.

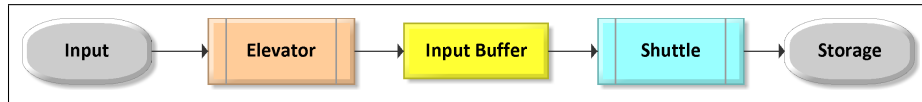


Abbildung 3.2: Prozess: Einlagerung

Die Auslagerung funktioniert, wie in Abbildung 3.3 dargestellt, in sehr ähnlicher Weise: Nach Aktivierung des Auslagerungsvorganges fährt ein Shuttle von seiner aktuellen Position zum Lagerplatz der auszulagernden Einheit. Das Shuttle nimmt sie dort auf und transportiert sie zum Auslagerungspuffer. Von dort aus wird sie vom Lift abgeholt und zur Übergabeposition (Position, an der die Lagereinheiten das System verlassen) gebracht. Mit der Entladung des Liftes endet der Auslagerungsvorgang.

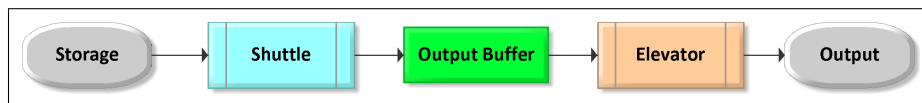


Abbildung 3.3: Prozess: Auslagerung

Der Umlagerungsvorgang ist eine Kombination der beiden vorher beschriebenen Prozesse mit der Besonderheit, dass er nicht nach außen hin wirksam wird. Das bedeutet, dass sich sowohl der Lagerort als auch der Bestimmungsort der umzulagernden Einheit innerhalb des Systems befinden. Ein Umlagerungsvorgang wird dann notwendig, wenn das System im Inneren optimiert werden soll, z. B. : Lagereinheiten werden erwartungsgemäß eine längere Zeit nicht mehr benötigt und daher bei einer geringen Auslastung des Systems in höhere Ebenen und/oder hintere Positionen umgelagert. Befinden sich der Lagerort und der Zielort in verschiedenen Ebenen, so wird zuerst ein Auslagerungsvorgang initiiert, der die Einheit in den Auslagerungspuffer transportiert. Von dort aus wird die Lagereinheit mit dem Lift in die Zielebene gebracht und dort, analog zum Einlagerungsvorgang, an ihrem Zielort wieder abgelegt. In der Folge soll die Möglichkeit der Umlagerung vernachlässigt werden, da eine maximale Auslastung des Systems vorausgesetzt wird und damit keine Kapazitäten mehr dafür zur Verfügung stehen.

Fügt man nun die beiden Prozesse zusammen so ergibt sich der in Abbildung 3.4 ersichtliche Gesamtprozess, der alle Systemkomponenten umfasst, die durchlaufen werden.

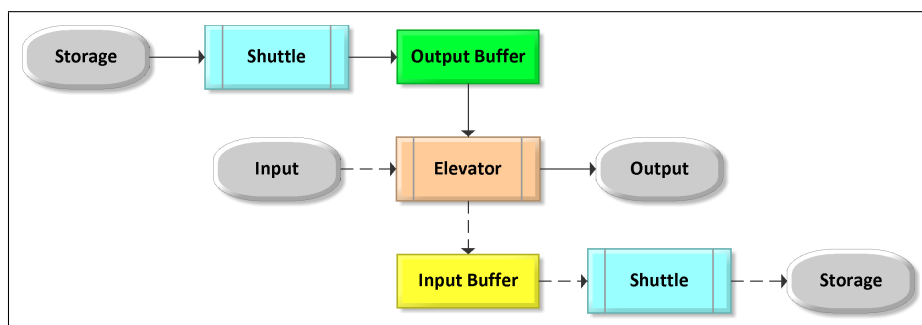


Abbildung 3.4: Kombination der Prozesse

Betrachtet man aber den realen Prozess genauer so wird klar, dass der dargestellte Prozess eine kritische Komponente nicht berücksichtigt: die tatsächliche verstrichene Zeit im System. Es wird nämlich der Eindruck erweckt, dass der Anfangszeitpunkt und der Endzeitpunkt der einzelnen Prozesse die Gesamtzeit widerspiegeln. Dies ist aber keineswegs der Fall, wenn man bedenkt, dass sich die Shuttles in ihren Ebenen völlig unabhängig vom Lift bewegen können. Durch diese Tatsache bedingt, können also in allen mit einem Shuttle versehenen Ebenen zeitgleich Prozesse stattfinden.

Da sich in einer Ebene nur ein Shuttle befindet, kann also auch nur ein Prozess stattfinden. Wurde durch die Beladung des Einlagerungspuffers ein Einlagerungsprozess gestartet, so endet dieser mit der Entladung des Shuttles an dem definierten Lagerort. Wurde ein Auslagerungsprozess durch Festlegen einer auszulagernden Einheit begonnen, so wird er durch das Befüllen des Auslagerungspuffers beendet. Nachdem diese Teilprozesse beendet sind, kann das Shuttle den nächsten Auftrag annehmen und einen weiteren Prozess starten. Die zeitliche Abfolge, bezogen auf die Systemzeit, wird aber durch den Lift festgelegt, da dieser nur eine Lagereinheit gleichzeitig transportieren kann. Hier kann also nur entweder ein Einlagerungsvorgang stattfinden, also der Lift bringt eine Einheit von der Übergabestation zum Einlagerungspuffer der Zielebene, oder ein Auslagerungsvorgang, also der Transport einer Einheit vom Auslagerungspuffer zur Übergabestation. Somit ist, in erster Näherung, die nach außen wirksame Zeit (Systemzeit) allein von der Zeit abhängig, die der Lift zum Transport benötigt. Dies gilt natürlich nur dann, wenn bei einem Einlagerungsvorgang der Einlagerungspuffer frei ist respektive bei einem Auslagerungsvorgang der Auslagerungspuffer schon befüllt ist. Diese Aspekte müssen sich auch im Modell wiederfinden.

Die wichtigen Systemkomponenten, die auch in einem Modell abgebildet werden müssen sind also:

- Lift(e)
- Shuttle(s)
- Puffer für die
 - Einlagerung
 - Auslagerung

Schon alleine mit diesen Komponenten lassen sich sehr viele verschiedene Systeme abbilden. Hier einige mögliche Varianten:

- System bestehend aus zwei Liften, die jeweils nur einlagern oder auslagern, dafür stehen pro Ebene ein Einlagerungs- und ein Auslagerungspuffer bereit.
- System bestehend aus einem Lift, sowohl einem Einlagerungs- und einem Auslagerungspuffer pro Ebene.
- System bestehend aus einem Lift und nur einem Puffer pro Ebene, der situationsbedingt entweder zur Ein- oder Auslagerung verwendet wird.

3.2 Modell der Prozessabläufe

Um die Prozessabläufe modellieren zu können, muss zuerst das Gesamtsystem definiert werden, auf das sich diese beziehen. Vergleicht man nun die Eigenschaften der in Abschnitt 3.1 erläuterten Varianten mit den in Abschnitt 1.2 festgelegten Anforderungen an die Simulation, so scheiden Systeme mit zwei Liften aus, da die Darstellung des geforderten Doppelspielzyklus nicht möglich ist. Um der Forderung nach einem realitätsnahen Modell zu entsprechen, scheidet auch die Variante mit nur einem Puffer aus, da die geometrischen Gegebenheiten die Anbringung von zwei Puffern stark favorisieren. Die zweite Variante ist also diejenige, die den Anforderungen am besten gerecht wird und deshalb ausschließlich betrachtet wird.

Abbildung 3.5 zeigt schematisch den Aufbau der gewählten Variante (fünf Ebenen mit fünf Lagerpositionen pro Ebene pro Seite). Der Aufriss wird aufgeklappt dargestellt, wobei die rechte Seite des Ganges nur angedeutet wird, da sich die Shuttles an denselben Positionen befinden. Die Pfeile geben die möglichen Bewegungsrichtungen an.

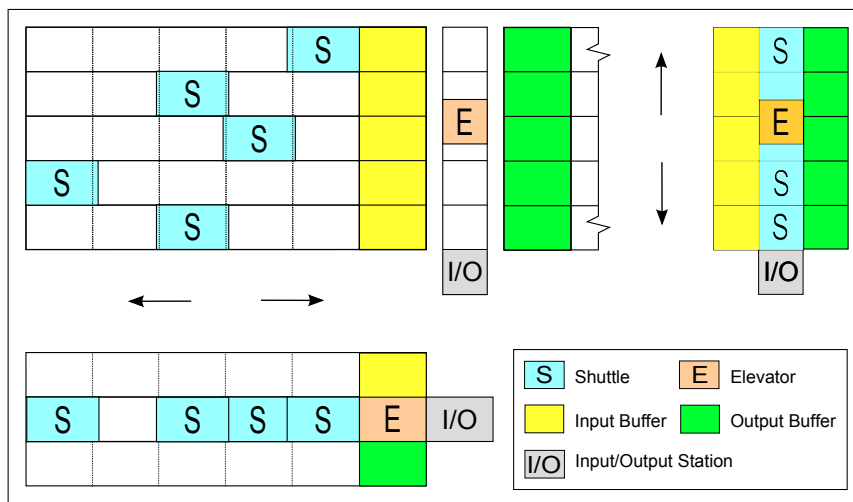


Abbildung 3.5: Aufbau des Shuttle Systems

Außerdem wurde bis jetzt angenommen, dass die Puffer bei Ankunft des Liftes schon den gewünschten Zustand haben. Diese Annahme ist aber unzulässig, da gerade in diesen Situationen Wartezeiten auftreten, die sich direkt auf die Systemzeit auswirken.

Abbildung 3.6 zeigt nun ein detailliertes Modell, das alle Anforderungen, die zuvor definiert wurden, hinreichend erfüllt. Im ersten Pool befinden sich die notwendigen Schritte, um einen Doppelspielzyklus in Systemzeit abzuwickeln. Daneben befindet sich der Pool, der dem Lift zugeordnet ist und dessen Aktionen während eines Zyklus repräsentiert. Der letzte Pool steht für das Regal an sich, welches sich in die zwei Lanes Shuttles und Puffer teilt. In diesen wird noch zwischen Einlagerung und Auslagerung unterschieden. Die waagrechteten Linien zwischen den Pools symbolisieren den Nachrichtenfluss zwischen den einzelnen Systemkomponenten. In der Folge werden die einzelnen Schritte genauer erläutert:

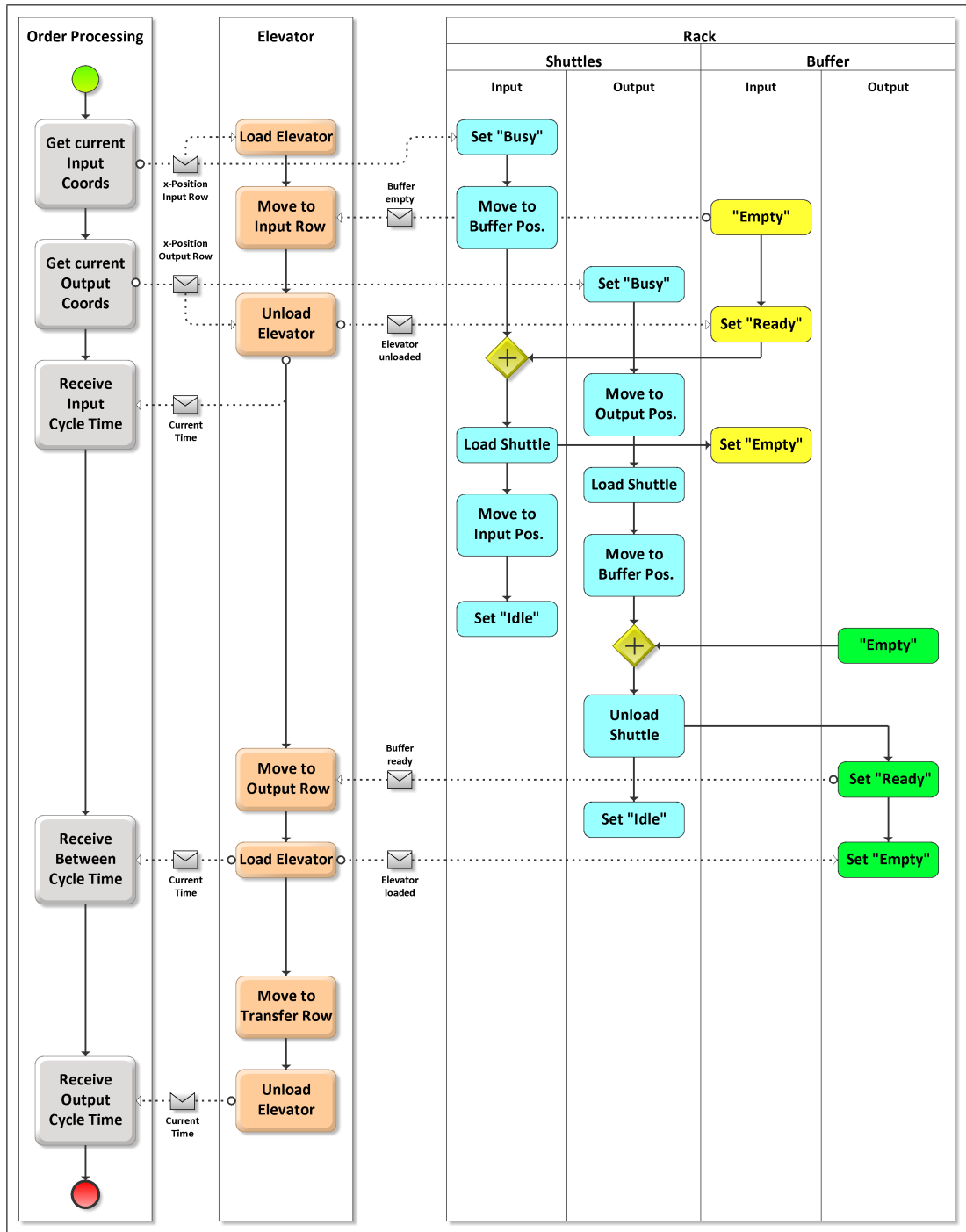


Abbildung 3.6: Modellierung der Prozessabläufe

Koordinaten für die Einlagerung: Die Position und die Ebene des Lagerplatzes werden an den Lift übergeben. Nach dem Ende des Ladevorganges in der Übergabeebene bewegt sich der Lift in die Zielebene, das Shuttle in der betreffenden Ebene wird aktiviert und bewegt sich zur Position des Puffers.

Koordinaten für die Auslagerung: In der Auslagerungsebene wird das Shuttle aktiviert, bewegt sich zur auszulagernden Einheit, wird beladen und kehrt zurück zur Position des Puffers. Sobald das Shuttle wieder entladen ist, steht es für den nächsten Auftrag bereit; sollte keiner anstehen, wird es in den Ruhemodus versetzt.

Berechnung der Einlagerungszeit: Sobald der Lift in der Einlagerungsebene angekommen ist und die Information vom Puffer bekommt, dass dieser leer ist, wird die Lagereinheit in den Puffer abgelegt. Dabei wird die aktuelle Systemzeit als Einlagerungszeit festgehalten. Sobald der Puffer gefüllt ist und das Shuttle an der Pufferposition angekommen ist, wird das Shuttle geladen und bringt die Einheit zur Zielposition. Erst wenn das Shuttle wieder leer ist, kann es einen weiteren Auftrag annehmen oder wird in den Ruhemodus versetzt. Dieser Prozess wird im Hintergrund ausgeführt, während der Lift schon einen anderen Zyklus durchläuft.

Berechnung der Zwischenzeit: Nach dem Entladen der Lagereinheit bewegt sich der Lift zur Auslagerungsebene. Dort angekommen, kann er erst wieder beladen werden, wenn der Puffer die Information sendet, dass er gefüllt ist. Dies ist dann der Fall, wenn das auslagernde Shuttle die aktuell transportierte Einheit im Puffer abgeladen hat. Mit dem Leeren des Auslagerungspuffers wird die vergangene Zeit als Zwischenzeit bezeichnet.

Berechnung der Auslagerungszeit: Der beladene Lift bewegt sich nun wieder zurück zur Übergabeebene. Nach der Entladung des Liftes verlässt die Einheit das System und die Auslagerungszeit wird gemessen.

Addiert man nun die Einlagerungs-, Zwischen- und Auslagerungszeit, so erhält man die tatsächlich verstrichene Doppelspielzeit für den aktuellen Zyklus resultierend aus den entsprechenden Koordinaten. Die Doppelspielzeit kann sich aus zwei Komponenten zusammensetzen:

- Minimalzeit (obligat) bestehend aus:
 - Liftzeiten: Zeiten, die aus der Bewegung des Liftes in die verschiedenen Ebenen resultieren
 - Transferzeiten: Zum Be- und Entladen der Systemkomponenten Lift und Puffer nötige Zeiten
- Wartezeiten (optional) bestehend aus:
 - Wartezeit der Einlagerung: Tritt auf, wenn sich der Lift in der Einlagerungsebene befindet, bevor der Einlagerungspuffer geleert wurde.
 - Wartezeit der Zwischenzeit: Tritt auf, wenn sich der Lift in der Auslagerungsebene befindet, bevor der Auslagerungspuffer gefüllt wurde.

Wird nun die Spielzeit für eine ausreichende Anzahl an Doppelspielzyklen berechnet und danach gemittelt, so entspricht das Ergebnis der repräsentativen mittleren Doppelspielzeit für die gewählten Systemparameter.

Vergleicht man nun Abbildung 3.4 mit Abbildung 3.6 so wird deutlich, dass das detaillierte Modell der Prozessabläufe schon in vielen Punkten der Komplexität des realen Systems entspricht, insbesondere was das Auftreten von Wartezeiten und die Aufteilung in nach außen wirksame Prozesse und Hintergrundprozesse angeht. Eine Problemstellung, die in diesem Modell noch nicht gelöst werden konnte, bezieht sich auf den zeitlichen Ablauf des Prozesses:

Würde sich das System genauso verhalten wie in Abbildung 3.6 beschrieben, wären stets maximal zwei Shuttles, eines zur Aus- und eines zur Einlagerung, in Bewegung, da die Aktivierung der Shuttles im selben Zyklus stattfindet, in dem die Lagereinheiten auch vom Lift transportiert werden. Dies würde dazu führen, dass das System nur sehr schwach ausgelastet wäre und es zu sehr hohen Wartezeiten kommt. Dieses Problem wird durch die Einführung des Parameters „aktive Shuttles“ gelöst und wird in Abschnitt 4.5 (S. 24) ausführlich dokumentiert.

3.3 Annahmen zum Aufbau der Simulation

Um nun unter Zuhilfenahme des Modelles der Prozessabläufe und unter Berücksichtigung der Gegebenheiten des realen Systems eine Simulation erstellen zu können, müssen, wie in der Übersicht in der Tabelle 3.1 dargestellt, einige vereinfachende Annahmen getroffen werden, die in der Folge erklärt und begründet werden.

	Reales System	Annahmen
Lagerverwaltung	chaotisch	chaotisch
Koordinaten Einlagerung	definiert	zufällig
Koordinaten Auslagerung	definiert	zufällig
Reihenfolge	optimiert	zufällig
Lagerpositionen	diskret	kontinuierlich
Anzahl der Ebenen	diskret	diskret
Geschwindigkeitsverlauf	stoßoptimiert	trapezförmig

Tabelle 3.1: Vergleich: Reales System und Annahmen

1) Lagerverwaltung

Sowohl im realen System als auch in den Annahmen wird ein chaotisches Lagersystem verwendet, was bedeutet, dass die Wahrscheinlichkeit einer Ein- bzw. Auslagerung an jedem Lagerort gleich groß ist.

Im realen System wird die Verwaltung der Lagerplätze jedoch von einer Lagerverwaltungs-Software gesteuert, sodass im Hintergrund ein gewisser Optimierungsprozess stattfindet. Dieses Verhalten wird in der Simulation nicht dargestellt.

2) Koordinaten Einlagerung

In der Realität kann natürlich bei einer Einlagerung nur ein Lagerort vergeben werden, der zu diesem Zeitpunkt gerade leer steht. Die Auswahl aus den verschiedenen Möglichkeiten trifft wieder das Lagerverwaltungssystem, welches den optimalen

Platz findet, also z. B. eine Ebene, die sich in der Nähe der nächsten Auslagerung befindet und in der das Shuttle momentan keinen aktiven Auftrag hat, damit sofort mit dem Einlagerungsprozess begonnen werden kann. Dieses hochkomplexe Verhalten kann durch die Simulation nicht dargestellt werden. Stattdessen werden die Koordinaten für die Einlagerung zufällig generiert und dann Zyklus für Zyklus abgearbeitet.

3) **Koordinaten Auslagerung**

Eine Auslagerung kann nur dort stattfinden, wo sich auch tatsächlich der geforderte Inhalt befindet. Die Reihenfolge der Auslagerungsvorgänge wird aber von den Bestellungen von außen diktiert. Da die Sequenzierung priorisiert wird, also die Reihenfolge, in denen die Lagereinheiten das System verlassen, ein entscheidender Faktor für das Zusammenspiel des Gesamtsystems ist, sind die Möglichkeiten der Optimierung durch Auswahl eines geeigneten Lagerortes wesentlich geringer.

Das Simulationsverhalten funktioniert wie in Punkt 2 erläutert, entspricht also einer vollkommen zufälligen Abfolge an Koordinaten, die strikt abgearbeitet wird. Da die Sequenzierung unbedingt eingehalten werden muss, stellt die getroffene Vereinfachung das reale System hinreichend gut dar.

4) **Reihenfolge**

Wie in den Punkten 2 und 3 besprochen, wird die Reihenfolge der Ein- und Auslagerungen durch die aktuellen Bestell- und Retourenaufträge festgelegt. Das Lagerverwaltungssystem kann in gewisser Weise optimierend eingreifen und z. B. bei einer höheren Anzahl an Bestellungen mehrere Auslagerungen hintereinander ausführen ohne dazwischen eine Einlagerung vorzunehmen bzw. vice versa.

Die Simulation soll eine reine Abfolge von Doppelspielzyklen sein, die Reihenfolge der Vorgänge ist also nur abhängig von den zufällig generierten Lagerkoordinaten.

5) **Lagerpositionen**

In einem realen System ist die Anzahl der Lagerplätze pro Ebene bestimmt durch die horizontale Teilung des Regals und die Breite der Lagereinheiten. Im Verwaltungssystem wird genau festgehalten ob eine Position frei oder belegt ist und welchen Inhalt eine Lagereinheit hat.

Da keine Überwachung der Lagerorte stattfindet und die Variation der Breite einer Lagereinheit keine Auswirkung auf die Ergebnisse haben soll, werden die Lagerpositionen kontinuierlich vergeben; der Minimalwert liegt bei Null, der Maximalwert ist mit der Länge des Regals bestimmt.

6) **Anzahl der Ebenen**

Die Anzahl der Ebenen stimmt genau mit der Anzahl der Shuttles überein, da sich immer ein Shuttle in jeder Ebene befindet. Hier würde eine kontinuierliche Vergabe der Koordinaten keinen Sinn machen, da dies in einer unendlichen Anzahl an Shuttles resultieren würde.

7) **Geschwindigkeitsverlauf**

Die Beschleunigungsvorgänge in der Wirklichkeit werden genau geregelt und so weit wie möglich ruckfrei ausgeführt, obwohl die beschleunigten Massen verhältnismäßig niedrig sind.

In der Simulation werden ausschließlich trapezförmige Geschwindigkeitsverläufe

verwendet. Da diese bei ausreichend großer Beschleunigung des Liftes schon bei moderaten Höhen des Systems erreicht werden und der Weg der Shuttles durch die kontinuierliche Vergabe der Koordinaten verschwimmt, soll die Möglichkeit von dreieckigen Verläufen außer Acht gelassen werden.

4 Aufbau der Simulation

In diesem Kapitel werden die Vor- und Nachteile grundlegender Methoden der Simulation besprochen sowie die Auswahl der am besten für die Problemstellung geeigneten Variante und deren Optimierung erklärt. Des Weiteren wird das Simulationsmodell so erweitert, dass sich das Problem der zeitlichen Abläufe lösen lässt. Den einzelnen Systemkomponenten werden Entitäten zugeordnet und die Werte, die diese annehmen können, werden erläutert.

4.1 Simulationsmethodik

Um eine Lösung für die mittlere Doppelspielzeit eines Shuttle Regal Systems zu berechnen, gibt es mehrere denkbare Ansätze:

- analytisch
- numerisch-kontinuierlich
- eventbasiert

Im Rahmen der Vorarbeit zu dieser Diplomarbeit wurden einige analytische Ansätze verfolgt, um auf diesem Wege eine repräsentative Lösung zu erhalten. Um die Komplexität des Systems nur annähernd darstellen zu können, wären elementare Vereinfachungen notwendig, die die Aussagekraft zu stark beeinflussen würden.

Ein weiterer Ansatz wäre die numerische Simulation: Für jeden Zeitschritt Δt wird ein Abbild des Systems erstellt und bei jedem Objekt werden die aktuelle Position und der aktuelle Status überprüft. Tritt in diesem Zeitschritt eine Bedingung in Kraft, die zuvor definiert wurde, so werden alle betreffenden Objekte aktualisiert und eventuell ein entsprechender Wert abgespeichert (z. B. Zykluszeit). Diese Methode ist zwar sehr genau, wenn Δt sehr klein ist, führt aber dazu, dass eine große Anzahl an Überprüfungen in jedem Δt notwendig ist, wobei verhältnismäßig selten eine Bedingung eintritt. Das bedingt einen hohen Zeitaufwand für eine einzelne Simulation, die die erhöhte Genauigkeit nicht rechtfertigen würde.

Die letzte Variante richtet sich nach einem eventbasierten Schema. Dabei werden die Bedingungen nicht, wie bei der kontinuierlichen Analyse, bei jedem Zeitschritt überprüft, sondern es wird ein Ereignisplan erstellt, der zu jedem Objekt den Zeitpunkt des nächsten Events, also diejenige Zeit, an der eine Bedingung eintritt und der nächste Schritt bestimmt werden muss, zuordnet. Diese Methode ist sehr effizient, wenn alle Ereignisse vorhersehbar sind, da die Anzahl der Überprüfungen sich dadurch minimiert.

4.2 Optimierung der eventbasierten Struktur

Vergleicht man nun die Charakteristik des in Kapitel 3.2 erarbeiteten Modells mit den in Kapitel 4.1 beschriebenen Methoden, zeigt sich schnell, dass die Anwendung einer eventbasierten Simulation sehr viele Vorteile mit sich bringt und sich das Modell daher sehr präzise darstellen lässt. Die Voraussetzungen für eine ereignisbasierte Simulation sind dadurch gegeben, dass sich für jeden auftretenden Basisprozess schon im Vorhinein die Zeit berechnen lässt, zu der dieser Prozess endet. Bekommt z. B. ein Shuttle den Befehl, eine Lagereinheit auszulagern, lässt sich die Zeit, bis sich das Shuttle von der aktuellen Position zum Ziel bewegt hat und beladen ist, einfach vorab berechnen. Die Basisprozesse werden niemals während des Ablaufes unterbrochen und Entscheidungen über den nächsten Schritt werden immer erst am Ende eines Prozesses getroffen.

Die Anwendung einer strikt eventbasierten Methode würde also bedeuten, dass jeder Zeitpunkt, an dem eine Änderung an einem einzigen Objekt der Simulation auftritt, in den Eventplan aufgenommen werden muss. In unserem Fall würde das bedeuten, dass beispielsweise die Zeitpunkte, an denen Shuttles beladen werden oder ihre Bewegungsrichtung ändern, auch im Eventplan aufscheinen müssten. Dieser Ansatz ist sehr effektiv für kleine Systeme, bei einer großen Anzahl von Shuttles führt dies aber zu einer hohen Anzahl von Ereignissen, die die Simulation verlangsamen.

Eine der Kernaussagen aus Kapitel 3.2 ist aber, dass die nach außen wirksame Zeit nur aus den Liftzeiten (Minimalzeiten) und den Wartezeiten besteht. Um also die Anzahl der Zeitpunkte zu minimieren, werden primär die Zeitpunkte, an denen der Lift einen Zyklus durchlaufen hat, in den Eventplan aufgenommen. Hintergrundprozesse werden zu einem Endzeitpunkt zusammengefasst: Beispielsweise setzt sich bei einer Auslagerung die Transportzeit (bis eine Lagereinheit im Puffer ankommt) aus den resultierenden Zeiten der Fahrt zum Lagerplatz, der Aufnahme der Einheit und der Fahrt zum Puffer zusammen. An den Stellen, an denen Wartezeiten auftreten können, werden die Zeitpunkte des Eventplanes mit den tatsächlichen Hintergrundprozesszeiten verglichen und damit der tatsächliche Zeitpunkt des Eintritts des Ereignisses bestimmt.

4.3 Von Systemkomponenten zu Entitäten

Wie schon in Kapitel 3.1 festgestellt, gibt es drei Grundkomponenten, die jedes System enthalten muss: Lift, Shuttles und Puffer. Die Simulation verlangt nun weitere Komponenten, um funktionieren zu können: die Koordinaten. In Abbildung 4.1 werden nun alle Komponenten und die daraus abgeleiteten Entitäten als Übersicht dargestellt.

Das **Shuttle** enthält also drei Informationen: den *Status*, den *Modus* und die *Position*.

Der *Status* definiert, ob das Shuttle im Moment einen Auftrag hat [*aktiv*] oder auf den nächsten Auftrag wartet [*inkativ*].

In der *Position* wird die x-Koordinate abgespeichert, an der sich das Shuttle am Ende des aktuellen Auftrages befinden wird.

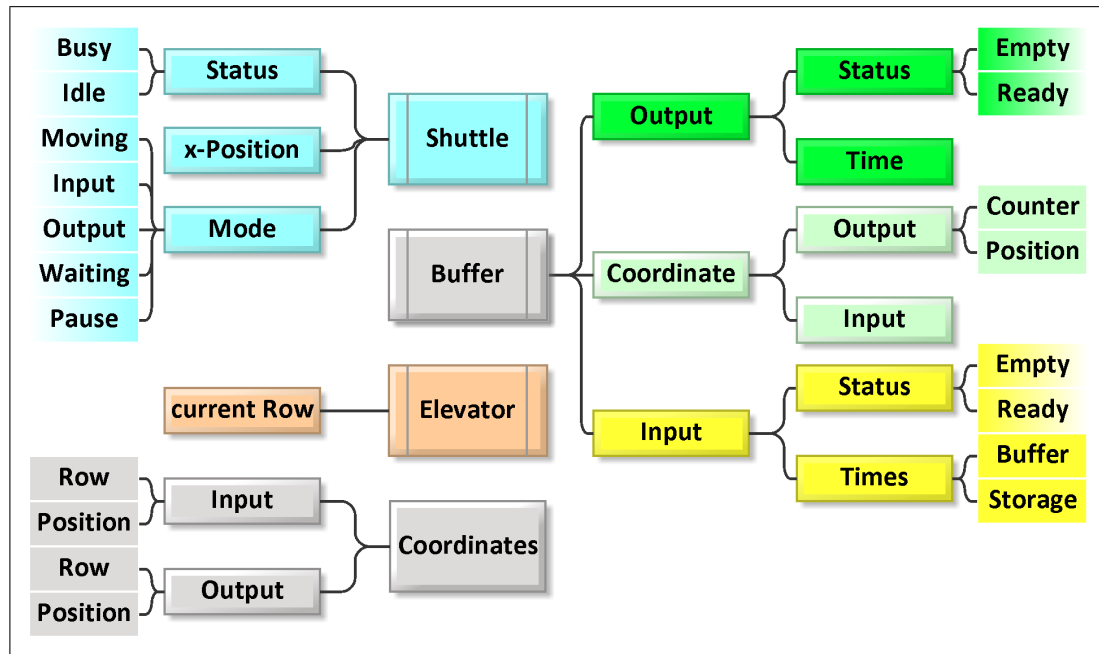


Abbildung 4.1: Entitäten mit Standardwerten

Der *Modus* gibt an, welchen Auftrag das Shuttle zur Zeit bearbeitet; hier gibt es mehrere verschiedene mögliche Werte:

[*bewegen*]: Das Shuttle bereitet sich auf einen Einlagerungsvorgang vor, bewegt sich zum Einlagerungspuffer und wartet darauf, dass dieser [*bereit*] wird.

[*einlagern*]: Das Shuttle führt einen Einlagerungsvorgang durch, fährt von seiner aktuellen Position zum Puffer und bringt dann die Lagereinheit zu seiner Zielposition.

[*auslagern*]: Das Shuttle führt eine Auslagerung durch, fährt von seiner aktuellen Position zur Position des Lagerortes und bringt dann die Lagereinheit zum Puffer.

[*warten*]: Das Shuttle kommt aus dem Modus [*auslagern*] zum Puffer, der aber noch den Status [*bereit*] hat. Das Shuttle muss warten, bis der Puffer [*leer*] wird. Erst anschließend kann das Shuttle weitere Aufträge annehmen.

[*ruhen*]: Das Shuttle hat den Status [*inaktiv*] und ruht, bis es einen neuen Auftrag erhält.

Der **Puffer** lässt sich in drei verschiedene Bereiche gliedern: die *Auslagerung*, die *Einlagerung* und die Speicherung der Koordinaten im *Koordinatenpuffer*.

Die Informationen, die unter *Auslagerung* gespeichert werden, sind einerseits der Status des Puffers und andererseits der Zeitpunkt, an dem der Puffer gefüllt sein wird. Der Status kann nun entweder [*bereit*] sein, wenn sich ein Lagereinheit darin befindet, oder im anderen Fall [*leer*]. Die Zeit wird bei der Aktivierung eines Auslagerungsvorganges berechnet und im Puffer gespeichert.

Die Informationen für die *Einlagerung* sind dieselben wie zuvor erläutert. Beim Speichern der Zeit muss hier aber zwischen zwei Zeiten unterschieden werden: der Zeitspanne, bis der Puffer [*leer*] ist (ab diesem Zeitpunkt kann der Lift wieder eine Lagereinheit im

Puffer ablegen) und dem Zeitpunkt an dem der Einlagerungsvorgang abgeschlossen ist (ab diesem Zeitpunkt ist der Prozess beendet und das Shuttle kann den nächsten Auftrag ausführen).

Der *Koordinatenpuffer* wird notwendig, da dem Shuttle die Position der nächsten Aus- bzw. Einlagerung mitgeteilt werden muss. Da pro Ebene immer nur eine Einlagerung stattfindet (nur eine Einheit kann sich im Puffer befinden), muss nur die Position der nächsten Einlagerung gespeichert werden. Im Gegensatz dazu können in einer Ebene mehrere Auslagerungsaufträge auf ein Shuttle warten, die dann im FIFO-Prinzip abgearbeitet werden müssen. Dazu ist es notwendig, mehrere verschiedene Positionen abzuspeichern und mit einem Zähler sicherzustellen, dass die Reihenfolge eingehalten wird.

Im **Lift** muss ausschließlich die Ebene, in der er sich momentan befindet, abgespeichert werden.

Da die **Koordinaten** der Ein- und Auslagerungsvorgänge und damit ihre Reihenfolge zufällig generiert werden (wie in Abschnitt 3.3 beschrieben), müssen diese ebenfalls zentral abgespeichert werden. Es werden für die Einlagerung und die Auslagerung sowohl die Ebene als auch die Position der Lagereinheit festgelegt.

4.4 Modell der eventbasierten Simulation

Mit den bisher erarbeiteten Grundlagen lässt sich nun aus dem in Kapitel 3.2 abgeleiteten Modell der Prozessabläufe (vgl. Abb. 3.6, S. 15) ein Simulationsmodell für einen Zyklus erstellen wie in Abbildung 4.2 dargestellt.

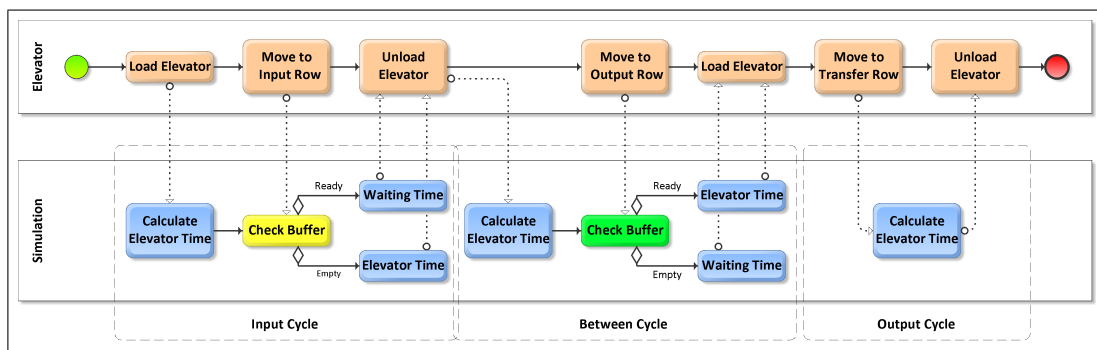


Abbildung 4.2: Ableitung der Simulationszyklen

Im oberen Pool wird der Prozessablauf des Liftes dargestellt, darunter die Schritte, die diesem in der Simulation zugeordnet sind. Die Simulation teilt sich nun in drei klar definierte Abschnitte: den Einlagerungszyklus, den Zwischenzyklus und den Auslagerungszyklus.

Der **Einlagerungszyklus** beginnt mit der Beladung des Liftes in der Ebene der Übergabestelle. Danach wird derjenige Zeitpunkt berechnet, zu dem der Lift in der Zielebene angekommen sein wird. Nun wird der Status des Einlagerungspuffers überprüft, ist er bereits [*leer*], kann der Prozess ohne Unterbrechung fortgesetzt werden. Ist er aber noch

[*bereit*], so kann der Prozess erst fortgesetzt werden, wenn der Puffer [*leer*] ist – es tritt eine Wartezeit auf. Der Zyklus endet mit der Entladung des Liftes.

Der **Zwischenzyklus** startet sofort nach dem Ende des letzten Zyklus mit der Berechnung der Liftzeit, die notwendig ist, um von der Einlagerungs- zur Auslagerungsebene zu kommen. Danach wird der Status des Auslagerungspuffers überprüft, ist dieser schon [*bereit*], wird der Prozess sofort fortgesetzt; ist er noch [*leer*], entsteht eine Wartezeit. Mit dem Beladen des Liftes endet dieser Zyklus.

Der **Auslagerungszyklus** umfasst nun nur mehr die Liftzeit von der Auslagerungsebene zur Ebene der Übergabestelle und die Zeit, die zum Entladen des Liftes notwendig ist.

4.5 Erweiterung des Modells: Die „aktiven Shuttles“

Am Ende von Abschnitt 3.1 wurde die Frage nach einem zeitlich richtigen Ablauf gestellt und bis jetzt noch nicht zufriedenstellend beantwortet. Um die geforderte Gleichzeitigkeit von mehreren Hintergrundprozessen gewährleisten zu können, muss ein weiterer Parameter eingeführt werden: die Anzahl der „aktiven Shuttles“. Abbildung 4.3 stellt die Auswirkungen dieses Parameters auf den Simulationsablauf grafisch dar.

Im linken Pool wird wieder der Ablauf der einzelnen Simulationszyklen analog zu Abbildung 4.2 dargestellt. Im rechten Pool werden die Abläufe der Shuttles detailliert beschrieben, geteilt in die Lanes Einlagerung und Auslagerung. Der Informationsfluss zeigt die logischen Verbindungen zwischen den beiden Pools.

Der wesentliche Unterschied liegt im Vorfeld des Zyklus: wie dargestellt wird nämlich die Auslagerung nicht im aktuellen Zyklus initialisiert, sondern schon eine bestimmte Anzahl an Zyklen zuvor. Diese Anzahl wird als „aktive Shuttles“ im Vorhinein festgelegt und ist während der gesamten Simulation konstant. Die „aktiven Shuttles“ geben damit die maximale Anzahl an Shuttles an, die zur selben Zeit eine Auslagerung vornehmen können.

In Abbildung 4.3 wurde die Anzahl der „aktiven Shuttles“ auf fünf festgelegt, also wurde die aktuelle Auslagerung schon vor fünf Zyklen begonnen: Wenn das Shuttle in der Zielebene zu diesem Zeitpunkt [*inaktiv*] ist, wird sofort die Auslagerungszeit berechnet und im Auslagerungspuffer gespeichert. Ist das Shuttle aber [*aktiv*] und führt einen Auftrag aus, wird der Lagerort dem Koordinatenpuffer hinzugefügt. Dort verbleibt er dann, bis das Shuttle alle vorherigen Aufträge abgearbeitet hat; erst wenn der Auftrag aktiviert wird, kann die Pufferzeit, zusammengesetzt aus der Transportzeit und der aktuellen Systemzeit, entsprechend abgespeichert werden. Sobald die im Puffer gespeicherte Zeit überschritten wurde, wird der Status auf [*bereit*] gesetzt. Die Schnittstelle mit dem aktuellen Zyklus ist wieder die Überprüfung des Status des Puffers; diese kann entweder zu einer Wartezeit führen (falls der Status noch [*leer*] ist, sobald der Lift im aktuellen Zyklus an der Zielebene ankommt) oder im anderen Fall zur Ausgabe der Minimalzeit führen.

Analoge Überlegungen lassen sich auch für die Einlagerung treffen, mit dem Unterschied, dass es nicht zielführend ist, eine große Anzahl an Shuttles mehrere Zyklen vor der tatsächlichen Einlagerung anzufordern. Dies würde zu unnötiger Fahrzeit der Shuttles führen, da ein Auslagerungsauftrag in diesem Zeitraum nicht bearbeitet werden könnte.

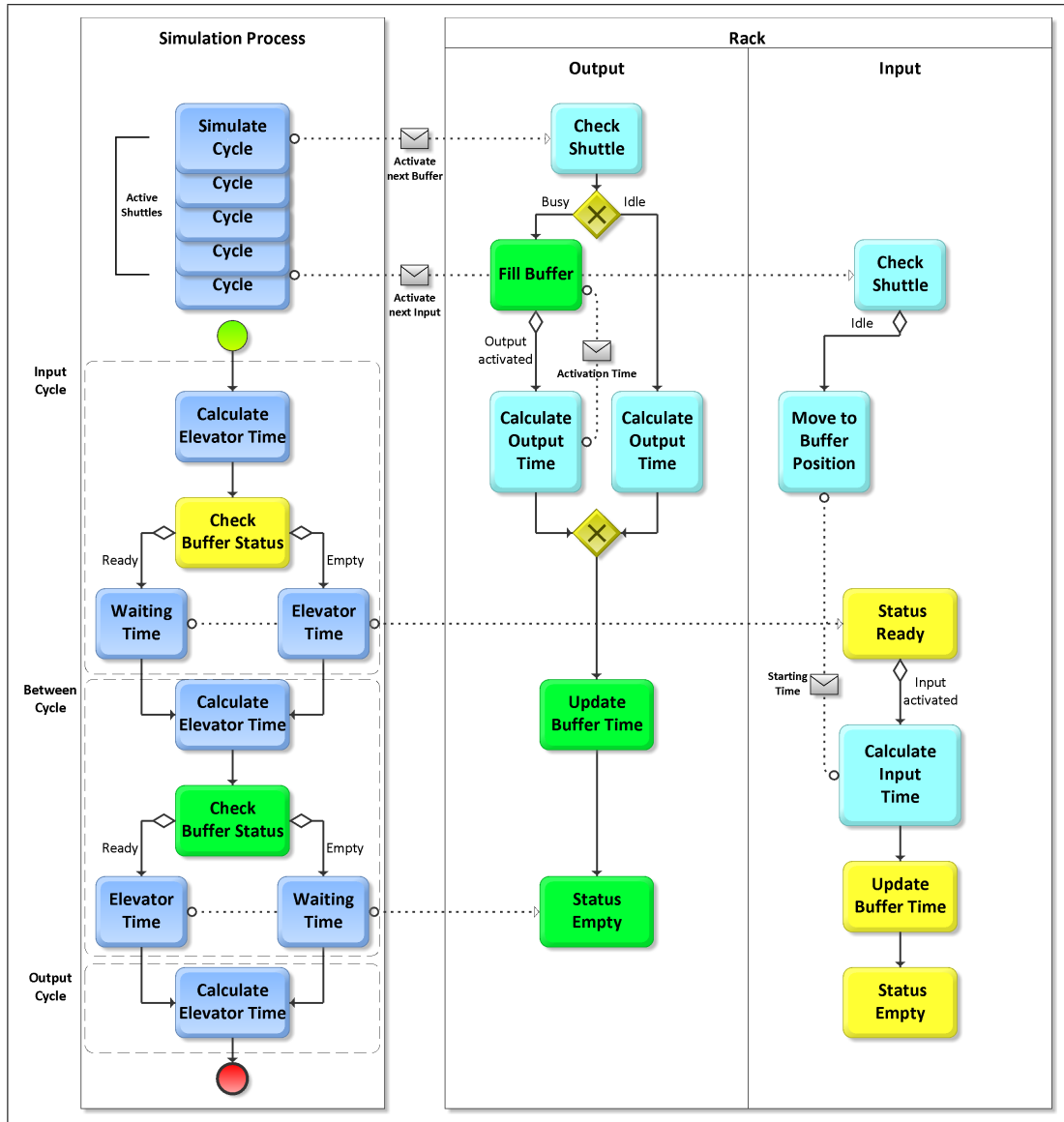


Abbildung 4.3: Ableitung der Simulationszyklen

Daher wird das Shuttle in der Einlagerungsebene einen Zyklus vor der tatsächlichen Einlagerung aktiviert. Falls es [*inaktiv*] ist, wird es in den Modus [*bewegen*] versetzt und fährt zur Pufferposition; dadurch kann der Einlagerungspuffer im aktuellen Zyklus schneller geleert werden und das Shuttle kann früher den nächsten Auftrag ausführen. Ist das Shuttle aber [*aktiv*], so wird der Auftrag im Koordinatenspeicher abgelegt und erst bei einer späteren Abfrage aktiviert.

4.6 Die Grenzen der Simulation

Durch die Einführung der „aktiven Shuttles“ ist es nun möglich, eine große Anzahl an Aufträgen im Voraus zu aktivieren, was in einem gewissen Bereich zu einer Verbesserung der gesamten Performance führt. Setzt man aber einen sehr hohen Wert ein, kann dies zur Instabilität des Modells führen. Betrachtet man folgende Situation, wird das deutlich:

Ein Shuttle hat einen Auslagerungsvorgang durchgeführt und damit wird der Auslagerungspuffer [*bereit*]. Da ein weiterer Auslagerungsvorgang in der Reihenfolge ansteht, wird dieser ausgeführt. In der Zwischenzeit hat der Lift den Auslagerungspuffer noch nicht geleert, aber eine Einlagerung durchgeführt und damit auch den Einlagerungspuffer [*bereit*] gesetzt. Kommt nun das beladene Shuttle am Puffer an, kann es nicht entladen werden und wird in den Modus [*warten*] versetzt. Kommt nun in der Reihenfolge eine weitere Einlagerung des Liftes vor einem Leeren des Auslagerungspuffers, kommt es zu einer sogenannten Deadlock-Situation, da die so zustande gekommene Situation sich nicht mehr auflösen lässt:

Das Shuttle ist mit einer Einheit beladen, die es, per Definition, nur in den Puffer abladen darf, der aber schon gefüllt ist. Der Lift kann seine Ladung, per Definition, nur in den Puffer abgeben, der ebenfalls gefüllt ist und vom Shuttle geleert werden müsste. Dies wird aber von der aktuellen Konfiguration des Shuttles nicht zugelassen.

Dieser Systeminstabilität wird in zweierlei Hinsicht entgegengewirkt:

1. Durch Priorisieren der Einlagerung
Dadurch wird bewirkt, dass das Shuttle, auch wenn in der Reihenfolge der Aufträge anders definiert, immer zuerst den Einlagerungspuffer leert, bevor es eine weitere Auslagerung vornimmt.
2. Durch Wechseln der Ebene
Falls es trotzdem in einer Ebene zu einer Deadlock-Situation kommen sollte, versucht das System auf die nächste Ebene auszuweichen und die Einlagerung dort durchzuführen.

Mit Hilfe dieser Annahmen wird versucht, das System in einem möglichst großen Bereich stabil zu halten.

5 Funktionsweise der Simulation

In diesem Kapitel werden die Erkenntnisse aus den Kapiteln 3 und 4 auf die Erstellung einer Simulation mit dem Softwarepaket MATLAB angewendet. Zuerst wird die Ableitung der einzelnen Simulationsschritte aus dem Prozessmodell definiert und die Zusammenhänge werden erklärt. In weiterer Folge werden alle Funktionen der Simulation diskutiert, und anhand von Beispielen soll der gesamte Ablauf nachvollziehbar gemacht werden. Abschließend soll die Arbeitsoberfläche des Simulationsprogrammes kurz erläutert werden, um die etwaige Benutzung des Paketes zu erleichtern.

5.1 Ablauf der Simulation

Die Abbildung 5.1 zeigt das Zusammenspiel der Benutzeroberfläche (dargestellt im oberen Pool) und der Simulation (dargestellt im unteren Pool); im Zwischenraum wird die Übergabe der notwendigen Informationen symbolisiert.

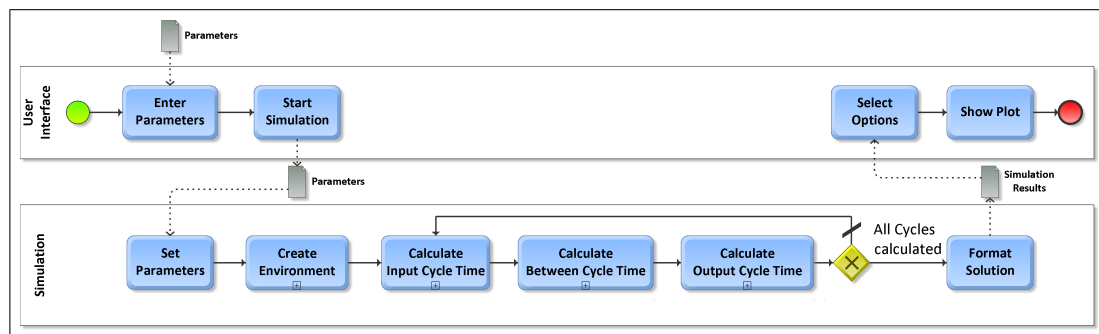


Abbildung 5.1: Funktionsabfolge: Simulationsüberblick

Im ersten Schritt ist eine Eingabe der benötigten Parameter in der Benutzeroberfläche notwendig und die Bestätigung des Starts der Simulation. Die Parameter werden in die Simulationsebene übergeben und dort in den Speicher übernommen. Als nächstes ist es notwendig, die Systemkomponenten zu generieren und mit den Werten für die Ausgangslage zu befüllen. Danach startet die eigentliche Simulation, die sich, wie schon in den Abschnitten 4.4 und 4.5 diskutiert, in die Zyklen für die Einlagerung, die Zwischenzeit und die Auslagerung unterteilt. Diese drei Zyklen sind alle Bestandteil von $[Simulate\ single\ Cycle]_f$ und werden zum besseren Verständnis einzeln diskutiert. Diese Abfolge wird so lange wiederholt, bis die vordefinierte Anzahl an zu simulierenden Zyklen vollständig durchlaufen ist. Als Ergebnis jedes Zyklus wird die Doppelspielzeit – unter Berücksichtigung der ihr zugrunde liegenden Zykluszeiten zwischengespeichert; die Systemzeit läuft aber

unverändert weiter.

Am Ende werden die Ergebnisse der einzelnen Zyklusberechnungen zusammengefasst und einheitlich formatiert. Diese Ergebnisse werden dann an die Benutzeroberfläche übergeben, dort können weitere Optionen gewählt werden und die Daten können sofort grafisch ausgewertet werden, wenn das gewünscht wird.

5.1.1 Die Parameter

Die Parameter, die in die Eingabemaske eingegeben und an die Simulation übergeben werden, lassen sich in drei Kategorien unterteilen: Die geometrischen Parameter, die Systemparameter und die Simulationsparameter. Die Tabellen 5.1, 5.2 und 5.3 zeigen eine Übersicht aller nötigen Eingaben.

Symbol	Name	Bedeutung	Einheit
l	Length	Länge des Ganges	m
n_S	Number of Shuttles	Anzahl der Ebenen	
$n_{ÜS}$	Position I/O	Ebene der Übergabestation	
h	Row Height	Höhe einer Ebene	m
w	Column Width	Breite einer Lagerposition	m

Tabelle 5.1: Geometrische Parameter

Es folgen einige Erläuterungen zur Verwendung der Parameter in Tabelle 5.1. Die Länge wird nur bei der Erzeugung der Lagerkoordinaten als Maximalwert benutzt, die Anzahl der Shuttles sowohl für die Koordinatenerzeugung als auch für die Generierung der Systemkomponenten. Wird die Position der Übergabe mit 0 definiert, so befindet sich die Station eine Ebene unter der ersten Shuttleebene. Die Höhe einer Ebene wird einerseits für die Berechnung der Anzahl der Lagerpositionen in einem System benötigt, andererseits muss sie auch als globale Variable zur Verfügung stehen, da der Weg des Liftes direkt mit der Anzahl der zurückgelegten Ebenen und deren Höhe zusammenhängt. Im Gegensatz dazu geht die Breite ausschließlich in die Berechnung der Lagerplätze ein, da die Positionen ja kontinuierlich verteilt werden.

In Tabelle 5.2 werden zwei verschiedene Arten von Parametern festgelegt: die kinematischen Zusammenhänge und die systemabhängigen Übergabezeiten. Beide müssen global abgespeichert werden, da sie von vielen Funktionen aufgerufen werden.

Die kinematischen Zusammenhänge definieren die Geschwindigkeit und Beschleunigung der Shuttles und des Liftes. Die Übergabezeiten legen fest, wie lange das Be- und Entladen der einzelnen Komponenten dauert. In Tabelle 5.2 wird hier nur die logische Abfolge der Einlagerung dargestellt, dieselben Zeiten müssen aber auch für die Auslagerung definiert werden, auch wenn sie sich normalerweise systembedingt nicht unterscheiden.

Tabelle 5.3 stellt die Parameter dar, die sich ausschließlich auf die Simulation beziehen. Hier wird festgelegt, wie viele Zyklen simuliert werden sollen, wie viele „aktive Shuttles“ es geben soll und an welcher Position sich die Shuttles am Beginn der Simulation befinden sollen. Eine Simulationsvariable, die zwar nicht in der Eingabemaske einzustellen ist, der

Symbol	Name	Bedeutung	Einheit
		<i>Elevator - Shuttles</i>	<i>Lift - Shuttles</i>
v_L, v_S	Velocity	Geschwindigkeit	$\frac{m}{s}$
a_L, a_S	Acceleration	Beschleunigung	$\frac{m}{s^2}$
		<i>Transfer Times</i>	<i>Übergabezeiten</i>
$t_{I \rightarrow L}, t_{L \rightarrow O}$	Transfer \leftrightarrow Lift	Übergabestation \leftrightarrow Lift	s
$t_{L \rightarrow P}, t_{P \rightarrow L}$	Lift \leftrightarrow Buffer	Lift \leftrightarrow Puffer	s
$t_{P \rightarrow S}, t_{S \rightarrow P}$	Buffer \leftrightarrow Shuttle	Puffer \leftrightarrow Shuttle	s
$t_{S \rightarrow Lag}, t_{Lag \rightarrow S}$	Shuttle \leftrightarrow Storage	Shuttle \leftrightarrow Lagerposition	s

Tabelle 5.2: Systemparameter

Symbol	Name	Bedeutung	Einheit
n_{sim}	Number simulated Cycles	Anzahl Simulationen	
n_{aS}	Number active Shuttles	Anzahl „aktive Shuttles“	
x_{in}	Initial Shuttle Position	Ausgangsposition	m
t_a	Time actual	Aktuelle Systemzeit	s

Tabelle 5.3: Simulationsparameter

aber dennoch eine zentrale Bedeutung zukommt, ist die aktuelle Systemzeit, an der sich der Fortschritt im Eventplan (vgl. Abschnitt 4.2) ablesen lässt.

5.1.2 Die Umgebung

Die Umgebung wird durch $[Create\ Environment]_f$ mit zwei optionalen Eingabeparametern erzeugt: dem Modus und dem Funktionsschalter. Diese kommen nur bei einer Variation, wie in Unterabschnitt 5.4.2 beschrieben, zum Einsatz. Ist der Modus $[wahr]$, wird derselbe Koordinatensatz nochmals verwendet. Dies macht dann Sinn, wenn auch der Funktionsschalter $[wahr]$ ist, denn dieser dient dazu, die Belegung von ein und derselben Ebene in zwei aufeinanderfolgenden Zyklen aufzuheben. Werden also zwei Simulationen so hintereinander ausgeführt, dass bei der ersten beide Parameter $[falsch]$ sind, bei der zweiten aber $[wahr]$, führt das dazu, dass bei beiden Funktionen dieselbe Reihenfolge an Koordinaten abgearbeitet wird und nur die Doppelbelegungen mit neuen Koordinaten versehen werden.

	Symbol	min	max
Position	x	0	l
Ebene	n	1	n_S

Tabelle 5.4: Werte für die Koordinaten

Position	Status	Modus
x_{in}	$[inaktiv]$	$[ruhen]$

Tabelle 5.5: Werte für die Shuttles

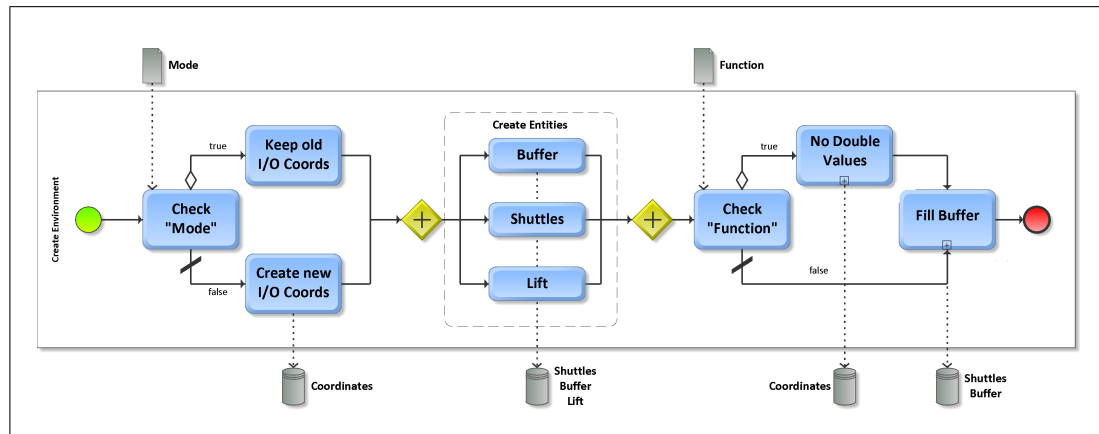


Abbildung 5.2: Funktionsabfolge: Umgebungserstellung

Der Standardweg wird durchschritten, wenn beide Parameter [*falsch*] sind. Dabei werden als erstes alle Koordinaten – nach dem in Tabelle 5.4 gezeigten Schema – für die Ein- und Auslagerung kreiert und zentral gespeichert.

Im nächsten Schritt werden die Systemkomponenten Lift (Ebene der Übergabestation), Shuttles (siehe Tab. 5.5) und Puffer (siehe Tab. 5.6) generiert. Der Auslagerungspuffer enthält nur einen Zahlenwert, nämlich den Zeitpunkt an dem sich die Lagereinheit im Puffer befinden wird (t_{aus}). Der Einlagerungspuffer enthält hingegen zwei Werte: den Zeitpunkt, an dem die Einlagerung abgeschlossen sein wird (t_{ein}), und denjenigen, an dem der Einlagerungspuffer geleert sein wird (t_{buf}). Die Anzahl der Shuttles sowie die der Einlagerungs- und Auslagerungspuffer ist n_S .

	Zeit	Status
Auslagerung	0	[<i>leer</i>]
Einlagerung	0 0	[<i>leer</i>]

Tabelle 5.6: Werte für die Puffer

Im letzten Schritt werden dann die ersten n_{aS} Auslagerungen initiiert und die erste Einlagerung vorbereitet. Damit ändern sich in den aktivierten Ebenen die Werte (wie in den Tabellen 5.7 dargestellt). Die Werte für t_{aus} und t_{buf} werden mit Hilfe von [*Calculate Shuttle Time*]_f errechnet (vgl. Unterabschnitt 5.2.1, S. 34).

<i>Puffer</i>			<i>Shuttles</i>		
	Zeit	Status	Position	Status	Modus
Auslagerung	t_{aus}	[<i>leer</i>]	0	[<i>aktiv</i>]	[<i>auslagern</i>]
Einlagerung	0 t_{buf}	[<i>leer</i>]	0	[<i>aktiv</i>]	[<i>bewegen</i>]

Tabelle 5.7: Werte in den aktiven Ebenen

Abschließend wird noch die Systemzeit t_a mit 0 festgelegt. Dieser Schritt wird nur am Beginn einer neuen Simulation durchlaufen.

5.1.3 Der Einlagerungszyklus

Der Einlagerungszyklus beginnt einmalig nach der Erstellung der Umgebung oder in allen anderen Fällen nach Abschluss der letzten Auslagerung. Als Eingabeparameter für diese Funktion dienen die aktuellen Einlagerungskordinaten und die des nächsten Zyklus, um die nächste Einlagerung aktivieren zu können. Diese werden aus den in $[Create\ Environment]_f$ erzeugten Koordinaten abgelesen, der aktuelle Zyklus n_a dient dazu, die richtige Zeile auszulesen. Der prinzipielle Ablauf lässt sich aus Abbildung 5.3 ablesen.

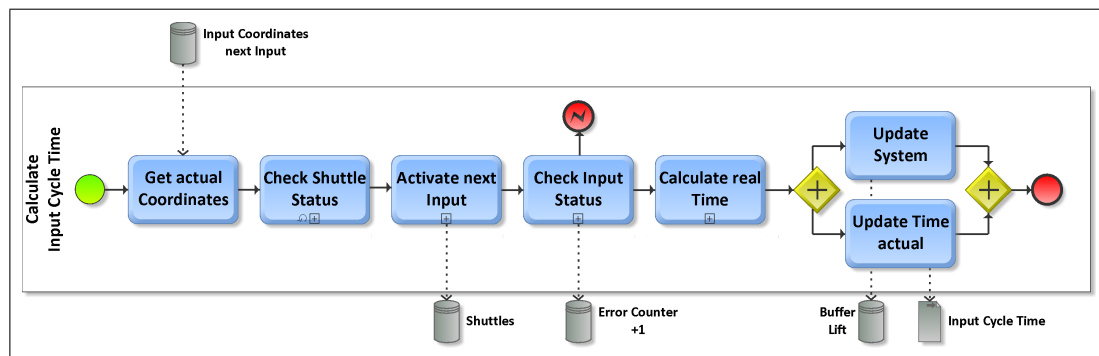


Abbildung 5.3: Funktionsabfolge: Einlagerungszyklus

Im ersten Schritt wird mit $[Check\ Shuttle\ Status]_f$ das gesamte System zum aktuellen Zeitpunkt t_a aktualisiert. Dieser Vorgang ist sehr komplex und wird in Unterabschnitt 5.2.3 ausführlich erläutert. Diese Funktion dient dazu, alle Hintergrundprozesse, die seit dem letzten Ausführen beendet wurden, im System festzuhalten und den Shuttles neue Aufträge zuzuweisen.

Danach wird versucht, die nächste Einlagerung vorzubereiten. Dies ist aber nur dann möglich, wenn das betreffende Shuttle *inaktiv* ist. Der nächste Schritt ist im Sinne der Deadlock-Prävention (vgl. Abschnitt 4.6) notwendig. Hier wird, falls in der Einlagerungsebene eine Deadlock-Situation herrscht, die Möglichkeit geprüft, in eine andere Ebene auszuweichen. Falls dies möglich ist, wird ein Zähler um eins erhöht und gibt so Auskunft über die Häufigkeit des Auftretens einer Deadlock-Situation. Sollte in keiner Ebene eine weitere Einlagerung möglich sein, so wird die Simulation mit einer Fehlermeldung beendet; diese gibt Auskunft über die Anzahl der tatsächlich berechneten Zyklen.

Sobald eine Einlagerung möglich ist, wird $[Calculate\ real\ Time]_f$ aktiviert. Diese Funktion dient dazu, die real vergangene Zeit festzustellen, also ob nur die Minimalzeit oder eine Wartezeit auftritt (vgl. Input Cycle in Abbildung 4.3). Auch dieser Vorgang wird im Abschnitt: Beschreibung der Funktionen (siehe Unterabschnitt 5.2.5) ausführlich beschrieben. Im Zuge dieser Funktion kommt es zu mindestens einem Aufruf von $[Check\ Shuttle\ Status]_f$.

Nachdem die Zeitspanne bis zur Einlagerung, anders gesagt die Einlagerungszeit, berechnet wurde, wird das System aktualisiert. Dabei wird die Position des Liftes auf n_{ein} gesetzt und der Einlagerungspuffer in dieser Ebene auf *[bereit]*, zusätzlich wird die Position x_{ein} im Koordinatenpuffer abgespeichert. Die Einlagerungszeit wird dann in den Ergebnissen gespeichert und schlussendlich zur Systemzeit t_a addiert, um den Vorgang abzuschließen.

5.1.4 Der Zwischenzyklus

Als Eingabeparameter für diesen Zyklus dienen einerseits die aktuellen Auslagerungskordinaten und andererseits wieder die Koordinaten der nächsten Einlagerung. Abbildung 5.4 zeigt die durchlaufenen Funktionen:

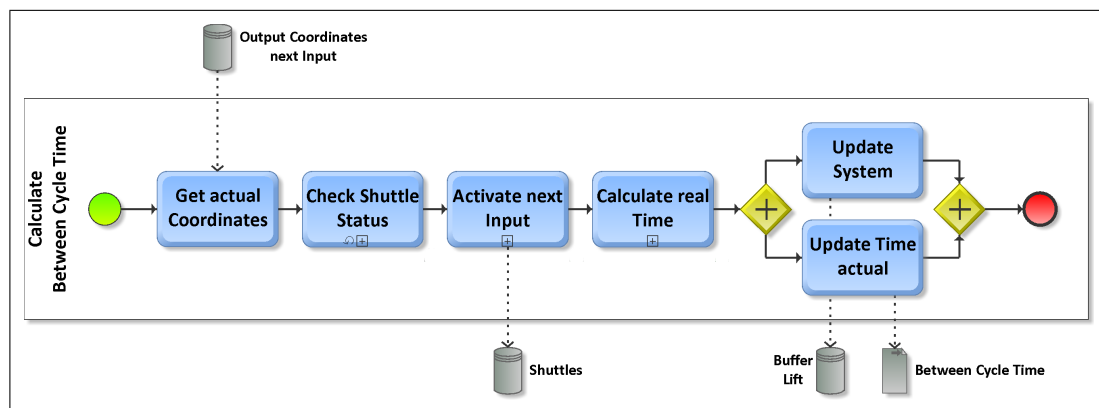


Abbildung 5.4: Funktionsabfolge: Zwischenzyklus

Nach dem Abrufen der aktuellen Koordinaten wird wieder *[Check Shuttle Status]_f* zur Systemzeit t_a aufgerufen, um das System zu aktualisieren. Danach wird erneut überprüft, ob das Shuttle während der verstrichenen Zeitspanne *[inaktiv]* geworden ist, um in diesem Fall die nächste Einlagerung vorzubereiten.

Im Anschluss daran wird wieder, analog zum Einlagerungszyklus, *[Calculate real Time]_f* aufgerufen. Nachdem die Zwischenzeit bestimmt wurde, wird wiederum das System aktualisiert, indem der Auslagerungspuffer *[bereit]* und die Position des Liftes n_{aus} gesetzt werden. Die Zwischenzeit wird im Ergebnis abgespeichert und abschließend zur Systemzeit t_a addiert.

5.1.5 Der Auslagerungszyklus

Der letzte Zyklus dient dazu, den Prozess abzuschließen und die nächste Einlagerung zu initialisieren bzw. die Anzahl der „aktiven Shuttles“ aufrecht zu erhalten. Als Eingabeparameter dienen wie gehabt die Koordinaten der nächsten Einlagerung, außerdem die Koordinaten der n_{as} -Zyklen nach dem aktuellen Auslagerungsvorgang. Die Abbildung 5.5 zeigt die ausgeführten Funktionen.

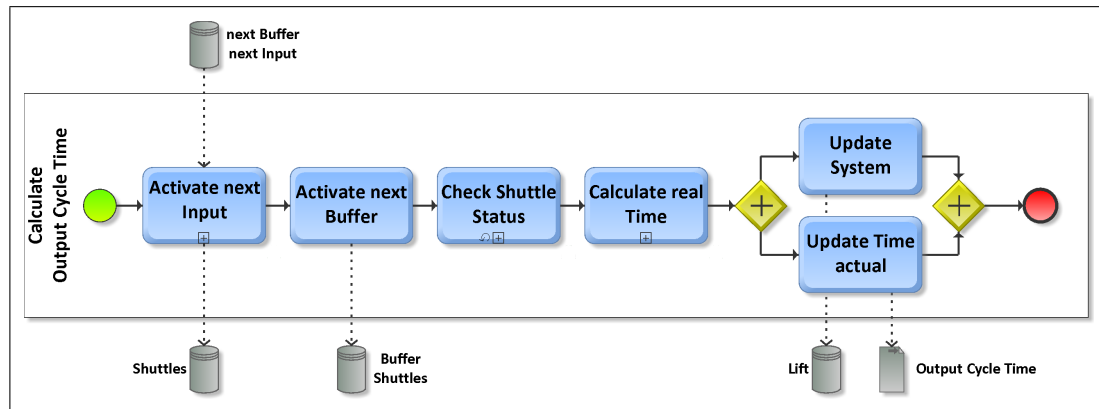


Abbildung 5.5: Funktionsabfolge: Auslagerungszyklus

Die Vorbereitung der nächsten Einlagerung läuft analog zu den vorher besprochenen Funktionen. Die Aktivierung des nächsten Puffers erfolgt nach folgendem Prinzip: Die übergebene Position x_{aus} wird im Koordinatenpuffer der Ebene n_{aus} abgelegt. Dies passiert unabhängig von der derzeitigen Auftragsituation des Shuttles, um die Anzahl der gleichzeitigen Aufträge konstant zu halten. Ist das Shuttle bereits *[aktiv]*, so wird der Beginn des Auftrages einfach nach hinten verschoben und kann erst gestartet werden, sobald der Zähler mit der Position des Auftrags im Koordinatenpuffer übereinstimmt. Sollte das Shuttle derzeit gerade *[inaktiv]* sein, wird es bei der nächsten Aktivierung von *[Check Shuttle Status]_f* auf *[aktiv]* gesetzt und in den Modus *[auslagern]* versetzt; gleichzeitig wird der Zähler um eins erhöht und entspricht damit der Position des aktuellen Auftrags im Koordinatenpuffer.

Die Abfolge der Funktionen *[Check Shuttle Status]_f* und danach *[Calculate real Time]_f* bleibt dieselbe. Danach wird die Position des Liftes auf $n_{ÜS}$ gesetzt, die Auslagerungszeit zum Ergebnis hinzugefügt und anschließend zur Systemzeit addiert.

Damit ist ein kompletter Zyklus durchlaufen. Sofern der aktuelle Zyklus n_a noch nicht die Anzahl der zu simulierenden Zyklen n_{sim} erreicht hat, wird der nächste Zyklus gestartet; dieser Vorgang, sowie die Speicherung des Ergebnisses im Gesamtergebnis wird von *[Simulation]_f* ausgeführt. Außerdem wird, sobald die Differenz aus der Anzahl der zu simulierenden Zyklen n_{sim} und dem Wert des aktuellen Zyklus n_a die Anzahl der „aktiven Shuttles“ n_{aS} erreicht, die Aktivierung des nächsten Puffers gestoppt; andernfalls würde die Anzahl der generierten Koordinaten überstiegen werden.

5.2 Beschreibung der Funktionen

In diesem Abschnitt soll die Arbeitsweise von Funktionen, deren Verständnis für den Ablauf unerlässlich ist, detailliert erläutert werden. Das Zustandekommen von Werten soll auch durch Gleichungen verdeutlicht werden. Die Simulation nutzt außer den hier beschriebenen Funktionen noch eine Reihe anderer Hilfsfunktionen (z. B. für die Transformation einer Struktur in eine Matrix); diese sollen hier aber nicht näher ausgeführt werden,

da ihre Funktionsweise rein technischer Natur ist und nicht im Zusammenhang mit der Logik des Systems steht.

5.2.1 [Calculate Shuttle Time]_f

Die Aufgabe dieser Funktion ist es, den gesamten Hintergrundprozess einer Shuttlebewegung durch die Berechnung der gesamten Zeit dieses Vorganges darzustellen. Dazu müssen die aktuelle Position des Shuttles x_{akt} sowie die Koordinaten des Lagerortes x_{Lag} und der Modus, also ob eine Ein- oder Auslagerung vorliegt, als Eingabeparameter festgelegt werden. Außerdem muss diese Funktion auf die zentral abgespeicherten Werte für die Systemparameter (vgl. Tabelle 5.2) zugreifen können. In Abbildung 5.6 wird der Entscheidungsvorgang erläutert.

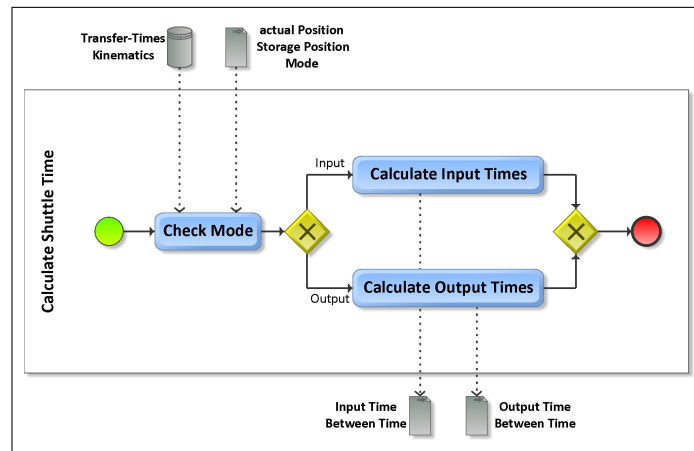


Abbildung 5.6: Funktionsweise: Shuttlezeit Berechnung

Zuerst sollen die Vorgänge bei einer **Einlagerung** erläutert werden. Mit der Trapezformel [für die Herleitung siehe AF09, S. 204-205] zur Berechnung der Fahrzeit ergibt sich für die Zeit, die das Shuttle benötigt, um von der aktuellen Position zum Puffer zu gelangen und dort die Lagereinheit aufzunehmen, folgender Zusammenhang:

$$t_{Buf} = \left(\frac{v_S}{a_S} + \frac{x_{akt}}{v_S} \right) + t_{P \rightarrow S} \quad (5.1)$$

Für die Zeit, bis die Einheit ihren Lagerort erreicht hat und abgeladen wurde, ergibt sich analog

$$t_{Lag} = \left(\frac{v_S}{a_S} + \frac{x_{Lag}}{v_S} \right) + t_{S \rightarrow Lag} \quad (5.2)$$

Für die gesamte Zeit des Prozesses gilt dann

$$t_{ein} = t_{Buf} + t_{Lag} \quad (5.3)$$

Zu beachten ist hier die Voraussetzung, dass der Puffer zum Zeitpunkt des Starts der Funktion bereits gefüllt ist.

Betrachtet man nun eine **Auslagerung**, so ergibt sich eine sehr ähnliche Abfolge – mit veränderten Übergabezeiten und einer unterschiedlichen Aufteilung des insgesamt zurückgelegten Weges

$$t_{\text{Lag}} = \left(\frac{v_S}{a_S} + \frac{|x_{\text{Lag}} - x_{\text{akt}}|}{v_S} \right) + t_{\text{Lag} \rightarrow S} \quad (5.4)$$

$$t_{\text{Buf}} = \left(\frac{v_S}{a_S} + \frac{x_{\text{Lag}}}{v_S} \right) + t_{S \rightarrow P} \quad (5.5)$$

$$t_{\text{aus}} = t_{\text{Lag}} + t_{\text{Buf}} \quad (5.6)$$

Je nach Modus des Aufrufes wird nun entsprechend die Einlagerungszeit oder die Auslagerungszeit errechnet und an die aufrufende Funktion übergeben.

5.2.2 [Calculate Lift Time]_f

Die Berechnung der Liftzeit basiert ebenfalls auf einer angepassten Form der Trapezformel für den Weg zwischen der aktuellen Ebene n_{akt} und der Zielebene n_{soll} . Da nur die Anzahl der Ebenen bekannt ist, entsteht der Weg erst durch Multiplikation mit dem Abstand der Ebenen. Die jeweilige Übergabezeit t_{transfer} ist abhängig von der aufrufenden Funktion; wird [Calculate Lift Time]_f z. B. aus [Calculate Input Cycle Time]_f aufgerufen, so entspricht t_{transfer} der Übergabezeit von der Übergabestation zum Lift $t_{I \rightarrow L}$.

$$t_L = \left(\frac{v_L}{a_L} + \frac{|n_{\text{soll}} - n_{\text{akt}}| \cdot h}{v_L} \right) + t_{\text{transfer}} \quad (5.7)$$

5.2.3 [Check Shuttle Status]_f

Diese Funktion dient zur Aktualisierung aller Hintergrundprozesse im Sinne der eventbasierten Simulation. Als Eingabeparameter dient die aktuelle Zeit, zu der die Überprüfung durchgeführt werden soll. Dazu ist es notwendig, für jeden möglichen Status des Shuttles eine eigene Entscheidungssequenz zu implementieren. Eine weitere Funktion, [Define next Step]_f (siehe S. 37), dient dazu, dem Shuttle einen neuen Auftrag zuzuweisen, sollte der Hintergrundprozess beendet sein. Die Abbildung 5.8 zeigt einen Überblick über den Ablauf und die Unterscheidungen dieser Funktion.

Im ersten Schritt werden die Daten von allen Shuttles, die [aktiv] sind, aus den zentral gespeicherten Werten für Shuttles und Puffer gesammelt. Der weitere Prozess läuft in Form einer Schleife ab, wobei die Anzahl der durchlaufenen Schleifen der Anzahl der Shuttles mit dem Status [aktiv] entspricht. Die Daten für das nächste Shuttle werden so lange geladen, bis alle Shuttles überprüft wurden.

Durch Überprüfung des Modus kann einer der folgenden Fälle eintreten: 1. [bewegen], 2. [ruhen], 3. [warten], 4. [einlagern] oder 5. [auslagern]. In der Folge sollen alle verschiedenen Möglichkeiten erläutert werden.

Betrachtet man Fall 1, so befindet sich das Shuttle auf dem Weg zum Puffer, um dort die nächste Einlagerung vorzunehmen. Ist der Einlagerungspuffer noch [leer], so geschieht

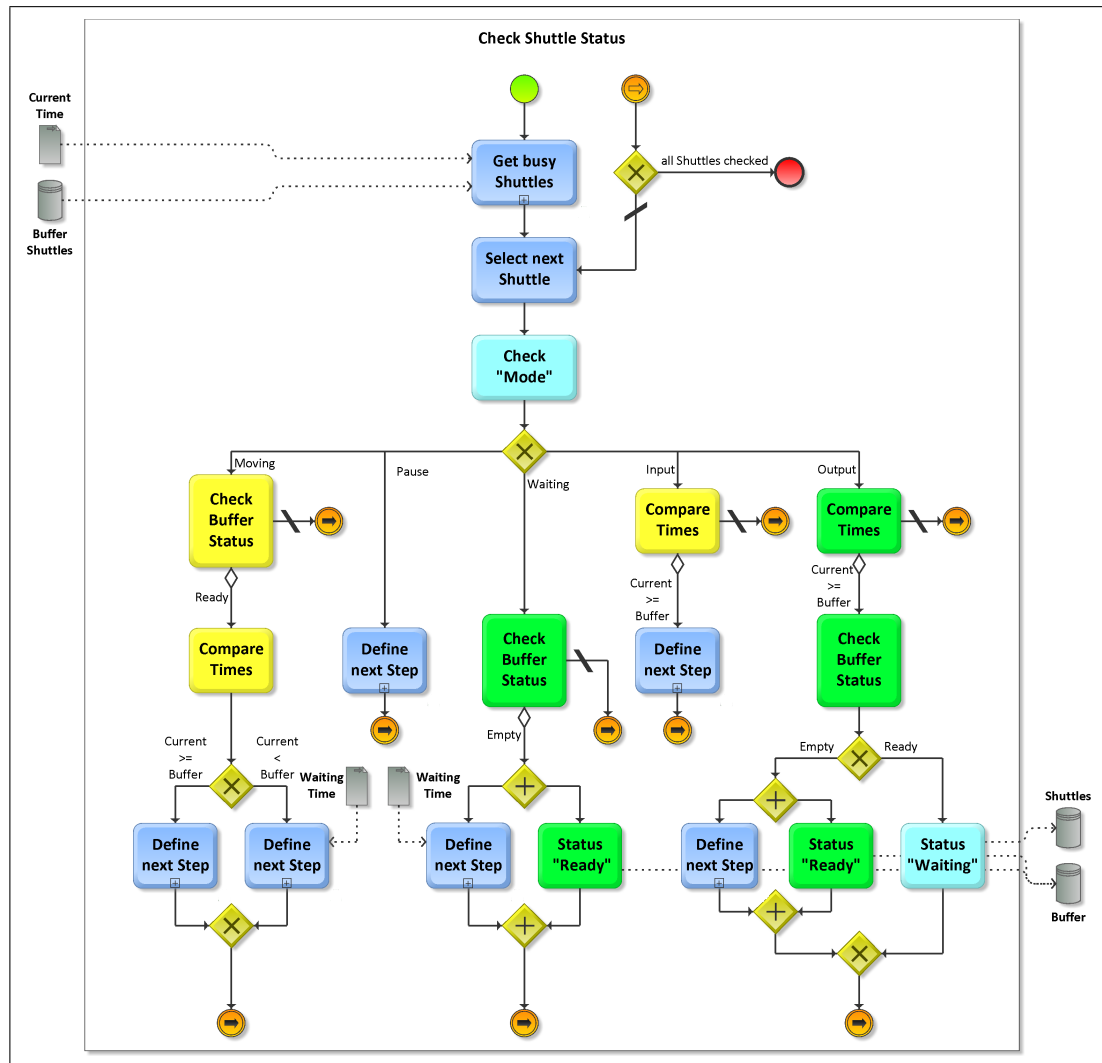


Abbildung 5.7: Funktionsweise: Shuttle Aktualisierung

zu diesem Zeitpunkt nichts und diese Schleife wird beendet. Ist der Puffer aber [bereit], so kann der Prozess weiterlaufen. Es wird nun mit Hilfe der Pufferzeit bestimmt, ob das Shuttle sich bereits an der Pufferposition befindet. Ist die aktuelle Zeit größer oder gleich der Zeit, die im Puffer gespeichert ist, ist das Shuttle bereits angekommen und $[Define\ next\ Step]_f$ wird ausgeführt, um dem Shuttle den nächsten Auftrag zu erteilen. Ist dies nicht der Fall, wird ebenfalls $[Define\ next\ Step]_f$ aufgerufen, diesmal aber mit dem optionalen Parameter Wartezeit, um die Endzeitpunkte des folgenden Vorganges richtig zu stellen. Als Wartezeit wird in diesem Fall die Differenz aus Puffer- und aktueller Zeit definiert. Damit ist die Aktualisierung dieses Shuttles abgeschlossen.

Der 2. Fall tritt dann ein, wenn zuvor ein Auslagerungsvorgang in einer Ebene ausgelöst wurde, dessen Shuttle [inaktiv] war. Durch den Aufruf von $[Define\ next\ Step]_f$ wird die Auslagerung vorgenommen und die aktuelle Schleife beendet.

Fall 3 tritt ein, wenn das Shuttle befüllt ist und an der Pufferposition wartet, da der Puffer

[*bereit*] und daher mit einer anderen Einheit belegt ist. Hat sich nach Überprüfung des Puffers an der Situation nichts geändert, verbleibt das Shuttle im Modus [*warten*] und das nächste Shuttle wird überprüft. Ist der Puffer aber inzwischen [*leer*], so kann die aktuelle Auslagerung beendet werden. Dadurch wird der Puffer wieder auf [*bereit*] gesetzt und [*Define next Step*]_f aufgerufen. Auch hier ist wieder die Übergabe einer Wartezeit notwendig. Diese entsteht durch die Entladung des Shuttles und entspricht damit der Übergabezeit $t_{S \rightarrow P}$. Dadurch wird der Endzeitpunkt des nächsten Vorganges um die angegebene Zeit korrigiert.

Befindet sich das Shuttle in einem Einlagerungsvorgang, so entspricht dies dem 4. Fall. Da die Endzeit des Vorganges im Einlagerungspuffer abgespeichert wurde, wird der aktuelle Zeitpunkt mit diesem verglichen. Ist die aktuelle Zeit kleiner als die Pufferzeit so ist der Vorgang noch nicht abgeschlossen und die Schleife wird beendet. Ist die aktuelle Zeit aber größer oder gleich der Pufferzeit, so wurde der Vorgang zwischen dem letzten Aufruf von [*Check Shuttle Status*]_f und dem aktuellen beendet. Danach wird [*Define next Step*]_f aufgerufen und diese Schleife beendet.

Der 5. und letzte Fall tritt ein, wenn das Shuttle eine Auslagerung durchführt. Auch hier wird zuerst die aktuelle Zeit mit der Pufferzeit verglichen, um zu überprüfen, ob der Vorgang inzwischen abgeschlossen wurde oder das Shuttle unverändert bleibt. Wenn die aktuelle Zeit größer oder gleich der Pufferzeit ist muss jedoch der Status des Puffers überprüft werden. Ist dieser nämlich bereits [*bereit*], so kann der Vorgang nicht abgeschlossen werden und das Shuttle wird in den Modus [*warten*] versetzt. Ist der Puffer aber [*leer*], so wird der Vorgang abgeschlossen. Dadurch wird der Puffer wieder [*bereit*] und [*Define next Step*]_f ausgeführt.

Dieser Vorgang wird solange wiederholt, bis alle Shuttles aktualisiert wurden. Die Abfolge der Ausführungen [*Check Shuttle Status*]_f wird durch [*Simulate single Cycle*]_f und deren Unterfunktion [*Calc real Time*]_f festgelegt. Dadurch ist die Berechnung der Zeiten im aktuellen Zyklus immer exakt.

5.2.4 [*Define next Step*]_f

Diese Funktion dient dazu, einem Shuttle, nachdem es einen Auftrag abgeschlossen hat, den nächsten Auftrag zuzuweisen, sofern es einen gibt. Sie erlaubt als optionalen Eingabeparameter die Übergabe einer Wartezeit. Außerdem kann sie direkt auf die zentral gespeicherten Entitäten Shuttles und Puffer zugreifen, um aktuelle Werte – bei Bedarf – abrufen bzw. abspeichern zu können. [*Define next Step*]_f wird ausschließlich von [*Check Shuttle Status*]_f aufgerufen. Die prinzipielle Abfolge der Funktionen ist in Abbildung 5.8 zu sehen. Wie in Abschnitt 4.6: Die Grenzen der Simulation (siehe S. 26) beschrieben, soll das Shuttle, sobald der Einlagerungspuffer gefüllt ist, eine Einlagerung durchführen. Ist er also [*bereit*], so wird ein Einlagerungsvorgang gestartet. Dazu wird [*Calculate Shuttle Time*]_f (vgl. Unterabschnitt 5.2.1) aufgerufen: mit den Koordinaten aus dem Koordinatenpuffer und der aktuellen Position des Shuttles. Die resultierende Einlagerungszeit wird dann als Pufferzeit gespeichert (sowohl die gesamte Zeit der Einlagerung als auch die Zeit, bis der Puffer geleert ist); außerdem wird das Shuttle auf [*einlagern*] gesetzt.

Ist der Einlagerungspuffer leer, wird als nächstes überprüft, ob noch eine Auslagerung

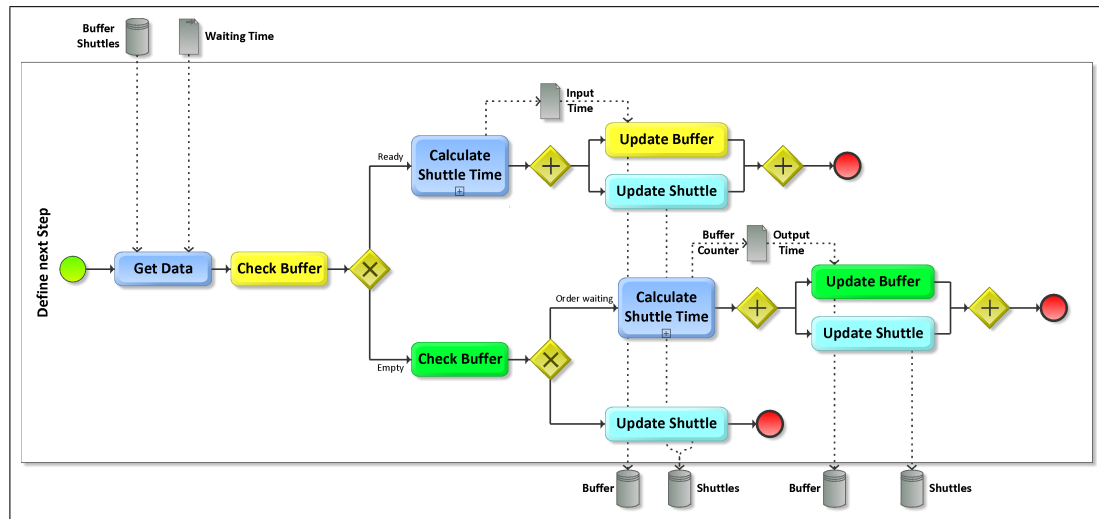


Abbildung 5.8: Funktionsweise: Nächster Auftrag

durchzuführen ist. Dazu wird die Anzahl der im Koordinatenpuffer gespeicherten Auslagerungskoodinaten mit dem Zähler verglichen. Ist die Anzahl größer als der Zähler, dann wird der nächste Auslagerungsvorgang aktiviert und der Zähler um eins erhöht. Die Koordinaten aus dem Puffer und die Position des Shuttles werden an $[Calculate\ Shuttle\ Time]_f$ (vgl. Unterabschnitt 5.2.1) übergeben; $[Calculate\ Shuttle\ Time]_f$ berechnet dann die Auslagerungszeit. Diese wird dann als Pufferzeit im Auslagerungspuffer abgespeichert und das Shuttle wird auf $[auslagern]$ gesetzt.

Ist der Koordinatenzähler gleich groß wie die Anzahl der Koordinaten im Puffer, so steht zur Zeit kein weiterer Auftrag an. Das Shuttle verbleibt an seiner aktuellen Position, der Status wird $[inaktiv]$ und der Modus $[ruhen]$.

Auch die Funktion $[Define\ next\ Step]_f$ arbeitet für Aufrufe des aktuellen Zyklus exakt. Da der Aufbau der Simulation möglichst modular und einfach aufgebaut sein soll (vgl. Abschnitt 4.2, S. 21), wird $[Define\ next\ Step]_f$ auch aufgerufen, wenn ein Hintergrundprozess beendet wird. Das Verhalten der Funktion basiert darauf, dass die neu gespeicherten Pufferzeiten aus der Addition der Resultate aus $[Calculate\ Shuttle\ Time]_f$ und der aktuellen Zeit entstehen. Dadurch kommt es aber bei Hintergrundprozessen zu einer gewissen Unschärfe, da für den Start des nächsten Auftrages nicht die Endzeit des letzten Auftrages, sondern die aktuelle Zeit verwendet wird, die schon weiter fortgeschritten ist. Da aber beim normalen Durchlauf der Simulation die Abstände der Aktualisierung relativ gering sind, wirkt sich der Fehler nicht signifikant auf das Gesamtsystem aus.

5.2.5 $[Calculate\ real\ Time]_f$

Mit Hilfe dieser Funktion wird für den Einlagerungs-, Zwischen- und Auslagerungszyklus die tatsächlich vergangene Zeit berechnet. Die Variable, die festlegt welcher Zyklus durchlaufen wird, wird als Modus bezeichnet. In Abschnitt 4.4 wurde bereits in Abbildung 4.2 (siehe S. 23) ein vereinfachtes Modell der verschiedenen Möglichkeiten vorgestellt.

In dieser Abbildung wird aber nicht darauf eingegangen, wie die Wartezeit tatsächlich zustande kommt. Abbildung 5.9 zeigt den gesamten Prozess, der notwendig ist, um die real vergangene Zeit zu erfassen. Als Eingabeparameter dienen die aktuellen Koordinaten des Liftes und des Shuttles in der betreffenden Ebene sowie die aktuellen Übergabezeiten und die aktuelle Zeit t_a . Außerdem werden die globalen Entitäten Shuttles und Puffer angesprochen.

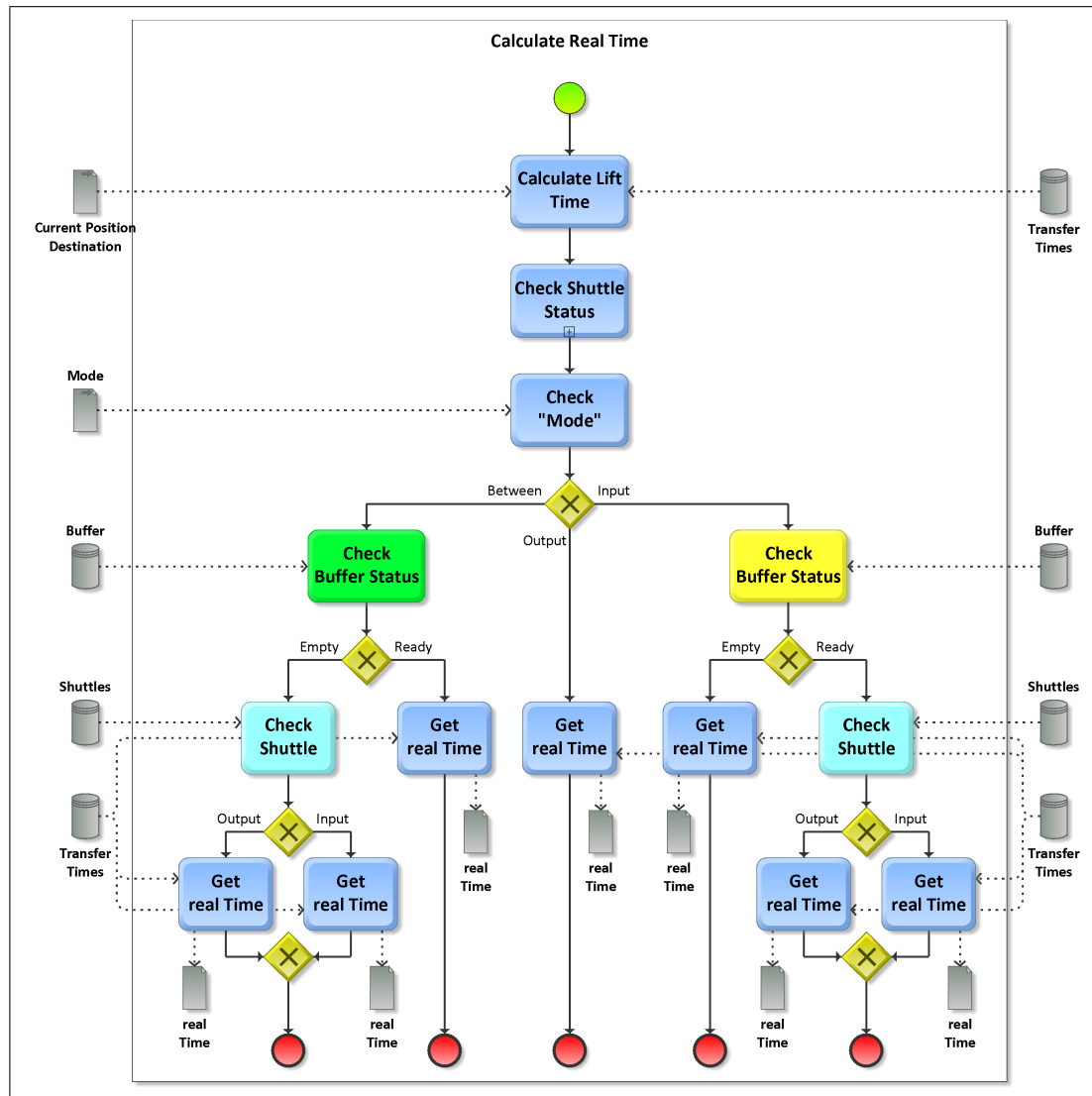


Abbildung 5.9: Funktionsweise: Zeitbestimmung

Nach der Ausführung von $[Calculate Lift Time]_f$ (vgl. Unterabschnitt 5.2.2) wird die Hilfszeit t_h als

$$t_h = t_L + t_a \quad (5.8)$$

definiert. Danach wird $[Check Shuttle Status]_f$ mit dem Argument t_h ausgeführt und damit die Situation bei der Ankunft des Liftes in der Zielebene dargestellt. Nun gibt es – abhängig vom Modus – folgende Möglichkeiten

1. die Einlagerung
2. den Zwischenzyklus
3. die Auslagerung

Betrachten wir nun Möglichkeit 1: diese tritt ein, wenn $[Calc\ real\ Time]_f$ im Einlagerungszyklus aufgerufen wurde. Als erstes wird der Einlagerungspuffer überprüft, ist dieser nämlich $[leer]$, kann der Prozess ohne Unterbrechung weitergeführt werden. Da die reale Zeit t_{real} als Differenz definiert ist, ergibt sie sich zu

$$\begin{aligned} t_{real} &= t_h + t_{L \rightarrow P} - t_a \\ &= t_L + t_{L \rightarrow P} \end{aligned} \quad (5.9)$$

und wird an die aufrufende Funktion übergeben. Ist der Puffer aber $[bereit]$, hängt das weitere Vorgehen vom Status des Shuttles ab. Dieser kann entweder a) $[auslagern]$ oder b) $[einlagern]$ sein.

Ergibt die Überprüfung a), wird der Zeitpunkt (t_{aus}), zu dem der Auslagerungsvorgang beendet sein wird, aus dem Puffer gelesen und $[Check\ Shuttle\ Status]_f$ mit dem Argument t_{aus} ausgeführt; dadurch wird das System am Ende des Vorganges dargestellt. Da sowohl die Übergabezeit, die zur Leerung des Puffers als auch diejenige, die zur Entladung des Liftes benötigt wird, nach außen wirksam ist, ergibt sich die reale Zeit nun zu

$$t_{real} = t_{aus} + t_{P \rightarrow S} + t_{L \rightarrow P} - t_a \quad (5.10)$$

Ergibt die Überprüfung aber b), wird der Zeitpunkt (t_{ein}), zu dem der Einlagerungsvorgang beendet sein wird, aus dem Puffer gelesen und $[Check\ Shuttle\ Status]_f$ mit dem Argument t_{ein} ausgeführt, um das System am Ende des Vorganges darzustellen. Danach wird – dank der automatischen Ausführung von $[Define\ next\ Step]_f$ – der nächste Einlagerungsvorgang gestartet und die neue Einlagerungszeit sowie der Zeitpunkt (t_{Puf}), zu dem der Puffer geleert sein wird, im Puffer gespeichert. Der Zeitpunkt, zu dem die aktuelle Einlagerung durchgeführt werden kann, entspricht also dem Wert von t_{Puf} und die reale Zeit ergibt sich zu

$$t_{real} = t_{Puf} + t_{L \rightarrow P} - t_a \quad (5.11)$$

und wird als Ergebnis übergeben.

Als nächstes soll Möglichkeit 2 betrachtet werden: diese tritt ein, wenn die Funktion im Zwischenzyklus aufgerufen wurde. Dieses Mal wird der Auslagerungspuffer überprüft; ist dieser $[bereit]$, kommt es zu keiner Wartezeit und die reale Zeit kann, unter Berücksichtigung der Übergabezeit vom Puffer zum Lift, als

$$\begin{aligned} t_{real} &= t_h + t_{P \rightarrow L} - t_a \\ &= t_L + t_{P \rightarrow L} \end{aligned} \quad (5.12)$$

berechnet werden. Ist der Puffer aber $[leer]$, so muss der Modus des Shuttles überprüft werden, um das weitere Vorgehen zu definieren. Dieser kann nun wieder entweder a) $[auslagern]$ oder b) $[einlagern]$ sein.

Tritt Fall a) ein, so führt das wieder zum Auslesen der Pufferzeit t_{aus} , mit der $[Check\ Shuttle$

$Status_f$ aufgerufen wird. Danach ist der Auslagerungspuffer $[bereit]$ und der Prozess kann fortgesetzt werden. Die reale Zeit ergibt sich mit der Übergabezeit damit zu

$$t_{\text{real}} = t_{\text{aus}} + t_{P \rightarrow L} - t_a \tag{5.13}$$

und wird wieder an die aufrufende Funktion übergeben.

Tritt aber der Fall b) ein, muss die Einlagerungszeit t_{ein} aus dem Puffer gelesen und damit $[Check Shuttle Status]_f$ aufgerufen werden. Falls das Shuttle, was durch das Priorisieren der Einlagerung passieren könnte, noch immer den Modus $[einlagern]$ hat, muss dieser Vorgang ein weiteres Mal wiederholt werden. Ist der Modus des Shuttles $[auslagern]$, so wird die aktuelle Pufferzeit t_{aus} (man beachte, dass diese Zeit die vorher ausgelesenen Einlagerungszeiten enthält) ausgelesen und die reale Zeit wird ebenfalls zu

$$t_{\text{real}} = t_{\text{aus}} + t_{P \rightarrow L} - t_a \tag{5.14}$$

Diese Zeit wird dann als Ergebnis übergeben.

Die 3. und letzte Möglichkeit tritt ein, wenn die Funktion im Auslagerungszyklus aufgerufen wurde. Dies ist auch die einfachste Variante da, wie schon ausführlich ausgeführt, keine Wartezeiten auftreten können. Damit ergibt sich die reale Zeit, unter Berücksichtigung der Übergabezeit vom Lift zur Übergabestation, zu

$$\begin{aligned} t_{\text{real}} &= t_h + t_{L \rightarrow O} - t_a \\ &= t_L + t_{L \rightarrow O} \end{aligned} \tag{5.15}$$

5.3 Exemplarische Berechnung der Doppelspielzeit

In diesem Kapitel sollen durch ein Zahlenbeispiel die in den Kapiteln 4 und 5 beschriebenen Abläufe und Funktionen anschaulich gemacht werden. Es wird sowohl eine Berechnung zu Beginn der Simulation als auch eine im laufenden Betrieb dokumentiert. Dazu werden die Parameter, wie in Tabelle 5.8 dargestellt, festgelegt.

Geometrie			System				Simulation		
$l =$	20 m	$v_L =$	$5 \frac{\text{m}}{\text{s}}$	$t_{I \rightarrow L} =$	2 s	$t_{L \rightarrow O} =$	2 s	$n_{\text{Sim}} =$	100
$h =$	0,3 m	$v_S =$	$2 \frac{\text{m}}{\text{s}}$	$t_{L \rightarrow P} =$	2 s	$t_{P \rightarrow L} =$	2 s	$n_{\text{aS}} =$	4
$w =$	0,5 m	$a_L =$	$7 \frac{\text{m}}{\text{s}^2}$	$t_{P \rightarrow S} =$	6 s	$t_{S \rightarrow P} =$	8 s	$x_{\text{in}} =$	10 m
$n_S =$	10	$a_S =$	$2 \frac{\text{m}}{\text{s}^2}$	$t_{S \rightarrow \text{Lag}} =$	8 s	$t_{\text{Lag} \rightarrow S} =$	6 s		
$n_{\text{ÜS}} =$	0								

Tabelle 5.8: Werte der Parameter

Der Ablauf der Simulation gestaltet sich nun analog zu dem in Abbildung 5.1 gezeigten Prozess.

5.3.1 Start der Simulation

Nach dem Setzen der Parameter wird zunächst $[Create\ Environment]_f$ durchgeführt; diese Funktion erzeugt nun einen Koordinatenvektor, der $100(n_{Sim})$ Zeilen hat. Hier ein Ausschnitt, der die relevanten Werte für die ersten Berechnungen zeigt:

$$[n_{ein} \quad x_{ein}] = \begin{bmatrix} 7 & 10,61 \\ 4 & 1,48 \\ \vdots & \vdots \end{bmatrix} \quad [n_{aus} \quad x_{aus}] = \begin{bmatrix} 5 & 17,70 \\ 9 & 2,36 \\ 6 & 12,80 \\ 6 & 8,21 \\ 8 & 2,40 \\ \vdots & \vdots \end{bmatrix}$$

Danach werden die ersten 4 (n_{as}) Shuttles auf die Auslagerung, das Shuttle in Ebene 7 auf die Einlagerung vorbereitet. Vom Ausgangszustand:

[10 [inaktiv] [ruhen]]

werden die auslagernden Shuttles auf

[0 [aktiv] [auslagern]]

und das einlagernde auf

[0 [aktiv] [bewegen]]

aktualisiert. Einerseits werden die Auslagerungszeiten in den entsprechenden Ebenen mit Hilfe der Gleichungen 5.4 - 5.6 berechnet und im Auslagerungspuffer gespeichert, andererseits wird die Zeit des [bewegen] Shuttles mit Gleichung 5.1 berechnet, wobei die Übergabezeit nicht mit einfließt, und im Einlagerungspuffer gespeichert:

$$[[Status] \quad t_{aus}] = \begin{bmatrix} \vdots & \vdots \\ [leer] & 28,70 \\ [leer] & 23,80 \\ \vdots & \vdots \\ [leer] & 21,00 \\ \vdots & \vdots \end{bmatrix} \quad [[Status] \quad (t_{ein}/t_{Puf})] = \begin{bmatrix} \vdots & \vdots \\ [leer] & (0/6,00) \\ \vdots & \vdots \end{bmatrix}$$

Zu beachten ist, dass in den Zyklen 3 und 4 eine Auslagerung in Ebene 6 stattfinden soll. Die Pufferzeit wird nur für den ersten Auftrag errechnet, der zweite Auftrag wird im Koordinatenpuffer gespeichert und erst aktiviert, nachdem der erste beendet wurde. Nun ist die Ausgangssituation für die Simulation geschaffen und der eigentliche Simulationzyklus beginnt.

5.3.2 Berechnung des ersten Zyklus

In diesem Beispiel soll die erste Zykluszeit detailliert und Schritt für Schritt berechnet werden. Diese ist nicht repräsentativ für die mittlere Spielzeit, da mit hoher Wahrscheinlichkeit eine Wartezeit im Zwischenzyklus auftritt.

Zu Beginn des Zyklus ist

$$t_a = 0$$

Die Koordinaten der ersten Einlagerung sind

$$[7 \quad 10,61]$$

und die der nächsten Einlagerung

$$[4 \quad 1,48]$$

Nun wird $[Check\ Shuttle\ Status]_f(0)$ ausgeführt. Dieser Schritt führt zu keiner Änderung im System, da noch keine Zeit vergangen ist. Als nächstes wird die nächste Einlagerung initialisiert. Da das Shuttle in Ebene 4 [*inaktiv*] ist, ändern sich Status und Modus des Shuttles sowie der Einlagerungspuffer analog wie oben beschrieben. Auch $[Check\ Input\ Status]_f$ ist erfolgreich – es kann natürlich keine Deadlock-Situation herrschen – und damit startet die Berechnung der realen Zeit mit $[Calculate\ real\ Time]_f$. Die Eingabeparameter sind

$$t_a = 0; n_{ist} = 0; n_{soll} = 7$$

Dadurch ergibt sich für die Liftzeit nach Gleichung 5.7 (die Übergabezeit $t_{I \rightarrow L}$ wird berücksichtigt)

$$t_L = 3,13$$

was in diesem Fall auch t_h entspricht. Da eine Einlagerung vorliegt und der Puffer in der Ebene 7 [*leer*] ist, errechnet sich die reale Zeit nach Gleichung 5.11 zu

$$t_{real} = 5,13$$

Nun wird das System aktualisiert und der Puffer in Ebene 7 wird dadurch [*bereit*]. Die aktuelle Zeit und die Position des Liftes werden zu

$$t_a = 5,13; n_{ist} = 7$$

Damit ist der Einlagerungszyklus beendet und der Zwischenzyklus beginnt. Die aktuellen Koordinaten der Auslagerung entsprechen

$$[5 \quad 17,70]$$

Durch Ausführen von $[Check\ Shuttle\ Status]_f(5,13)$ wird erkannt, dass der Puffer in Ebene 7 [*bereit*] ist und das Shuttle sich im Modus [*bewegen*] befindet. Dies führt dazu, dass die

Einlagerungszeiten, nach Gleichung 5.1 und Gleichung 5.2, inklusive Wartezeit berechnet werden und führt dazu, dass sich das Shuttle und der Einlagerungspuffer folgendermaßen ändern:

$$\begin{bmatrix} \vdots & \vdots \\ 10,61 & [\text{aktiv}][\text{einlagern}] \\ \vdots & \vdots \end{bmatrix} \quad \begin{bmatrix} \vdots & \vdots \\ [\text{leer}] & (26,31/12,00) \\ \vdots & \vdots \end{bmatrix}$$

Da die Aktivierung der nächsten Einlagerung bereits erfolgt und das Shuttle in Ebene 4 nun *[aktiv]* ist, wird dieser Schritt übersprungen und direkt zur Berechnung der realen Zeit übergegangen. Die Eingabeparameter sind nun wie folgt

$$t_a = 5,13; n_{ist} = 7; n_{soll} = 5$$

Die Liftzeit, errechnet nach Gleichung 5.7, ergibt sich zu

$$t_L = 0,83$$

und die Hilfszeit nach Gleichung 5.8 beträgt

$$t_h = 5,97$$

Wird nun $[\text{Check Shuttle Status}]_f(5,97)$ ausgeführt, so wird festgestellt, dass der Auslagerungspuffer in Ebene 5 noch *[leer]* ist. Da sich das Shuttle im Modes *[auslagern]* befindet, entspricht die Auslagerungszeit der Pufferzeit und die reale Zeit berechnet sich nach Gleichung 5.13 zu

$$t_{\text{real}} = 25,57$$

Da während dieses Vorganges nochmals $[\text{Check Shuttle Status}]_f(28,70)$ ausgeführt wurde, wurden auch andere Hintergrundprozesse abgeschlossen; die aktuelle Situation der Shuttles und des Auslagerungspuffers ergibt sich zu

$$\begin{bmatrix} \vdots & \vdots \\ 0 & [\text{inaktiv}][\text{ruhen}] \\ 0 & [\text{aktiv}][\text{auslagern}] \\ \vdots & \vdots \\ 0 & [\text{inaktiv}][\text{ruhen}] \\ \vdots & \vdots \end{bmatrix} \quad \begin{bmatrix} \vdots & \vdots \\ [\text{bereit}] & 28,70 \\ [\text{bereit}] & 52,91 \\ \vdots & \vdots \\ [\text{bereit}] & 21,00 \\ \vdots & \vdots \end{bmatrix}$$

Die Auslagervorgänge in den Ebenen 5 und 9 wurden also in der Zwischenzeit beendet, in Ebene 6 wurde der nächste Auftrag gefunden und aktiviert. Die Koordinaten für den Auftrag werden aus dem Koordinatenpuffer ausgelesen. Berechnet man mit den Gleichungen 5.4 bis 5.6 die Shuttlezeit, so ergibt sich

$$t_S = 24,21$$

Man würde nun erwarten, dass die neue Auslagerungszeit die Summe aus t_5 und der alten Pufferzeit 21,00 also

$$t_{aus} = 45,21$$

beträgt. Tatsächlich tritt aber das am Ende von Unterabschnitt 5.2.4 geschilderte Problem auf. Da $[Define\ next\ Step]_f$ erst bei $t = 28,70$ ausgeführt wird, wird diese Zeit als Startzeit für den Prozess herangezogen und die Shuttlezeit ergibt sich zu

$$t_{aus} = 52,91$$

Danach wird eine weitere Aktualisierung des Systems durchgeführt, der Puffer in Ebene 5 wird $[leer]$ und die aktuelle Zeit sowie die Position des Liftes werden zu

$$t_a = 30,70; n_{ist} = 5$$

Der Auslagerungszyklus beginnt nun wieder mit der, schon erfolgten, Aktivierung der nächsten Einlagerung. Danach wird das nächste „aktive Shuttle“ vorbereitet, die Koordinaten hierfür werden der fünften Ebene der Auslagerungskordinaten entnommen und sind damit

$$[8\ 2,40]$$

Da das Shuttle in Ebene 8 $[inaktiv]$ ist, wird es auf $[aktiv]$ gesetzt und der Koordinatenpuffer gefüllt. Nachdem $[Check\ Shuttle\ Status]_f(30,70)$ ausgeführt wurde, wird die korrekte Auslagerungszeit berechnet und das Shuttle sowie der Puffer werden angepasst.

$$\begin{bmatrix} \vdots & \vdots \\ 0 & [aktiv][auslagern] \\ \vdots & \vdots \end{bmatrix} \quad \begin{bmatrix} \vdots & \vdots \\ [leer] & 51,70 \\ \vdots & \vdots \end{bmatrix}$$

Die Eingabeparameter für $[Calculate\ real\ Time]_f$ sind nun wie folgt

$$t_a = 30,70; n_{ist} = 5; n_{soll} = 0$$

Da sich das System im Auslagerungszyklus befindet, lässt sich nach der Berechnung der Liftzeit die reale Zeit direkt mit Gleichung 5.15 berechnen

$$t_{real} = 3,01$$

Nun werden abschließend noch die aktuelle Zeit und die Liftposition aktualisiert

$$t_a = 33,71; n_{ist} = 0$$

und die Zykluszeiten einzeln

$$[t_{Einlagerung}\ t_{Zwischen}\ t_{Auslagerung}] = [5,13\ 25,57\ 3,01]$$

beziehungsweise als Summe

$$t_{\text{Zyklus}} = 33,71$$

ausgegeben. Dieser Wert entspricht natürlich der aktuellen Systemzeit. Abschließend soll noch ein Abbild des aktuellen Zustandes des Systems, aufgeteilt in Shuttles, Auslagerungs- und Einlagerungspuffer, präsentiert werden. Dieses umfasst die Ebenen 4 bis 9, um aufzuzeigen, was sich während eines Zyklus verändert hat:

⋮		⋮		⋮		⋮	
0	[aktiv][bewegen]		[leer]	0		[leer]	(0/6)
0	[inaktiv][ruhen]		[leer]	28,70		[leer]	(0/0)
0	[aktiv][auslagern]		[bereit]	52,91		[leer]	(0/0)
10,61	[aktiv][auslagern]		[leer]	0		[leer]	(26,31/12)
0	[aktiv][auslagern]		[leer]	51,70		[leer]	(0/0)
0	[inaktiv][ruhen]		[bereit]	21		[leer]	(0/0)
	⋮		⋮		⋮		⋮

5.3.3 Berechnung zur Laufzeit

Im zweiten Beispiel nun die Situation in Zyklus 50 dargestellt. Die Änderungen in den Hintergrundprozessen werden nicht mehr explizit erläutert, sondern sollen an Hand des Simulationsabbildes am Ende des Zyklus erklärt werden. Die Parameter sind noch dieselben wie in Tabelle 5.8 definiert.

Die Werte für die aktuelle Zeit und die Liftposition sind

$$t_a = 690,25; n_{ist} = 0$$

Nach der Aktivierung der nächsten Einlagerung sieht die jeweilige Situation der Shuttles, sowie des Auslagerungs- und Einlagerungspuffers wie folgt aus:

6,33	[aktiv][einlagern]		[leer]	521,95		[leer]	(702,49/690,32)
12,61	[aktiv][einlagern]		[leer]	198,04		[leer]	(708,02/692,71)
0	[inaktiv][ruhen]		[leer]	649,48		[leer]	(62,88/47,89)
0	[aktiv][auslagern]		[leer]	695,84		[leer]	(437,61/426,22)
0	[aktiv][bewegen]		[leer]	636,45		[leer]	(679,55/597,90)
0	[inaktiv][ruhen]		[bereit]	674,18		[leer]	(464,97/449,80)
0	[aktiv][bewegen]		[leer]	581,63		[leer]	(691,79/679,70)
0	[aktiv][auslagern]		[leer]	710,80		[leer]	(516,41/503,40)
0	[inaktiv][ruhen]		[leer]	641,95		[leer]	(620,01/605,55)
0	[inaktiv][ruhen]		[leer]	549,66		[leer]	(523,77/514,62)

Die aktuellen Koordinaten aus dem Koordinatenvektor betragen:

$$[n_{ein} \quad x_{ein}] = \begin{bmatrix} \vdots & \vdots \\ 5 & 2,24 \\ 7 & 0,28 \\ \vdots & \vdots \end{bmatrix} \quad [n_{aus} \quad x_{aus}] = \begin{bmatrix} \vdots & \vdots \\ 6 & 3,32 \\ \vdots & \vdots \\ 1 & 17,42 \\ \vdots & \vdots \end{bmatrix}$$

Beim Durchlaufen des Einlagerungszyklus bewegt sich der Lift von der Ebene 0 zur Ebene 5; da der Einlagerungspuffer [*leer*] ist, berechnen sich die reale Zeit mit Gleichung 5.9 zu

$$t_{\text{real}} = 5,01$$

und die aktuelle Zeit zu

$$t_a = 695,67$$

Danach wird der Zwischenzyklus ausgeführt: Der Lift bewegt sich von Ebene 5 in die Ebene 6 und kann dort, da der Puffer [*bereit*] ist, ohne Unterbrechung die Lagereinheit aufnehmen. Damit ergibt sich die reale Zeit, nach Gleichung 5.12, zu

$$t_{\text{real}} = 2,77$$

und die aktuelle Zeit zu

$$t_a = 698,44$$

Im Schlusszyklus wird nun nur noch die Zeit berechnet, bis die Einheit das System verlassen hat. Mit Gleichung 5.15 ergibt sich die reale Zeit zu

$$t_{\text{real}} = 3,07$$

und die aktuelle Zeit zu

$$t_a = 701,52$$

Damit ergeben sich die Zykluszeiten zu

$$[t_{\text{Einlagerung}} \quad t_{\text{Zwischen}} \quad t_{\text{Auslagerung}}] = [3,01 \quad 2,77 \quad 3,07]$$

beziehungsweise als Summe

$$t_{\text{Zyklus}} = 10,86$$

Diese Zeit eignet sich wesentlich besser, um den ganzen Prozess zu charakterisieren, da keine Wartezeit auftritt.

Abschließend wird wieder die Situation der Shuttles, des Auslagerungs- und Einlagerungspuffers betrachtet:

6,33	[aktiv][einlagern]	[leer]	521,95	[leer]	(702,49/690,32)
12,61	[aktiv][einlagern]	[leer]	198,04	[leer]	(708,02/692,71)
0	[inaktiv][ruhen]	[leer]	649,48	[leer]	(62,88/47,89)
0	[inaktiv][ruhen]	[bereit]	695,84	[leer]	(437,61/426,22)
2,24	[aktiv][einlagern]	[leer]	636,45	[leer]	(711,79/701,67)
0	[inaktiv][ruhen]	[leer]	674,18	[leer]	(464,97/449,80)
0	[aktiv][bewegen]	[leer]	581,63	[leer]	(691,79/679,70)
0	[aktiv][auslagern]	[leer]	710,80	[leer]	(516,41/503,40)
0	[inaktiv][ruhen]	[leer]	641,95	[leer]	(620,01/605,55)
0	[inaktiv][ruhen]	[leer]	549,66	[leer]	(523,77/514,62)

In Ebene 4 wurde der Auslagerungsprozess beendet, in Ebene 5 wurde eine Einlagerung gestartet und der Puffer in Ebene 6 wurde geleert. Das aktivieren des nächsten „aktiven Shuttles“ blieb im Moment ohne direkte Folgen, da in Ebene 1 noch eine Einlagerung durchgeführt wird.

5.4 Funktionen der Benutzeroberfläche

Die Benutzeroberfläche soll die Bedienung der Simulation erleichtern, sodass keine aufwändige Codeeingabe notwendig ist. Sie dient dazu, die Eingabeparameter aufzunehmen und die Ergebnisse der Simulation visuell aufzubereiten. Außerdem ermöglicht sie es, mehrere Simulationen hintereinander mit verschiedenen Parametern automatisiert zu durchlaufen.

5.4.1 Die Simulation

Diese Einstellung ermöglicht es, eine einzelne Simulation mit definierten Parametern zu starten, die in der Eingabemaske definiert werden. Dadurch wird die Funktion $[Simulation]_f$ aufgerufen. Außerdem wurde ein grafisches Feedback implementiert, das den Fortschritt der aktuellen Simulation anzeigt. Des Weiteren gibt es die Möglichkeit, die Ergebnisse der aktuellen Simulation einem schon vorhandenen Datensatz hinzuzufügen.

Nach dem Abschluss der Simulation gibt es die Option, die Ergebnisse als .mat Datei abzuspeichern oder sie in den Workspace von MATLAB zu exportieren, um dort mit den Werkzeugen des Softwarepakets weitere Untersuchungen durchzuführen. Die Funktion $[Create Data Matrix2]_f$ dient zur Umwandlung der – als Structure gespeicherten – Ergebnisse in eine Matrix, um die Grundlage für einen manuellen Plot zu schaffen. Wird keine Option gewählt, werden die Resultate nur lokal gespeichert und können dadurch grafisch dargestellt werden.

5.4.2 Die Variation

Um eine große Menge an verschiedenen Simulationsergebnissen zu produzieren, ist die Simulation nicht geeignet, da jedes Mal die neuen Parameter manuell eingegeben werden müssten. Deshalb wurde die Funktion $[Variation]_f$ implementiert, die automatisiert eine vorgegebene Reihe an verschiedenen Setups hintereinander berechnet und die Ergebnisse übersichtlich zusammenfasst. Die Parameter, die variiert werden können, sind:

- Länge
- Anzahl der Shuttles
- Anzahl der „aktiven Shuttles“
- Position der Übergabestelle
- Erzeugung der Koordinaten

Die ersten vier Optionen bedürfen keiner weiteren Erklärung. Die Erzeugung der Koordinaten bezieht sich auf die Funktion $[Create\ Environment]_f$ (vgl. Unterabschnitt 5.1.2, S. 29) und ermöglicht es, die Koordinaten in einer solchen Weise zu modifizieren, dass dieselbe Ebene nicht zwei Mal in direkter Folge für einen Auftrag verwendet wird. Dies dient einerseits einer Verbesserung der Performance und andererseits dem qualitativen Vergleich mit der Simulation der Koordinaten ohne vorherige Modifikation. Die Abbildung 5.10 zeigt die Hierarchie, mit der die Parameter variiert werden können.

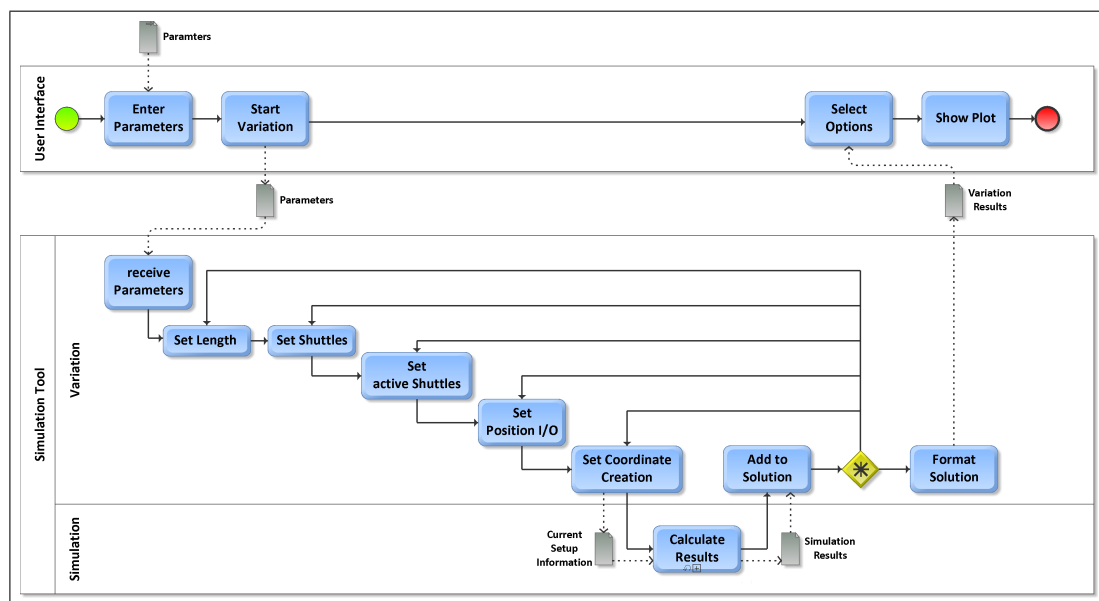


Abbildung 5.10: Funktionsabfolge der Variation

Der obere Pool zeigt, wie schon bei der Simulation, die Benutzeroberfläche, in die die Parameter eingegeben werden. Alternativ zu den geometrischen Parametern sowie den System- und Simulationsparametern können die oben genannten Variationsparameter spezifiziert werden. Diese legen dann Wertebereiche mit der Schrittweite sowie unterer und oberer Grenze fest.

Die Funktionsweise von $[Variation]_f$ lässt sich als verschachtelte Schleife beschreiben. Jeder

Schleifendurchgang erzeugt eine eigene Simulation, die mit den aktuellen Parametern $[Simulation]_f$ aufgerufen wird. Die Ergebnisse werden jeweils in eine gesamte Lösung zusammengefasst und übersichtlich strukturiert. Ein Setup wird immer von innen heraus variiert; der Vorgang soll beispielsweise anhand einer Variation der Länge (20 m und 40 m) und der Shuttles (10, 20 und 30 Shuttles) erläutert werden. Die Schleife führt zu folgender Simulationsabfolge:

- Länge 20 m
 - 10 Shuttles
 - 20 Shuttles
 - 30 Shuttles
- Länge 40 m
 - 10 Shuttles
 - 20 Shuttles
 - 30 Shuttles

Am Ende werden sechs verschiedene Setups simuliert. Es kann also mit wenigen Eingaben eine sehr hohe Anzahl an Ergebnissen für unterschiedliche Konfigurationen generiert werden.

Abbildung 5.11 zeigt nun die tatsächliche Benutzeroberfläche, die der Erstellung von Simulationen und Variationen dient; die Auswahl lässt sich im Feld *Options* festlegen. Danach werden die Parameter in die vordefinierten Felder eingegeben und mit dem Druck auf *Set Parameters* bestätigt; dann wird, je nach vorheriger Auswahl, der entsprechende Button freigegeben. Zusätzlich lässt sich noch die Box *Load existing data* aktivieren, um die aktuellen Ergebnisse einem zuvor erzeugten Datensatz hinzuzufügen. Die grafische Darstellung kann über die Auswahl der gewünschten Option in der Gruppe *Plot Options* und Bestätigung mit dem Button *Plot* erreicht werden.

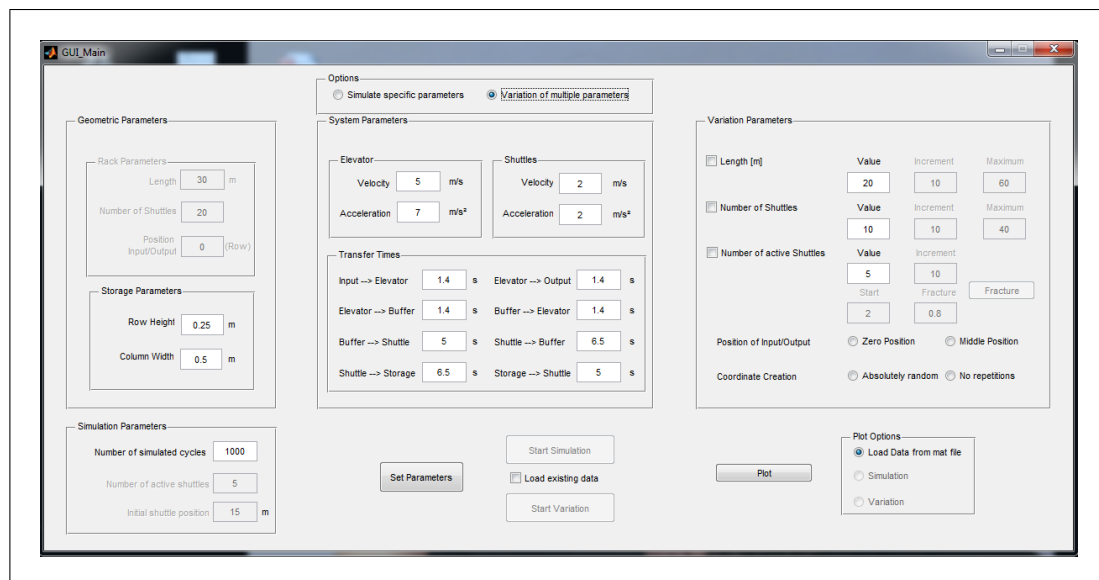


Abbildung 5.11: Die Benutzeroberfläche GUI Main

Der Aufruf dieses Fensters erfolgt, nachdem die Simulation im Working Directory von MATLAB geladen wurde, mit der Eingabe des Befehls GUI_Main im Command Window.

5.4.3 Die grafische Ausgabe

Ein Plot kann direkt aus der Benutzeroberfläche gestartet werden; als Datengrundlage dafür dient entweder eine zuvor ausgeführte einzelne Simulation oder eine Reihe an Simulationen, die mit Hilfe der zuvor beschriebenen Variation durchgeführt wurden. Alternativ dazu können Daten auch in Form einer .mat Datei geladen werden. Die Verarbeitung von Variationen mit einer großen Anzahl an Datensätzen kann einige Minuten in Anspruch nehmen.

Für die Darstellung der Ergebnisse gibt es folgende Optionen:

- detaillierte Darstellung der Zeiten unterteilt in
 - Einlagerungs-, Zwischen- und Auslagerungszeit
 - optimale und reale Zeit
 - normalisierte Werte
- Variation
 - der Länge
 - der Shuttles
 - der „aktiven Shuttles“
- allgemeine Optionen
 - Multiplot
 - individuelle Benennung
 - Export
- dreidimensionale Grafiken
 - Doppelspielzeiten
 - Umschlagleistungen

Durch die Auswahl der ersten Option kann die Zusammensetzung der Zeiten für jeden einzelnen Zyklus einer bestimmten Simulation analysiert werden. Diese kann nun entweder geteilt in die einzelnen Zyklusabschnitte, oder in Form der realen und der optimalen Zeiten dargestellt werden.

Die zweite Option ermöglicht es, verschiedene Bereiche von Werten zu vergleichen. Dabei werden zwei Parameter konstant gehalten und der dritte, zwischen den eingegebenen Grenzen, variiert. Dabei werden nicht die einzelnen Zyklen, sondern die mittleren Doppelspielzeiten bzw. die daraus resultierenden Umschlagleistungen aufgetragen.

Durch Aktivieren der Option Multiplot, wird die nächste Grafik der aktuellen hinzugefügt, zusätzlich lässt sich die Benennung der einzelnen Kurven individualisieren. Jeder Plot kann außerdem exportiert werden, um ihn in einem gewählten Format abzuspeichern; gleichermaßen kann die Grafik zuvor mit dem Ploteditor von MATLAB bearbeitet werden. Außerdem ist es möglich, sich eine Vielzahl an Daten als übersichtliche 3D-Grafik anzeigen zu lassen. Dadurch können große Datenmengen verglichen werden. Die möglichen Darstellungsarten umfassen sowohl die Anzeige der Werte – abhängig von den Parametern

Länge und Anzahl der Shuttles – als auch die Darstellung in Abhängigkeit der Anzahl der Lagerplätze.

Abbildung 5.12 zeigt die Darstellung der Benutzeroberfläche, in der die zuvor beschriebenen Optionen ausgewählt werden können. Mit dem Drücken des Buttons *Create Plot* wird ein neuer Plot in der Oberfläche generiert. Der Button *Edit/Save Plot* dient zum Aufruf der Grafik in einem separaten Fenster, dort kann die Grafik mit den MATLAB eigenen Tools weiter bearbeitet und personalisiert werden.

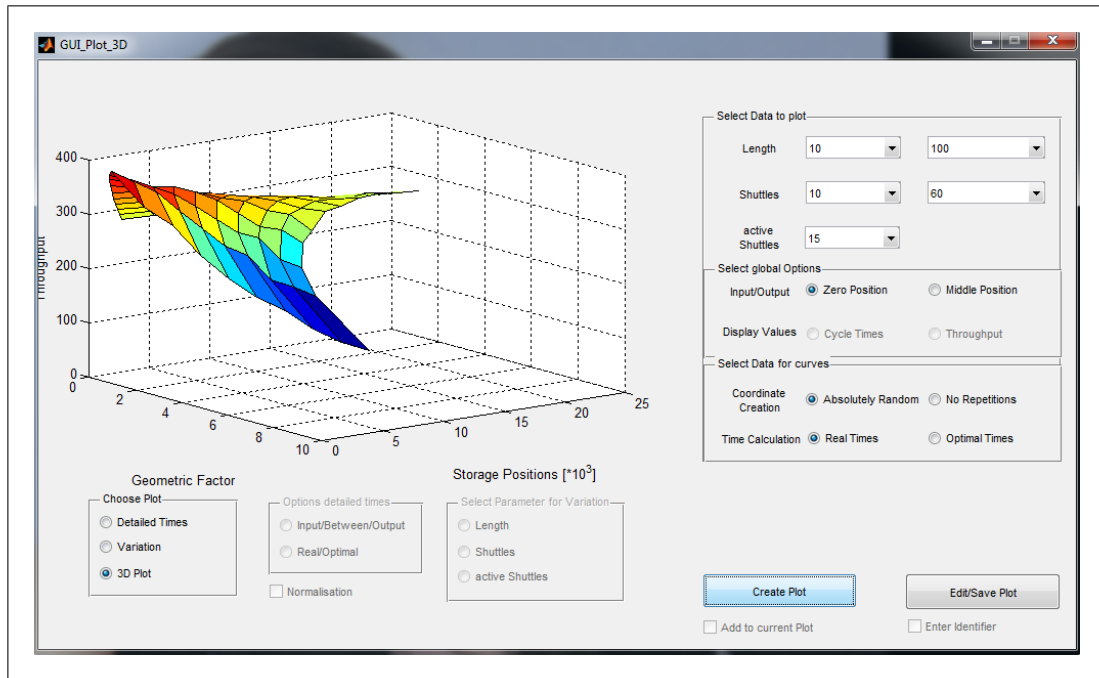


Abbildung 5.12: Die Benutzeroberfläche GUI Plot 3D

Dieses Fenster kann nur vom Hauptfenster aus geöffnet werden, da sonst die für die Darstellung notwendigen Daten nicht übergeben werden.

6 Ergebnisse und Interpretation

In diesem Kapitel wird erläutert, welche Werte die Simulationsergebnisse beinhalten und wie diese zustande kommen. Danach werden die Ergebnisse – grafisch aufbereitet – präsentiert und ausführlich diskutiert. Insbesondere soll dabei zuerst auf die Zusammensetzung einzelner Zyklen eingegangen werden und danach auf die Auswirkung von Parametervariationen. Außerdem sollen die Ergebnisse noch mit der Masterarbeit von Herrn Castells [Cas13], die parallel am Institut für Konstruktionswissenschaften und technische Logistik der Technischen Universität Wien entstanden ist, verglichen werden. Abschließend sollen mögliche Fehlerquellen und deren Auswirkungen sowie denkbare Lösungsansätze besprochen werden.

6.1 Die Simulationsergebnisse

In Kapitel 5, Funktionsweise der Simulation, wurde ausführlich darauf eingegangen, wie das Ergebnis eines einzelnen Zyklus berechnet wird. Eine Simulation umfasst nun n_{Sim} verschiedene Zyklen, die hintereinander durchlaufen werden. Am Ende erhält man also eine Lösung, die für jeden Zyklus die entsprechende Einlagerungs-, Zwischen- und Auslagerungszeit beziehungsweise die daraus resultierende Doppelspielzeit beinhaltet. Daraus wird mit Gleichung 6.1 die mittlere Doppelspielzeit $\overline{t_{DS}}$ berechnet:

$$\overline{t_{DS}} = \frac{1}{n_C} \cdot \sum_{i=1}^{n_C} t_{DS}(i) \quad (6.1)$$

Als Vergleichswerte für die Effizienz des simulierten Prozesses werden – neben den realen Doppelspielzeiten – auch die optimalen Doppelspielzeiten herangezogen. Diese entstehen durch Vernachlässigung der im Zyklus auftretenden Wartezeiten und lassen sich mit der Überlegung rechtfertigen, dass in einem realen System die Lagerverwaltungssoftware eben diese auftretenden Wartezeiten minimiert. Diese Werte werden ebenfalls sowohl für jeden einzelnen Zyklus als auch als mittlere optimale Doppelspielzeit $\overline{t_{DSopt}}$ gespeichert. Diese wird berechnet, wie in Gleichung 6.1 beschrieben.

Alternativ dazu werden auch die Umschlagleistungen für die mittleren realen und optimalen Doppelspielzeiten nach Gleichung 6.2 berechnet und im Ergebnis festgehalten:

$$R = \frac{3600}{\overline{t_{DS}}} \quad (6.2)$$

Zusätzlich werden die gesamten Koordinaten archiviert, um z. B. dieselbe Simulation mit veränderten System- oder Simulationsparametern wiederholen zu können. Ebenfalls wird

die ungefähre Anzahl der im System befindlichen Lagerpositionen nach Gleichung 6.3 berechnet:

$$n_{Pos} \approx \frac{l}{w} \cdot n_S \quad (6.3)$$

Weitere abgespeicherte Daten sind die Wartezeiten pro Zyklus, aus dem, in Gleichung 6.4 beschriebenen, einfachen Zusammenhang

$$t_W = t_{DS} - t_{DSopt} \quad (6.4)$$

Da, trotz einer großen Anzahl von verschiedene Simulationen, die Ergebnisse übersichtlich und nachvollziehbar verwaltet werden sollen, bietet MATLAB die Möglichkeit, Daten als Struktur abzuspeichern. Damit wird jedes Element eindeutig zugeordnet und ist leicht zu finden. Die Hierarchie leitet sich aus der in Unterabschnitt 5.4.2 beschriebenen Variationsreihenfolge ab und kann wie folgt dargestellt werden:

- Länge
- Anzahl der Shuttles
- Ebene der Übergabestation
- Anzahl der „aktiven Shuttles“
- Koordinatenerzeugung
 - zufällig
 - zufällig, ohne Wiederholungen

6.2 Die Definition der Parameter

Alle Simulationen wurden mit den in Tabelle 6.1 dargestellten konstanten Eingabeparametern berechnet. Um möglichst realitätsnahe Simulationsergebnisse zu erzielen, wurden die Systemparameter freundlicherweise von der Firma Knapp zur Verfügung gestellt.

Geometrie		System			Simulation
$h = 0,3 \text{ m}$	$v_L = 5 \frac{\text{m}}{\text{s}}$	$t_{I \rightarrow L} = 1,4 \text{ s}$	$t_{L \rightarrow O} = 1,4 \text{ s}$	$n_{\text{Sim}} = 1500$	
$w = 0,5 \text{ m}$	$v_S = 2 \frac{\text{m}}{\text{s}}$	$t_{L \rightarrow P} = 1,4 \text{ s}$	$t_{P \rightarrow L} = 1,4 \text{ s}$		
	$a_L = 7 \frac{\text{m}}{\text{s}^2}$	$t_{P \rightarrow S} = 5 \text{ s}$	$t_{S \rightarrow P} = 6,5 \text{ s}$		
	$a_S = 2 \frac{\text{m}}{\text{s}^2}$	$t_{S \rightarrow \text{Lag}} = 6,5 \text{ s}$	$t_{\text{Lag} \rightarrow S} = 5 \text{ s}$		

Tabelle 6.1: fixe Parameter der Simulation

Die anderen Parameter wurden variiert, um eine große Anzahl an verschiedenen Systemen abzubilden. Dazu wurden Systeme nach folgendem geometrischen Schema aufgebaut:

$$l = [10 \quad 20 \quad 30 \quad \dots \quad 100] \text{ m}$$

$$n_s = [5 \quad 10 \quad 15 \quad \dots \quad 60]$$

$$x_{in} = \frac{l}{2}$$

Für die Position der Übergabestelle wurden pro System zwei unterschiedliche Situationen dargestellt:

$$n_{S\ddot{U}} = 0 \quad \text{oder} \quad n_{S\ddot{U}} = \left\lceil \frac{n_S}{2} \right\rceil$$

Für die „aktiven Shuttles“ schließlich wurde das Maximum mit dem 1,5-fachen der Anzahl an vorhandenen Shuttles ($1,5 \cdot n_S$) gewählt, um die Simulation stabil zu halten. Bis zum Erreichen des Maximalwerts wird folgendes Schema verwendet:

$$n_{aS} = [2 \quad 4 \quad 5 \quad 6 \quad 8 \quad 10 \quad 15 \quad 20 \quad \dots \quad 1,5 \cdot n_S]$$

6.3 Analyse der Zyklen

Für die Betrachtung der einzelnen Zyklen sollen aus den generierten Daten drei Referenzsysteme ausgewählt werden, an Hand derer die Charakteristika der Simulationen erarbeitet werden sollen. Die Parameter dieser Systeme sind in Tabelle 6.2 zu sehen.

Name	l	n_S	$n_{\ddot{U}S}$	n_{aS}	Koordinaten
System 1	30 m	20	0	10	Zufall
System 2	60 m	20	0	10	Zufall
System 3	90 m	20	0	10	Zufall

Tabelle 6.2: Überblick Systeme

In der Folge wird jede grafische Darstellung bei ihrem ersten Auftreten mit einer Legende versehen. Diese gilt dann für alle nachfolgenden Abbildungen desselben Typs analog.

6.3.1 Die Systeme im Detail

System 1

Zu Beginn wird ein kleines System betrachtet. Bei einer Höhe von 6 m ist der Regalgang 30 m lang, die Übergabestation befindet sich auf der Position 0, also eine Ebene unter der ersten Shuttle-Ebene. Die Anzahl der „aktiven Shuttles“ wurde auf die Hälfte der vorhanden Shuttles festgesetzt und beträgt damit 10.

Abbildung 6.1 zeigt nun die Zusammensetzung der Doppelspielzeiten unterteilt in die Einlagerungs-, Zwischen-, und Auslagerungszeiten für jeden Zyklus. Jeder Peak steht für eine Wartezeit, wobei die Farbe angibt, ob diese während eines Einlagerungsvorganges oder eines Zwischenzyklus auftritt. Die Linie für die Auslagerungszeiten ist annähernd konstant, da hier keine Wartezeiten auftreten können. Die Linie für die mittlere Einlagerungszeit liegt über den anderen beiden, was auf die Definition dieses Zyklus zurückzuführen ist: Es werden hier nämlich zwei Mal Übergabezeiten eingerechnet. Aus dieser Grafik lässt sich

also ablesen, dass vereinzelt Wartezeiten auftreten, wobei sich diese ungefähr zu gleichen Teilen auf die beiden Vorgänge aufteilen.

In Abbildung 6.2 ist derselbe Vorgang zu sehen, diesmal werden aber die einzelnen Zeiten normiert, wie in Gleichung 6.5 zu sehen ist.

$$t^* = \frac{t}{t_{DS}} \tag{6.5}$$

Dabei werden die Zeiten aller Einzelzyklen auf die mittlere Doppelspielzeit bezogen. Mit dieser Darstellung lassen sich verschiedene Systeme einfacher vergleichen, das Verhalten ist aber dasselbe wie in Abbildung 6.1.

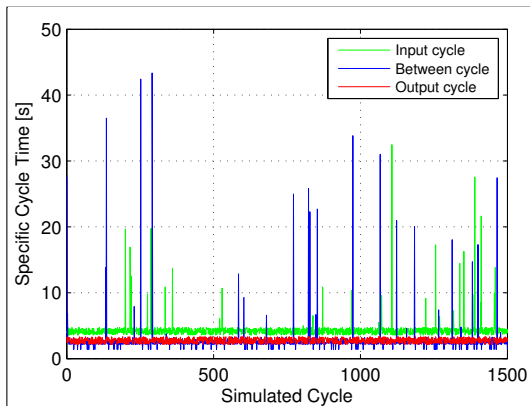


Abbildung 6.1: System 1 – Zykluszeiten

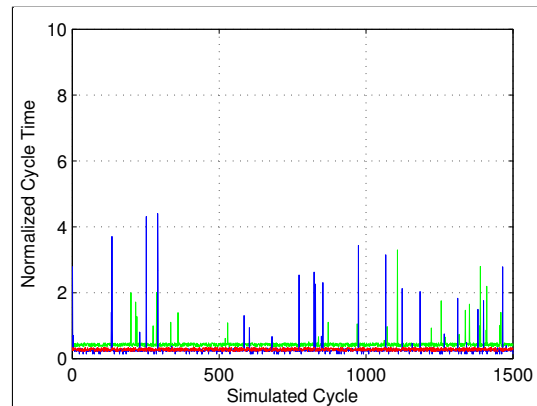


Abbildung 6.2: System 1 – Zykluszeiten normiert

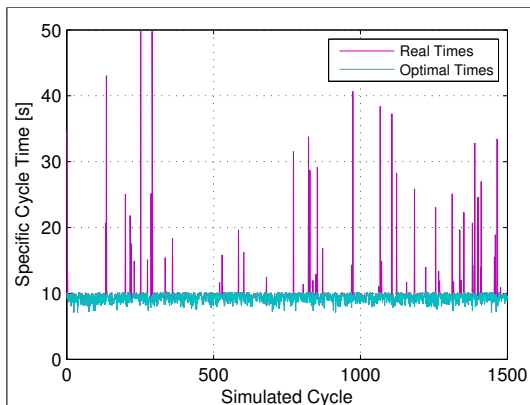


Abbildung 6.3: System 1 – Doppelspielzeiten

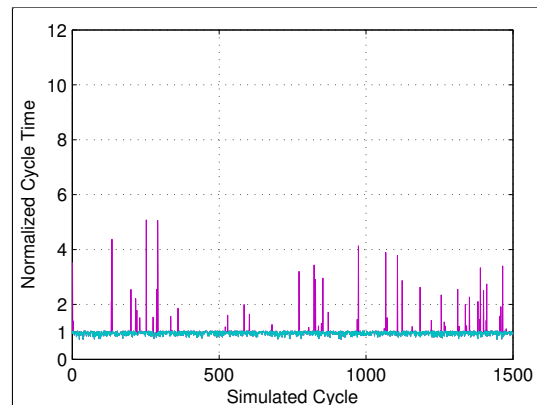


Abbildung 6.4: System 1 – Doppelspielzeiten normiert

Im Gegensatz zu den zuvor besprochenen Abbildungen zeigt Abbildung 6.3 die Aufteilung in die tatsächliche Doppelspielzeit und die optimale Zeit. Letztere fluktuiert nur schwach und ist, aufgrund ihrer Definition, ausschließlich von der Höhe des Systems abhängig. In den realen Zeiten entspricht wieder jeder Peak einer Wartezeit, wobei die Entstehung nicht mehr nachvollziehbar ist. Abbildung 6.4 zeigt wieder die nach Gleichung 6.5 normierten Doppelspielzeiten. Da immer die mittlere Doppelspielzeit als Referenz gewählt wird,

entspricht diese der Bezugslinie bei $t^* = 1$. Da sich die optimalen Zeiten annähernd auf dieser Linie befinden, lässt sich daraus schließen, dass die mittlere Doppelspielzeit sehr nah an der optimalen liegt und die auftretenden Wartezeiten insgesamt keine große Auswirkung haben.

Dieses Verhalten lässt sich einfach erklären: Da das System relativ klein ist, sorgen die 10 „aktiven Shuttles“ dafür, dass die Auslagerungspuffer beim Eintreffen des Liftes sehr oft schon gefüllt sind. Wartezeiten treten in der Regel auf, wenn dieselbe Ebene öfters hintereinander mit einem Auftrag belegt wird. Sollte es in so einem Fall zu Wartezeiten kommen, sind diese relativ niedrig, da die Shuttles den Weg relativ schnell zurücklegen können

System 2

In diesem System werden alle Parameter konstant gehalten, nur die Länge wird auf 60 m ausgedehnt. Die Abbildung 6.6 zeigt die Zeiten der einzelnen Zyklen. Im Vergleich zu System 1 fällt sofort auf, dass sich die Anzahl der auftretenden Wartezeiten wesentlich erhöht hat. Auch die maximale Länge der Zyklen hat sich von 50 s in System 1 auf 110 s mehr als verdoppelt. Außerdem lässt sich beobachten, dass die maximalen Wartezeiten im Einlagerungszyklus wesentlich geringer sind als die im Zwischenzyklus. Dieses Verhalten lässt sich auch in Abbildung 6.6 beobachten, da hier wieder die Normierung durchgeführt wurde. Vergleicht man nun diese Abbildung direkt mit Abbildung 6.2 so sieht man auch die verhältnismäßig höheren Peaks im selben Maßstab.

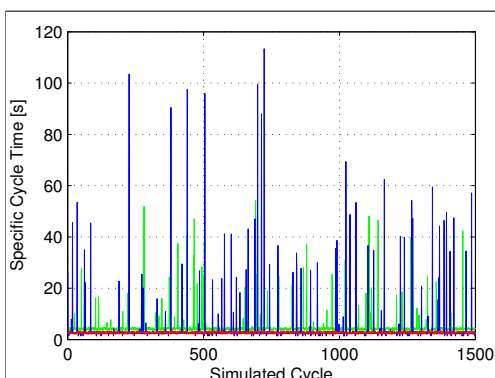


Abbildung 6.5: System 2 – Zykluszeiten

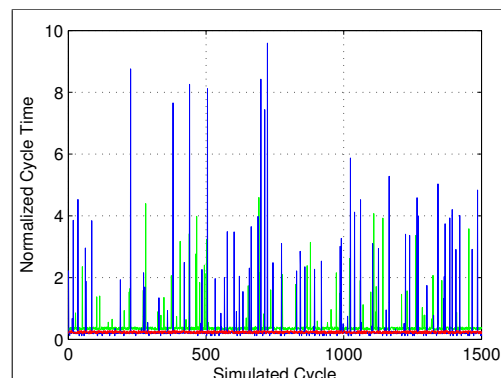


Abbildung 6.6: System 2 – Zykluszeiten normiert

Abbildung 6.7 zeigt erneut die tatsächlichen und die optimalen Doppelspielzeiten. Auch hier zeigen sich die Auswirkungen des erhöhten Auftretens von Wartezeiten deutlich, ebenso die Erhöhung der maximalen Zeiten. Die Linie der optimalen Zeiten bleibt aber beim selben Wert wie in System 1. Betrachtet man nun die normierten Doppelspielzeiten in Abbildung 6.8, kann schon eine Verschiebung der optimalen Zeiten unter die Bezugslinie festgestellt werden. Das zeigt, dass die auftretenden Doppelspielzeiten im Vergleich zu den optimalen schon erhöht ist und sich das System nicht im optimalen Bereich befindet.

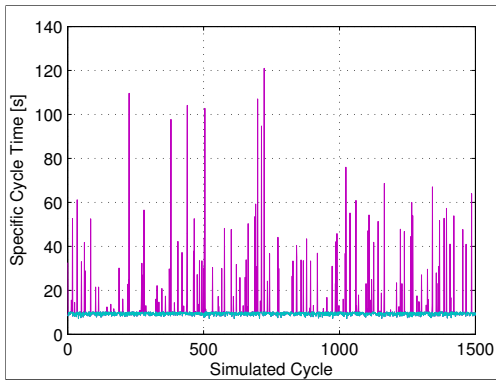


Abbildung 6.7: System 2 – Doppelspielzeiten

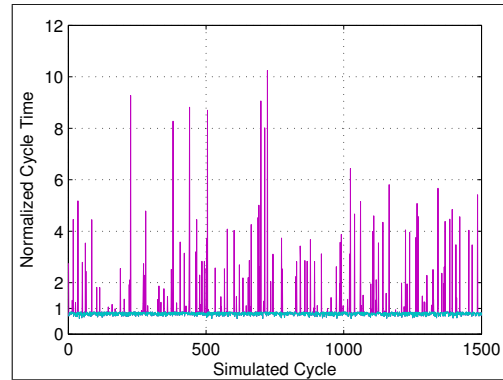


Abbildung 6.8: System 2 – Doppelspielzeiten normiert

Die Erklärung dieses Verhaltens leitet sich nun direkt aus dem Vergleich mit System 1 ab. Die Erhöhung der Länge bei konstanter Höhe führt dazu, dass die Fahrzeiten der Shuttles ansteigen. Dadurch kommt es öfters zu Wartezeiten im Zwischenzyklus, da die angeforderten Lagereinheiten noch nicht im Puffer eingetroffen sind. Wird nun eine Ebene zwei Mal hintereinander angesprochen, muss der Lift sehr lange auf das Shuttle warten. Außerdem führen die erhöhten Fahrzeiten auch dazu, dass öfters ein Einlagerungspuffer nicht rechtzeitig geleert werden kann. Dadurch erhöht sich auch das Auftreten von Wartezeiten im Einlagerungszyklus. Diese fallen aber nicht so hoch aus, da der Lift schon entladen werden kann, sobald der Puffer geleert wurde. Die Fahrzeit des Shuttles zum Lagerort ist nicht mehr relevant für den Hauptprozess. Im Gegensatz dazu schlägt sich bei einer Auslagerung die gesamte Transportzeit des Shuttles, von der aktuellen Position zur Lagerposition und retour, in der Wartezeit nieder.

System 3

Im dritten System wird die Länge nochmals um 30 m auf 90 m erhöht, alle anderen Parameter bleiben wieder konstant. Dadurch wird, wie in Abbildung 6.9 dargestellt, die Frequenz, mit der Wartezeiten auftreten, weiter erhöht. Auch die maximal auftretenden Zykluszeiten steigen weiter an und liegen nun schon bei 150 s. Die normierten Doppelspielzeiten in Abbildung 6.10 geben dieses Verhalten ebenfalls wieder. Hier ist anzumerken, dass, obwohl die maximalen Zeiten gestiegen sind, das Verhältnis zur mittleren Doppelspielzeit in etwa dasselbe ist wie in System 2. Diese Tendenz kann durch die ebenfalls stark ansteigende mittlere Doppelspielzeit erklärt werden.

Die realen und optimalen Doppelspielzeiten, die in Abbildung 6.11 dargestellt werden, liegen ebenfalls auf einem höheren als bei den beiden anderen Systemen, bei Spitzenzeiten bis zu 180 s, wenn in einem Zyklus sowohl bei der Einlagerung als auch im Zwischenzyklus eine Wartezeit auftritt. In Abbildung 6.12 liegen die optimalen Zeiten nun schon deutlich unter der Bezugslinie, was bedeutet, dass die mittlere Doppelspielzeit sich noch weiter von einem günstigen Bereich entfernt hat.

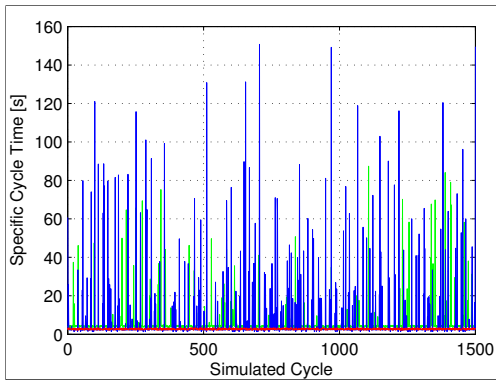


Abbildung 6.9: System 3 – Zykluszeiten

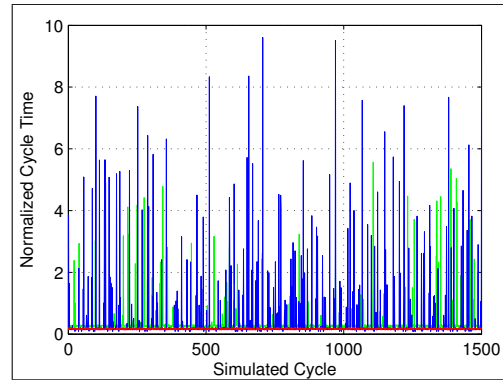


Abbildung 6.10: System 3 – Zykluszeiten normiert

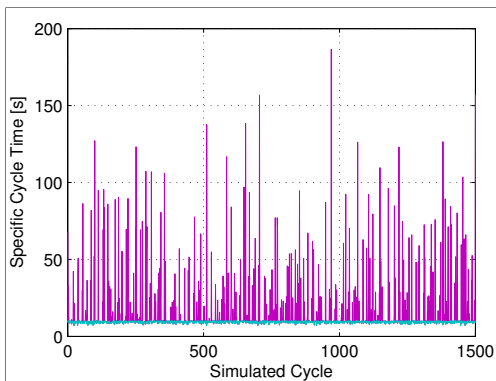


Abbildung 6.11: System 3 – Doppelspielzeiten

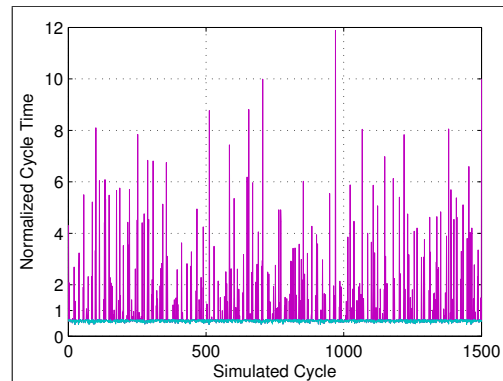


Abbildung 6.12: System 3 – Doppelspielzeiten normiert

Das Verhalten dieses Systems ist eine weitere Steigerung der schon bei System 2 beobachteten Tendenz. Die nochmals erhöhten Fahrzeiten der Shuttles führen zu denselben Effekten wie schon beschrieben und erhöhen weiter die Häufigkeit, mit der Wartezeiten auftreten sowie deren Maximalwerte.

6.3.2 Vergleich der Grundsysteme

Nun sollen die, bei den einzelnen Systemen ausgeführten, Erklärungen des Systemverhaltens noch durch weiteres Zahlenmaterial unterlegt werden. In den zuvor behandelten Systemen wurden die Zeiten der einzelnen Zyklen betrachtet. Zuerst soll Abbildung 6.13 näher erläutert werden, in der die mittleren Zykluszeiten aufgetragen sind. Daraus lässt sich die durchschnittliche Doppelspielzeit der einzelnen Systeme und ihre Zusammensetzung ermitteln. Betrachtet man nun die Tendenzen, so zeigt sich, dass die Einlagerungszeit leicht ansteigt und die Auslagerungszeit annähernd konstant bleibt. Die Zwischenzeit steigt hingegen stark an. In Abbildung 6.14 sind nun die gemittelten Einzelzyklen der drei Systeme im Verhältnis zur mittleren Doppelspielzeit aufgetragen. Betrachtet man nun zuerst die Einlagerung und die Auslagerung, so zeigt sich, dass deren Anteil von System 1

bis System 3 kontinuierlich abnimmt, während bei den Zeiten des Zwischenzyklus genau die umgekehrte Tendenz erkennbar ist.

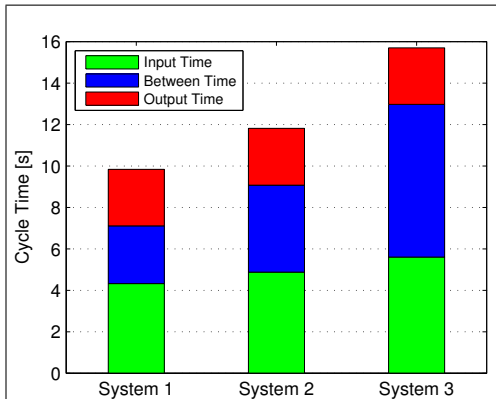


Abbildung 6.13: Zusammensetzung der Zykluszeiten – absolut

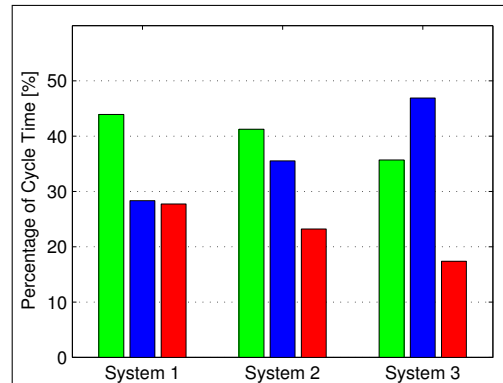


Abbildung 6.14: Zusammensetzung der Zykluszeiten – relativ

Um diesen Trend erklären zu können, werden im nächsten Schritt die Zeiten nicht nach ihrer Zuordnung zu den Zyklen, sondern nach ihrem Entstehen gruppiert. Es ergibt sich folgende Einteilung:

- Übergabezeiten
- Fahrzeit des Liftes
- Wartezeit

Die Übergabezeiten, die im Hauptzyklus auftreten, werden als Systemparameter festgelegt und sind damit unabhängig vom aktuellen geometrischen Setup. Die Fahrzeit des Liftes hängt allein von der Höhe ab und ist damit ebenfalls in den drei betrachteten System konstant. Einzig die Wartezeiten können sich verändern, sie steigen stark an. Abbildung 6.15 zeigt die anteilmäßige Zusammensetzung der Doppelspielzeit in Bezug auf ihre Entstehung.

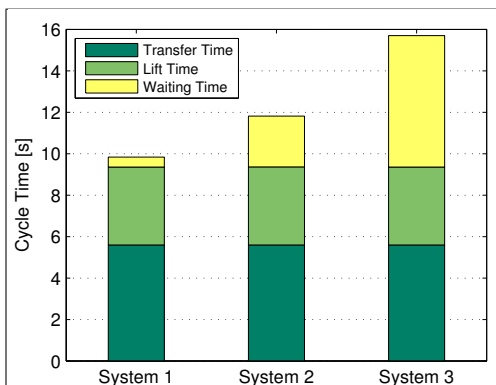


Abbildung 6.15: Entstehung der Zykluszeiten – absolut

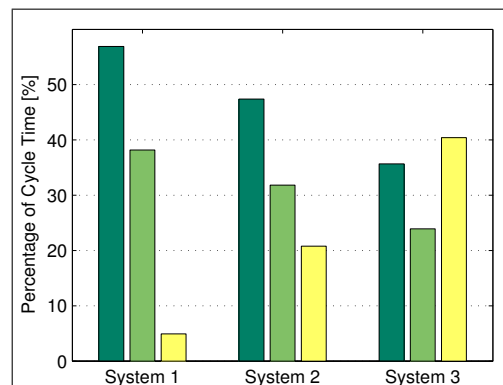


Abbildung 6.16: Entstehung der Zykluszeiten – relativ

In Abbildung 6.16 werden abschließend die prozentualen Anteile an der mittleren Doppel-

spielzeit dargestellt. Der Anteil der Wartezeiten steigt also von 5 % in System 1 auf 40 % in System 3.

Mit diesen Grafiken bestätigen sich die Erklärungen bei den Einzelsystemen. Durch die ungünstigen Systemparameter treten immer höhere Wartezeiten auf, welche sich immer stärker auf den Zwischenzyklus auswirken. Dieser Zusammenhang ist der ausschlaggebende Faktor für die wachsenden mittleren Doppelspielzeiten.

6.3.3 Vergleich von modifizierten Grundsystemen

In der Folge soll beschrieben werden, wie sich die Antwort der Systeme verändert, wenn zusätzlich zur Länge ein weiterer Parameter verändert wird. Da in Unterabschnitt 6.3.2 schon ausführlich auf die Entstehung der Zykluszeiten eingegangen wurde, werden nur mehr die Doppelspielzeiten für System 1 und System 3 dargestellt, da diese die Tendenz ausreichend dokumentieren.

Änderung der Höhe

In der Folge sollen die drei Grundsysteme erneut betrachtet werden, jedoch wird die Anzahl der Shuttles auf 40 verdoppelt, was zu einer Höhe von 12 m führt. In Abbildung 6.17 sind die normierten mittleren Doppelspielzeiten in System 1 abgebildet. Vergleicht man diese nun mit dem Grundsystem (siehe Abbildung 6.4), so sieht man, dass fast keine Wartezeiten auftreten. Dasselbe Bild zeigt sich auch, wenn man die Abwandlung von System 3 (zu sehen in Abbildung 6.18) mit dem Grundsystem (siehe Abbildung 6.12) vergleicht. Die Häufigkeit, mit der Wartezeiten auftreten, hat deutlich abgenommen und die optimale Zeit liegt sehr nah an der Bezugslinie.

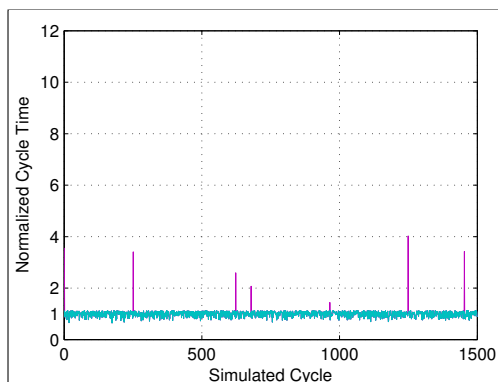


Abbildung 6.17: System 1 – Doppelspielzeiten normiert $n_S = 40$

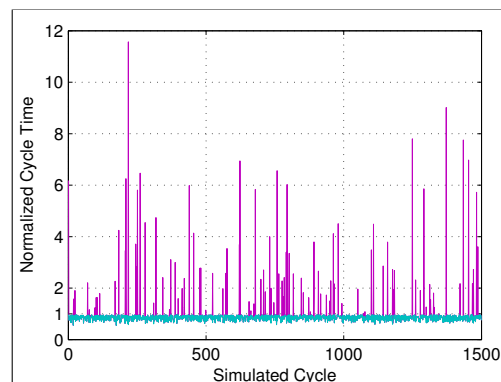


Abbildung 6.18: System 3 – Doppelspielzeiten normiert $n_S = 40$

Stellt man nun der zeitlichen Zusammensetzung der ursprünglichen Systeme (siehe Abbildung 6.13) die in Abbildung 6.19 dargestellten gegenüber, so zeigen sich einige Unterschiede: Die Doppelspielzeit liegt nunmehr bei System 1 bei 11 s statt bei 10 s, sie steigt

aber nicht so stark an und liegt in System 3 mit 13 s schon deutlich unter der des Standard-systems. Betrachtet man die Entstehung der Zeiten in Abbildung 6.20, erkennt man, dass die Wartezeit in den Systemen 1 und 2 nur einen sehr kleinen Teil der Doppelspielzeit ausmachen. Diese setzt sich zu ungefähr gleichen Teilen aus den Übergabezeiten und der Fahrzeit des Liftes zusammen. Auch in System 3 liegt der Anteil der Wartezeit nur bei 15%, wohingegen bei der nicht modifizierten Variante die Wartezeit schon 40% der Spielzeit einnimmt.

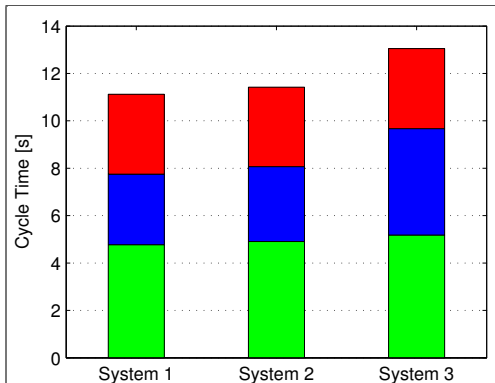


Abbildung 6.19: Zusammensetzung der Zykluszeiten – absolut $n_S = 40$

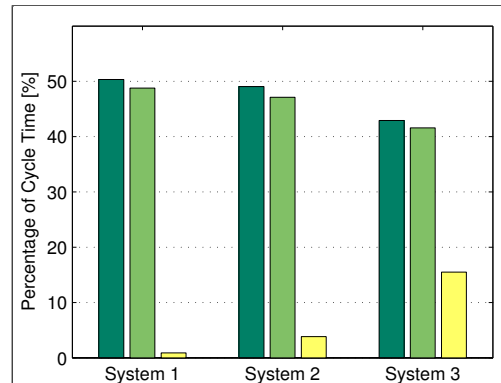


Abbildung 6.20: Entstehung der Zykluszeiten – relativ $n_S = 40$

Diese Tendenzen lassen sich durch folgenden Überlegungen erklären: Durch die Erhöhung der Anzahl der Shuttles verteilen sich die Aufträge über mehr Ebenen und die Wahrscheinlichkeit, dass eine Ebene mehrmals hintereinander aktiviert wird, sinkt dadurch. Dadurch treten in einer Simulation wesentlich weniger Wartezeiten auf. Gleichzeitig führt die größere Höhe des Systems zur Erhöhung der durchschnittlichen Fahrzeit des Liftes pro Zyklus, dadurch steigt die mittlere Doppelspielzeit bei kurzen Systemen, bei längeren Systemen gleicht sich diese aber mit den verminderten Wartezeiten aus und die Doppelspielzeiten sinken im direkten Vergleich.

Änderung der „aktiven Shuttles“

Um die Auswirkungen des Parameters „aktive Shuttles“ beobachten zu können, wird die Anzahl auf die Hälfte, also auf 5 reduziert. Dadurch werden nur mehr 5 Auslagerungsvorgänge vorbereitet. In Abbildung 6.21 sieht man wieder die normierte Doppelspielzeit. Verglichen mit den Zeiten des Grundsystems (siehe Abbildung 6.4) so ist eine deutliche Steigerung der Anzahl der auftretenden Wartezeiten zu beobachten. Dieses Verhalten zeigt sich ebenfalls bei einer Länge von 90 m, wie in Abbildung 6.22 dargestellt ist. Auffällig dabei ist, dass, obwohl fast in jedem Zyklus Wartezeiten auftreten, die Höhe der maximalen Wartezeit geringer ist als beim Ausgangssystem (siehe Abbildung 6.12). Die optimale Doppelspielzeit liegt aber ganz deutlich unter der Bezugslinie, ein weiteres Zeichen dafür, dass das System weit entfernt vom optimalen Zustand arbeitet.

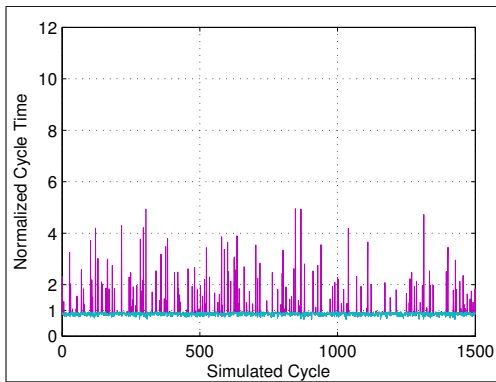


Abbildung 6.21: System 1 – Doppelspielzeiten – normiert – $n_{aS} = 5$

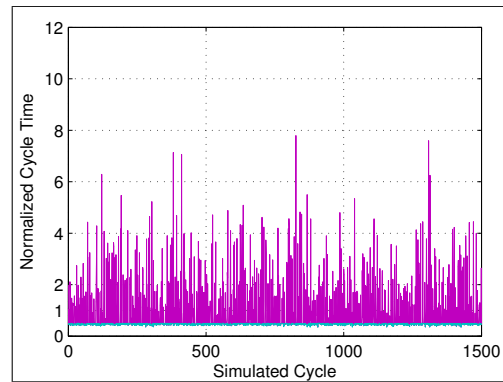


Abbildung 6.22: System 3 – Doppelspielzeiten – normiert – $n_{aS} = 5$

Das wird auch deutlich, wenn man die in Abbildung 6.23 dargestellte Zusammensetzung der mittleren Doppelspielzeiten bzw. ihre Werte betrachtet. Ist der Unterschied zu den Ausgangssystemen bei System 1 noch sehr gering, ungefähr 1 s, so liegt er bei System 3 schon bei über 4 s. Betrachtet man nun die Entstehung der Zeiten in Abbildung 6.24 so sieht man, dass dieser Unterschied durch die überproportional große Wartezeit entsteht, die in System 3 schon über 50 % der mittleren Doppelspielzeit einnimmt.

Durch die verringerte Anzahl an vorbereiteten Auslagerungen kommt es wesentlich öfter zu Wartezeiten, was die durchschnittlichen Doppelspielzeiten erhöht. In großen Systemen reicht die Vorlaufzeit nicht aus, um die Auslagerung rechtzeitig abzuschließen. Da fast in jedem Zyklus Wartezeiten auftreten, wirkt sich der in Unterabschnitt 5.2.4 beschriebene Effekt der Zeitverschiebung sehr stark aus, da das System sich nie im optimalen Bereich befindet, wo die Abstände der Zeitüberprüfung sehr gering sind. Daher ist die Zusammensetzung der Werte in System 1 und 2 wahrscheinlich genauer als die in System 3, wobei die Entwicklung qualitativ sicherlich wirklichkeitstreu ist.

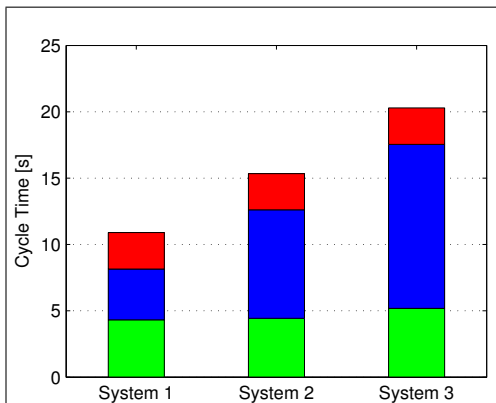


Abbildung 6.23: Zusammensetzung der Zykluszeiten – absolut – $n_{aS} = 5$

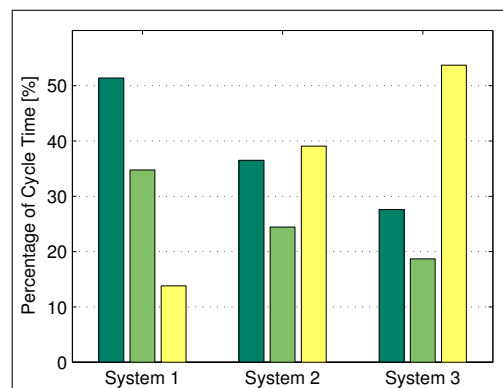


Abbildung 6.24: Entstehung der Zykluszeiten – relativ – $n_{aS} = 5$

Änderung der Übergabeposition

Durch das Verschieben der Übergabeposition in die mittlere Ebene des Systems, also in Ebene 10, soll die Auswirkung dieses Parameters gezeigt werden. Da sich die Darstellung der normierten Doppelspielzeiten nur sehr wenig von der der ursprünglichen Systeme unterscheidet, wird auf entsprechende Abbildungen verzichtet und auf die Abbildungen 6.4 und 6.12 verwiesen. Die Zusammensetzung weist jedoch Unterschiede auf, die in der Folge erläutert werden. Vergleicht man nun die Zykluszeiten im modifizierten System (wie in Abbildung 6.25) dargestellt mit dem Grundsystem, so ist nur in System 1 eine geringe Verbesserung der mittleren Doppelspielzeit zu erkennen, während die anderen beiden Systeme sich in etwa demselben Bereich befinden. Abbildung 6.26 zeigt, dass im Vergleich mit dem Originalsystem die Anteile der Liftzeiten um durchschnittlich 5% verringert sind.

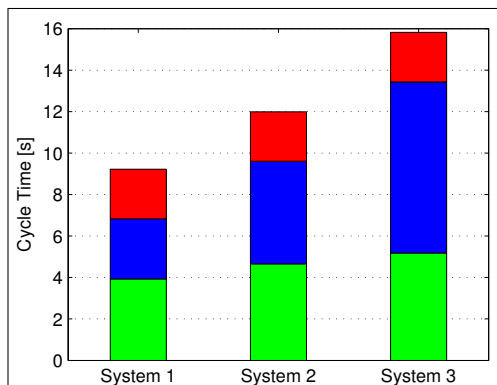


Abbildung 6.25: Zusammensetzung der Zykluszeiten – absolut –
 $n_{\text{ÜS}} = 10$

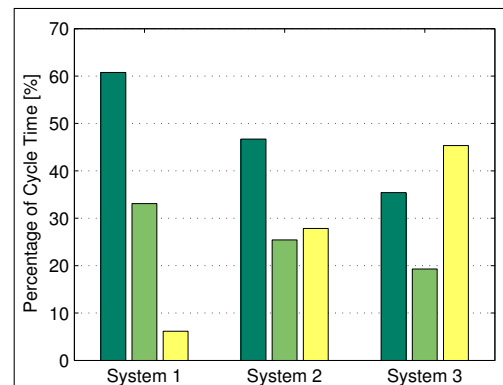


Abbildung 6.26: Entstehung der Zykluszeiten – relativ –
 $n_{\text{ÜS}} = 10$

Da in den beschriebenen Systemen nur 20 Shuttles vorhanden sind, wirkt sich die Veränderung der Übergabeposition nicht stark aus. Bei einem System mit mehr Ebenen wäre das Einsparungs-Potential erwartungsgemäß höher, als hier beobachtet werden kann.

Änderung der Koordinatenerzeugung

Als letztes soll noch beschrieben werden, welche Auswirkung die Änderung der Koordinatenerzeugung von ‚völlig zufällig‘ auf ‚zufällig, aber ohne direkte Wiederholungen‘ hat. Dazu werden wieder nur die Vergleiche der Zusammensetzung der Doppelspielzeiten herangezogen, da sich ein relevanter Unterschied erst hier zeigt. Stellt man nun die Doppelspielzeiten dieser Modifikation aus Abbildung 6.27 denen der ursprünglichen Systeme gegenüber, so lässt sich feststellen, dass die aktuellen Werte unter den originalen liegen. Die Differenz steigert sich von System 1, bei dem diese nur wenige Zehntel Sekunden beträgt, über 1 s bei System 2 bis hin zu 1,5 s bei System 3.

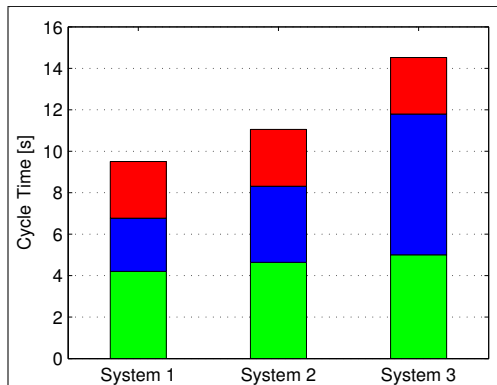


Abbildung 6.27: Zusammensetzung der Zykluszeiten – absolut – ohne Wh.

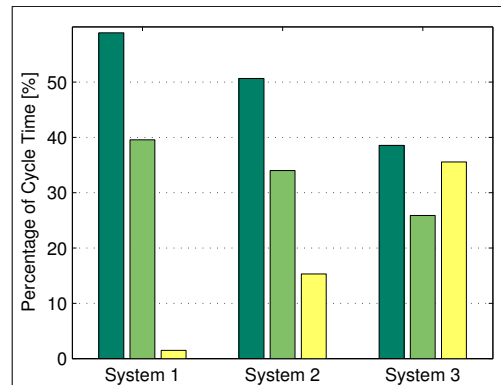


Abbildung 6.28: Entstehung der Zykluszeiten – relativ – ohne Wh.

Die Senkung der Doppelspielzeit wird, wie aus Abbildung 6.28 ersichtlich, durch Verringerung der Wartezeiten während der Zyklen erreicht. Diese liegt durchschnittlich 5% unter den Vergleichswerten der ursprünglichen Systeme.

Durch das Verhindern einer unmittelbar hintereinander folgenden Aktivierung einer Shuttle-Ebene wird die Ursache für die höchsten Wartezeiten beseitigt. Es können immer noch Wartezeiten auftreten, dies passiert aber seltener und im Schnitt in verringertem Ausmaß. Diese Vorgehensweise wirkt sich besonders in Systemen mit großen Längen aus, da die Wartezeiten dort einen größeren Anteil der Doppelspielzeit ausmachen.

6.4 Analyse der Variationen

Bis jetzt wurden exemplarisch einzelne Systeme sowie Tendenzen und Verhalten der Systeme besprochen. In diesem Abschnitt soll eine Vielzahl von Simulationsergebnissen zur Verifizierung dieser Tendenzen herangezogen werden. Durch Variation einzelner Parameter bei verschiedenen Systemkonfigurationen sollen übersichtliche Graphen erstellt werden.

6.4.1 Variation der Länge

Zunächst wird untersucht, inwiefern sich die vorangegangenen Beobachtungen aus den detailliert besprochenen Systemen 1 bis 3 auch auf Systeme anderer Länge anwenden lassen. Dazu werden auf der Abszisse die Länge der Systeme und auf der Ordinate die mittleren Doppelspielzeiten aufgetragen. Abbildung 6.29 zeigt die Zeiten für Systeme von Längen zwischen 10 m und 100 m. Die weiteren Parameter der Simulation: Anzahl der Shuttles beträgt 20, die der „aktiven Shuttles“ 10 und die Übergabestelle befindet sich in Ebene 0. In Abbildung 6.30 ist die Anzahl der Shuttles auf 40 erhöht. Die oberste Kurve zeigt die Werte bei völlig zufälliger Koordinatenerzeugung, darunter sind als Vergleich die

Zeiten bei der Variante ohne direkte Wiederholungen aufgetragen. Als Referenz für die Größenordnung der Änderung ist außerdem die optimale Zeit abgebildet.

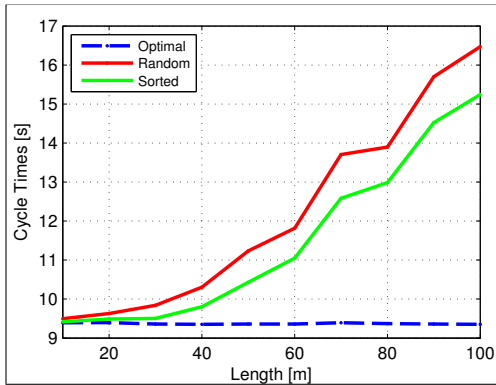


Abbildung 6.29: Vergleich der Koordinatenerzeugung – $n_S = 20; n_{aS} = 10$

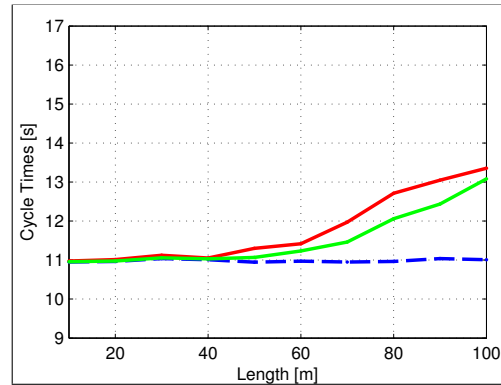


Abbildung 6.30: Vergleich der Koordinatenerzeugung – $n_S = 40; n_{aS} = 10$

Betrachtet man nun die Entwicklung der Doppelspielzeiten in Abbildung 6.29 so zeigt sich, dass diese schon ab einer Länge von 20 m kontinuierlich ansteigen. Die Änderung der Koordinatenerzeugung hat zur Folge, dass die sortierten durchschnittlichen Doppelspielzeiten unterhalb der zufällig generierten Zeiten liegen; der qualitative Verlauf bleibt aber in etwa derselbe. Die optimale Zeit ist konstant, da sie nur von der Höhe des Systems abhängt. Die Erklärung für diese Tendenz wurde schon in Unterabschnitt 6.3.1 erarbeitet: Der Anstieg der Wartezeiten wirkt sich entscheidend auf die Doppelspielzeit aus, durch die größeren Längen treten längere Wartezeiten auf.

Im Vergleich dazu zeigt Abbildung 6.30, dass bei einer größeren Anzahl von Shuttles eine Abweichung von der optimalen Zeit erst bei einer Länge von 50 m eintritt. Auch danach steigt die mittlere Doppelspielzeit nicht so stark an wie in Abbildung 6.29. Die zweite Kurve liegt wieder unter der der zufälligen Koordinaten, der Unterschied ist im Vergleich aber ebenfalls geringer. Durch die größeren Höhenunterschiede steigt aber auch die optimale Zeit an, sodass die mittleren Doppelspielzeiten auch in dem Bereich unter 50 m über den in Abbildung 6.29 dargestellt liegen.

Als nächstes soll die Auswirkung der Position der Übergabestelle erläutert werden. In den Abbildungen 6.31 und 6.32 werden jeweils die mittleren Doppelspielzeiten und die optimalen Zeiten für die Position der Übergabestelle in Ebene 0 und in der Mitte des Liftes dargestellt. Alle anderen Parameter verändern sich im Vergleich zu den bisherigen Betrachtungen nicht.

In Abbildung 6.31 wird wieder ein System mit 20 Shuttles betrachtet. Wird die Position der Übergabestelle in die Ebene 10 gelegt, so sinken dadurch die Fahrzeiten des Liftes. Daraus resultieren die niedrigeren optimalen Zeiten im Vergleich zu einer Übergabestelle in Ebene 0. Außerdem sieht man, dass die mittleren Doppelspielzeiten sinken, solange sich das System in einem Bereich nahe der optimalen Zeiten bewegt. Nimmt die Wartezeit überhand, so lässt sich keine wesentliche Verbesserung mehr feststellen.

Der selbe Trend lässt sich auch in Abbildung 6.32 beobachten. Solange die Doppelspielzeiten nicht weit von der optimalen Zeit abweichen, erreicht man durch die Veränderung der

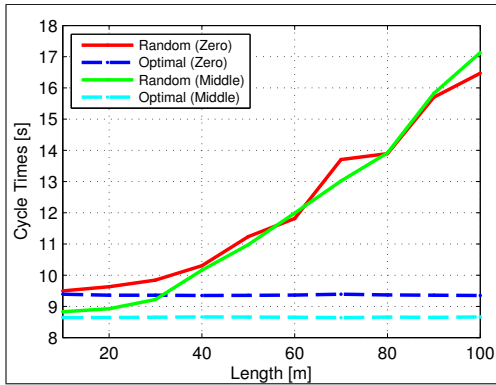


Abbildung 6.31: Vergleich der Übergabestelle – $n_S = 20; n_{aS} = 10$

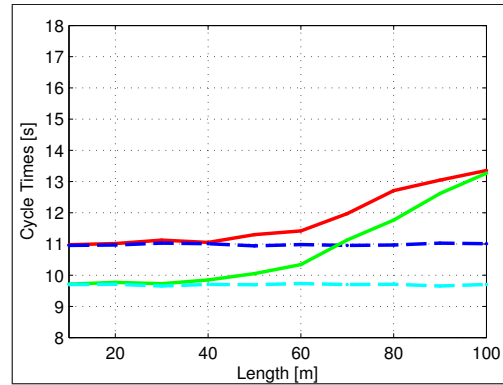


Abbildung 6.32: Vergleich der Übergabestelle – $n_S = 40; n_{aS} = 10$

Position der Übergabestelle eine Optimierung des Systems. Hier liegt der Bereich, in dem die beiden Zeiten ähnlich sind, außerhalb der simulierten Werte.

Abschließend soll die Auswirkung der „aktiven Shuttles“ auf die mittleren Doppelspielzeiten erörtert werden. Zu diesem Zweck zeigen die Abbildungen 6.33 und 6.34 die Änderung der Zeiten für 5 - 30 „aktive Shuttles“. Als Referenz dient wieder die optimale Doppelspielzeit.

In Abbildung 6.33 wird der Verlauf der mittleren Doppelspielzeiten für ein System mit 20 Ebenen dargestellt. Während bei 5 „aktiven Shuttles“ bei Erhöhung der Länge noch sehr hohe Doppelspielzeiten auftreten, können diese durch eine größere Anzahl an „aktiven Shuttles“ noch reduziert werden. Der Unterschied bei mehr als 10 „aktiven Shuttles“ ist eher gering und wird sich erst in einem Bereich, der nicht simuliert wurde, vollständig ausbilden. Es bleibt aber festzustellen, dass die Doppelspielzeiten bei 30 „aktiven Shuttles“ durchgehend höher sind, als diejenigen bei nur 20. Hier kann damit argumentiert werden, dass aufgrund der Überauslastung, es sind ja nur 20 Ebenen vorhanden, zusätzliche Wartezeiten entstehen, da alle Shuttles die ganze Zeit mit Auslagerungen beschäftigt sind. Die Abbildung 6.34 zeigt nun dieselben Graphen, jedoch werden hier 40 Ebenen betrachtet.

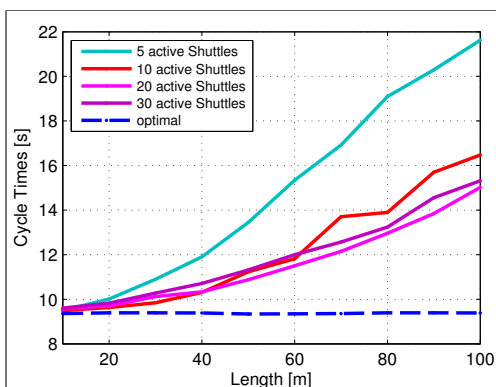


Abbildung 6.33: Vergleich der „aktiven Shuttles“ – $n_S = 20$

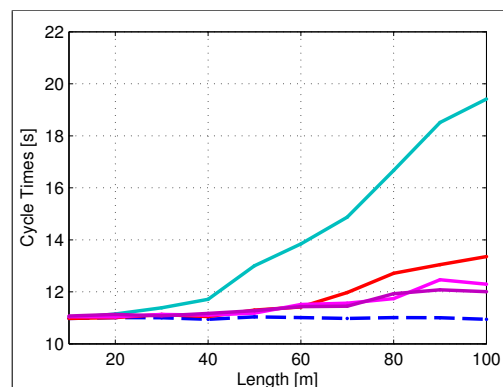


Abbildung 6.34: Vergleich der „aktiven Shuttles“ – $n_S = 40$

Die generellen Tendenzen bleiben dieselben, jedoch kann man bei Systemen ab einer Länge von 60 m feststellen, dass eine höhere Anzahl an „aktiven Shuttles“ eine weitere Verbesserung der Doppelspielzeiten bewirken kann. Ob die Anzahl aber auf 20 oder 30 erhöht wird, scheint hingegen keine signifikante Auswirkung zu haben. Hier liegt die Anzahl der „aktiven Shuttles“ noch unter der der Ebenen, dadurch kommt es nicht zu dem vorher beschriebenen Effekt.

6.4.2 Variation der Shuttles

Nachdem die Auswirkungen der Länge auf die mittleren Doppelspielzeiten ausführlich besprochen wurden, soll nun auf den Einfluss der Anzahl der Shuttles eingegangen werden. Dazu werden Systeme mit 10 bis 60 Ebenen bei verschiedenen Längen betrachtet, um die Tendenzen der Systeme zu bestätigen. Betrachtet man nun Abbildung 6.35, so zeigt sich eindeutig der schon beschriebene Effekt der steigenden optimalen Zeiten, wenn die Anzahl der Shuttles erhöht wird. Hier werden die mittleren Doppelspielzeiten für Systeme mit der Länge von 40 m und 10 „aktiven Shuttles“ dargestellt. Der Unterschied zwischen den zufällig generierten Koordinaten und der vorsortierten Variante ohne direkte Wiederholungen tritt nur in Systemen geringer Höhe auf. Ist nur mehr die Fahrzeit des Liftes ausschlaggebend für die mittlere Doppelspielzeit, das System befindet sich also im Bereich ohne Wartezeiten bei 100 %iger Auslastung des Liftes, so bringt auch die Elimination der Wiederholungen keine weitere Verbesserung mit sich. Aus diesen Graphen lässt sich nun das Minimum der mittleren Doppelspielzeit für dieses System bestimmen, es liegt bei ungefähr 20 Shuttles.

In Abbildung 6.36 wird derselbe Sachverhalt für Systeme mit einer Länge von 80 m dargestellt. Hier lässt sich wieder beobachten, dass ein System mit großer Länge, aber nur wenigen Ebenen hohe Wartezeiten verursacht. Die durch die Koordinatenerzeugung entstandene Differenz zeigt wieder denselben qualitativen Verlauf, bei leicht verminderten absoluten Werten. In dieser Konfiguration wird der Bereich der optimalen Liftauslastung erst bei 60 Shuttles erreicht, das Minimum der Doppelspielzeit hat sich auf 35 Shuttles

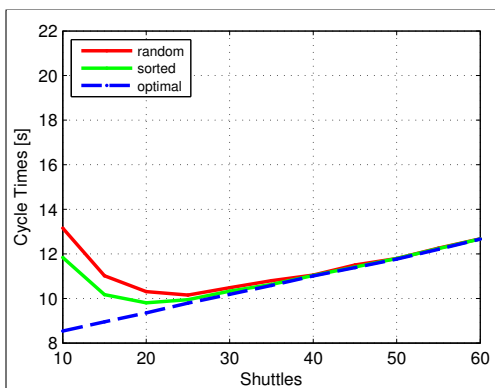


Abbildung 6.35: Vergleich der Koordinatenerzeugung –
 $l = 40 \text{ m}; n_{aS} = 10$

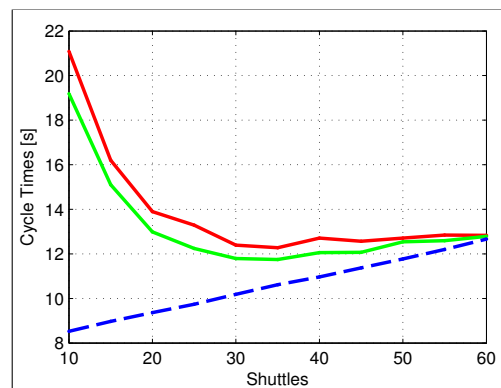


Abbildung 6.36: Vergleich der Koordinatenerzeugung –
 $l = 80 \text{ m}; n_{aS} = 10$

erhöht, bei weiterer Erhöhung der Shuttles bleiben die erreichten Zeiten aber annähernd konstant.

Als nächstes soll wieder die Auswirkung der Position der Übergabestelle auf die mittlere Doppelspielzeit erörtert werden. In Abbildung 6.37 werden die beiden Varianten: Position in Ebene 0 oder Position in Ebene $\frac{n_S}{2}$ dargestellt, bei einem System der Länge von 40 m und mit 10 „aktiven Shuttles“. Durch die veränderte Position der Übergabestelle wird direkt die Fahrzeit des Liftes reduziert, was wiederum eine Senkung der optimalen Zeit zur Folge hat, wie leicht zu erkennen ist. Die dementsprechenden Doppelspielzeiten nähern sich nun den optimalen Zeiten an, am Anfang jedoch, wo noch lange Wartezeiten auftreten, lässt sich kein Unterschied zwischen den Zeiten feststellen.

Verändert man nun den Parameter Länge auf 80 m, wie in Abbildung 6.38 zu sehen ist, so zeigt sich, dass ein Unterschied erst ab einer Anzahl von 30 Shuttles entsteht. Das Minimum der Doppelspielzeit verschiebt sich jedoch von 30 Shuttles, wenn sich die Übergabestelle in Ebene 0 befindet, zu 55 Shuttles, wenn sie sich in Ebene 28 befindet.

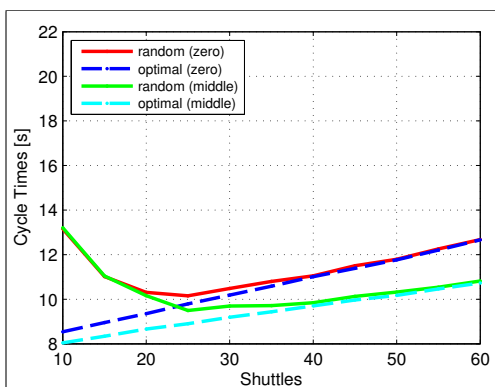


Abbildung 6.37: Vergleich der Übergabestelle –
 $l = 40 \text{ m}; n_{aS} = 10$

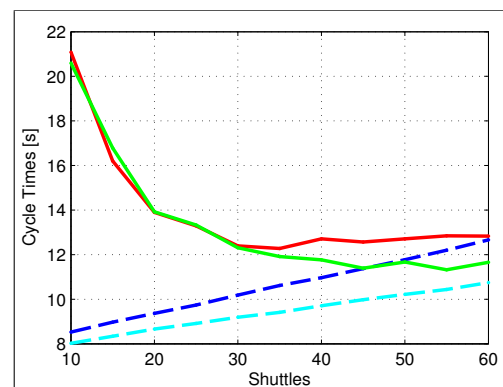


Abbildung 6.38: Vergleich der Übergabestelle –
 $l = 80 \text{ m}; n_{aS} = 10$

Betrachtet man nun den Einfluss der „aktiven Shuttles“ auf die Entwicklung der Doppelspielzeiten, so lässt sich diese aus Abbildung 6.39 für Systeme mit einer Länge von 40 m ablesen. Nachdem eine anfängliche Erhöhung der „aktiven Shuttles“ von 5 auf 10 die Doppelspielzeiten deutlich senken kann, bewirkt eine weitere Änderung dieses Parameters keine weitere Verbesserung des Verlaufes. Zu beachten ist hierbei aber, dass in diesem Datensatz nur Systeme mit $n_{aS} \leq 1,5 \cdot n_S$ enthalten sind, sodass Graphen mit einer hohen Anzahl an „aktiven Shuttles“ erst ab einer gewissen Systemhöhe beginnen. Wie sich ein derartiges System verhält, wird in Unterabschnitt 6.4.3 anhand eines anderen Datensatzes erläutert.

Den Einfluss dieses Parameters auf ein größeres System – mit einer Länge von 80 m – stellt Abbildung 6.40 dar. Aus dieser Grafik lässt sich ableiten, dass eine geringe Anzahl an „aktiven Shuttles“ die Spielzeiten dauerhaft stark ansteigen lassen, bei 5 „aktiven Shuttles“ bleibt die mittlere Doppelspielzeit ab einer Höhe von 30 Ebenen annähernd konstant bei 16 s. Hier kann durch eine Erhöhung der „aktiven Shuttles“ auf 20 noch eine weitere Verbesserung der Doppelspielzeiten erreicht werden.

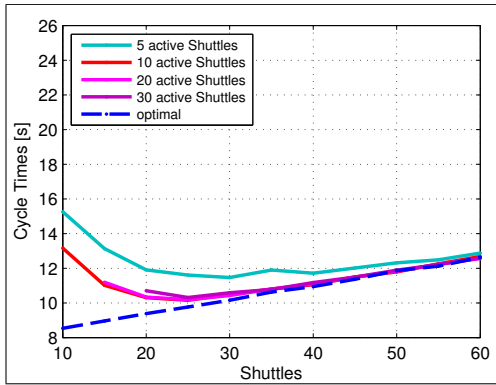


Abbildung 6.39: Vergleich der „aktiven Shuttles“
– $l = 40$ m

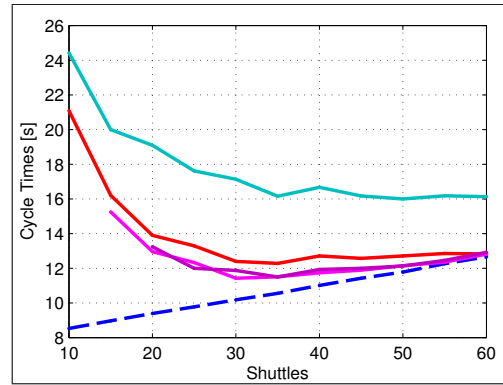


Abbildung 6.40: Vergleich der „aktiven Shuttles“
– $l = 80$ m

In den Abbildungen 6.41 und 6.42 werden nun die Auswirkungen der Länge auf die Doppelspielzeit bei einer bestimmten Anzahl an „aktiven Shuttles“ (10 in Abb. 6.41 bzw. 20 in Abb. 6.42) gezeigt. Die Doppelspielzeiten sind in beiden Fällen im Bereich niedriger Anzahlen an Shuttles stark erhöht, wobei höhere Längen diesen Effekt noch verstärken. Nach den hohen Maximalwerten im vorderen Bereich sinken die Werte immer weiter ab, bis sie sich schließlich den optimalen Zeiten nähern. Es lässt sich also für jede Längen-Linie ein Bereich finden, in dem die Doppelspielzeit ein Minimum aufweist. Die Erhöhung der „aktiven Shuttles“ in Abbildung 6.42 zeigt dieselben Tendenzen, die Maximalwerte liegen aber deutlich niedriger und auch die Annäherung an die optimale Zeit beginnt bei einer geringeren Anzahl an Shuttles.

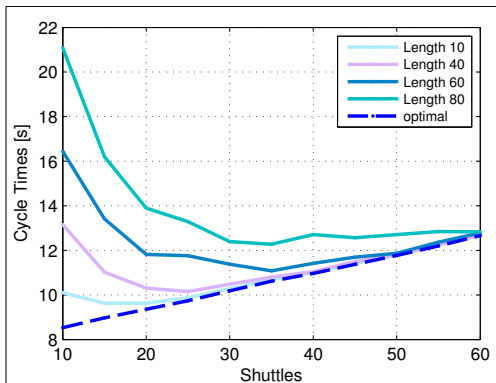


Abbildung 6.41: Einfluss der Länge – $n_{aS} = 10$

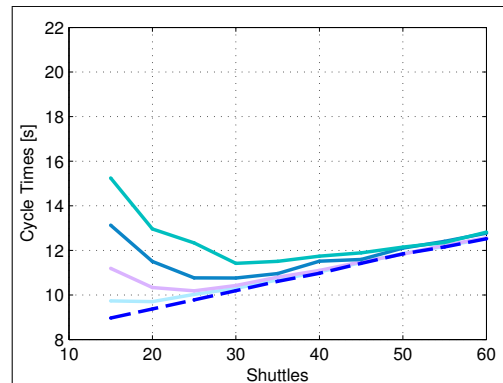


Abbildung 6.42: Einfluss der Länge – $n_{aS} = 20$

6.4.3 Variation der „aktiven Shuttles“

In den vorangegangenen Kapiteln wurde bereits auf die Auswirkung des Parameters „aktive Shuttles“ im Kontext der jeweiligen Variationen eingegangen. In diesem Abschnitt werden die Erkenntnisse nochmals vertieft, indem die Zeiten bei direkter Veränderung der „aktiven Shuttles“ für verschiedene Systeme dargestellt werden. Zuerst wird der Einfluss

der Koordinatenerzeugung auf die Entwicklung der Doppelspielzeiten besprochen. Dafür werden zwei Systeme verglichen, beide haben eine Höhe von 20 Ebenen, unterscheiden sich aber in der Länge: Während das eine System nur 30 m lang ist, beträgt die Länge beim zweiten 80 m. In Abbildung 6.43 wird ersteres System dargestellt, hier lässt sich wieder beobachten, dass die maximalen Spielzeiten bei einem kurzen System eher niedrig ausfallen und nach einer anfänglichen Verbesserung durch Erhöhen der „aktiven Shuttles“ schon bald kein Einfluss mehr festgestellt werden kann. Der Graph, der die Werte ohne direkte Wiederholung beschreibt, liegt tendenziell unter der bei einer zufälligen Verteilung, sogar in dem Wertebereich, in dem der Graph annähernd konstant an der optimalen Zeit liegt. Abbildung 6.44 zeigt nun das zweite beschriebene Setup, ein System das wesentlich länger als hoch ist. Hier zeigt sich, dass bei einer geringen Anzahl an „aktiven Shuttles“ sehr hohe Doppelspielzeiten auftreten, die bei einer Erhöhung dann deutlich abnimmt. Die Differenz der beiden Graphen ist am Anfang relativ gering und prägt sich erst ab ungefähr 5 „aktiven Shuttles“ deutlich aus. Ab einer Anzahl von 15 „aktiven Shuttles“ wirkt sich eine Erhöhung nicht mehr positiv auf die Doppelspielzeit aus, obwohl die optimale Zeit noch nicht erreicht wurde. Daraus lässt sich schließen, dass dieser Parameter nur bis zu einem bestimmten Wert einen positiven Einfluss auf die Zusammensetzung der Zeiten hat. Ein gewisser Anteil der Wartezeit lässt sich nicht vermeiden, auch wenn die Anzahl der vorbereiteten Aufträge stark gesteigert wird. Die Koordinatenerzeugung wirkt sich erst aus, sobald eine gewisse Anzahl an „aktiven Shuttles“ erreicht ist, davor treten bei jedem Zyklus Wartezeiten auf, sodass die Vermeidung von direkten Wiederholungen sich nicht auf die mittleren Doppelspielzeiten auswirkt.

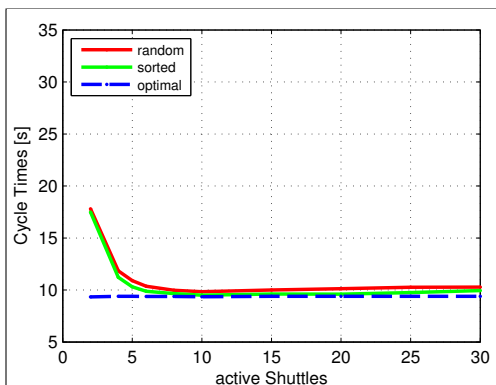


Abbildung 6.43: Vergleich der Koordinatenerzeugung – $l = 30 \text{ m}; n_S = 20$

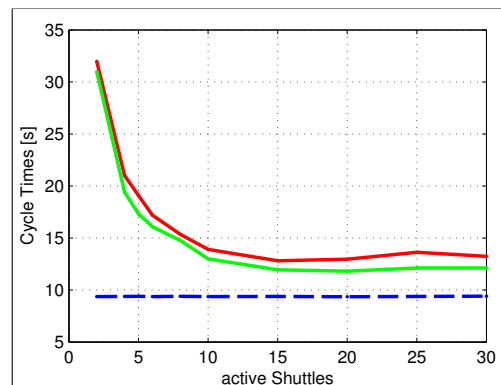


Abbildung 6.44: Vergleich der Koordinatenerzeugung – $l = 80 \text{ m}; n_S = 20$

Die Abbildungen 6.45 und 6.46 zeigen nun die Verläufe von Längen zwischen 20 m und 80 m bei einer Höhe von 20 bzw. 40 Ebenen. Hier zeigen sich in Abbildung 6.45 die zuvor beschriebenen Tendenzen: bei größeren Längen steigen einerseits die maximalen Doppelspielzeiten im Bereich niedriger „aktiver Shuttles“, andererseits nähern sich die Zeiten einem linearen Verlauf an. Der Punkt, an dem dieser Bereich erreicht wird, verschiebt sich zu immer höheren Werten. Die geringe Anzahl an Ebenen resultiert in einem konstanten Anteil an Wartezeit, der auch durch die „aktiven Shuttles“ nicht ausgeglichen werden kann. Betrachtet man nun ein System mit 40 Ebenen, wie in Abbildung 6.46 dargestellt, so

sieht man, dass sich die zuvor beschriebenen Trends wiederholen. Der Unterschied liegt aber im Bereich der hohen Werte der „aktiven Shuttles“: Hier zeigt sich, dass die optimalen Zeiten bei allen Längen erreicht werden. Durch die größere Höhe liegt diese aber ungefähr 2s über der im zuvor beschriebenen System. Durch die größere Anzahl an Ebenen ist die Wahrscheinlichkeit eines Auftretens von Wartezeiten geringer und kann durch eine Erhöhung des Parameters gänzlich eliminiert werden.

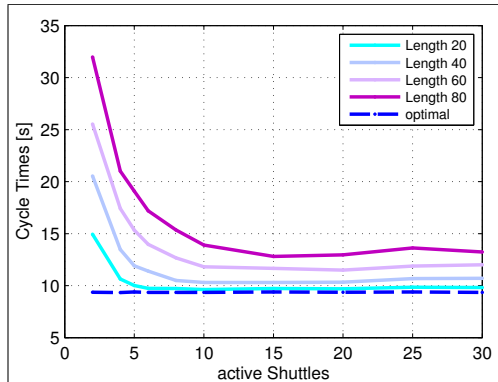


Abbildung 6.45: Einfluss der Länge bei hohen „aktiven Shuttles“ – $n_S = 20$

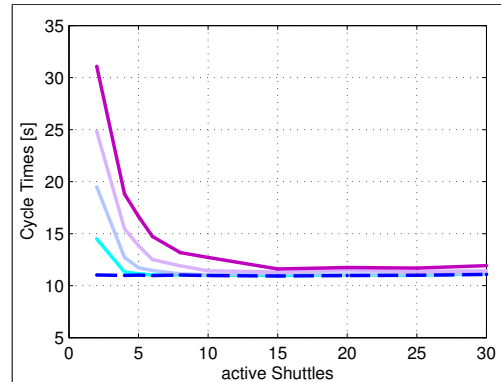


Abbildung 6.46: Einfluss der Länge bei hohen „aktiven Shuttles“ – $n_S = 40$

Die zuvor beschriebenen Tendenzen legen den Rückschluss nahe, dass eine Erhöhung der „aktiven Shuttles“ zwar nicht immer eine Verbesserung der Doppelspielzeiten mit sich bringt, aber sich zumindest nicht negativ auswirkt. Um dieses Verhalten zu untersuchen, wurde eine zweite Satz an Simulationen durchgeführt, bei der für jede Ebenen-Anzahl Werte für die „aktiven Shuttles“ zwischen 10 und 70, bei einer Schrittweite von 10 und einem Startwert von 5 berechnet wurde. Dies steht im Gegensatz zu den Daten, die für die anderen Grafiken herangezogen werden, da dort die maximale Anzahl der simulierten „aktiven Shuttles“ von der Anzahl der Ebenen abhängt, wie in Abschnitt 6.2 beschrieben. Abbildung 6.47 zeigt wieder den Einfluss der Länge, diesmal aber bis zu Werten von 70 „aktiven Shuttles“. Bis 30 sieht der Verlauf annähernd gleich aus wie in Abbildung 6.45, erhöht man aber die Anzahl der „aktiven Shuttles“ noch weiter, so zeigt sich, dass die mittleren Doppelspielzeiten wieder ansteigen. Dieser Effekt entsteht dadurch, dass das System in allen Ebenen Auslagerungen vorbereitet und dadurch die Einlagerungen nicht mehr vorbereitet werden können. Das führt zu erhöhten Wartezeiten im Einlagerungszyklus, die sich direkt auf die mittlere Doppelspielzeit auswirken. Betrachtet man nun Abbildung 6.48, so zeigt sich, dass die Zeiten auch bei hohen „aktiven Shuttles“ annähernd konstant bleiben, nur bei dem Graphen, der eine Länge von 80 m repräsentiert, kann am Ende eine leicht steigende Tendenz festgestellt werden. Der Beginn des steigenden Bereichs wurde also weiter nach rechts verschoben, die Auswirkungen beginnen also erst bei einer höheren Auslastung.

Nun soll abschließend noch gezeigt werden, wie sich die Anzahl der Shuttles auf die Spielzeiten auswirkt. Dazu werden in den Abbildungen 6.49 und 6.50 bei einer Länge von 40 m bzw. 80 m für verschiedene Anzahlen an Ebenen die mittleren Doppelspielzeiten und die zugehörigen optimalen Zeiten dargestellt.

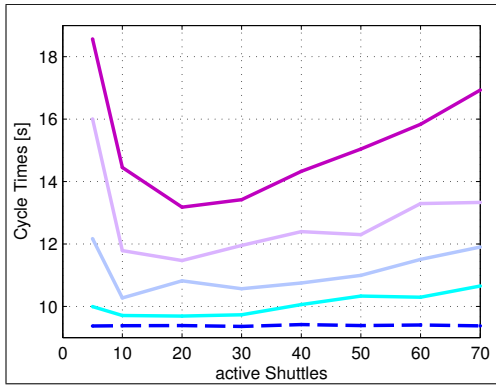


Abbildung 6.47: Einfluss der Länge – $n_s = 20$

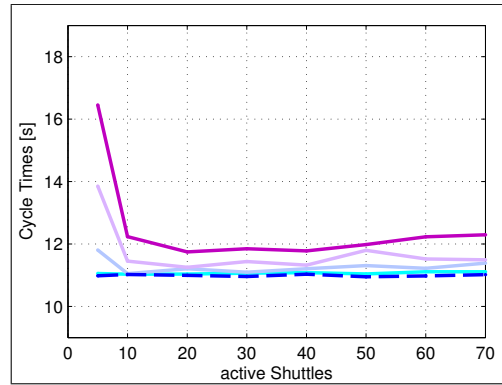


Abbildung 6.48: Einfluss der Länge – $n_s = 40$

Bei den in Abbildung 6.49 dargestellten Systemen mit geringerer Länge zeigt sich wieder der bekannte Effekt: eine geringe Anzahl von Ebenen führt zu höheren Spielzeiten und einer Wartezeit, die auch bei erhöhen der „aktiven Shuttles“ nicht verschwindet. Erhöht sich nun die Anzahl der Shuttles, so steigen einerseits die optimalen Zeiten, andererseits nähern sich die Doppelspielzeiten aber den optimalen Zeiten an, sodass der konstante Anteil an Wartezeiten entfällt. Dies führt dazu, dass die Doppelspielzeit bei 30 Ebenen unter der von 50 Ebenen liegt. Betrachtet man nun die Systeme mit größerer Länge in Abbildung 6.50, so erkennt man, dass die mittlere Doppelspielzeit bei einer geringen Höhe stark erhöht ist. Die Anzahl der Shuttles (30 und 50) hingegen bewirkt keinen signifikanten Unterschied mehr, da durch den Anstieg der optimalen Zeit lediglich die auftretenden Wartezeiten kompensiert werden.

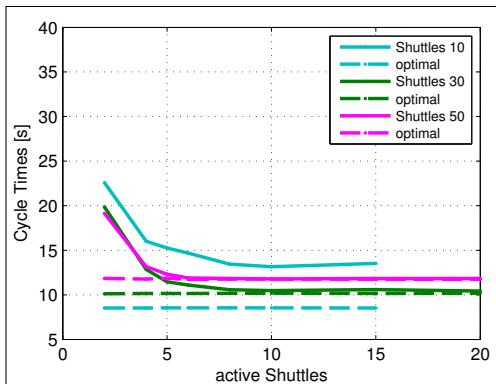


Abbildung 6.49: Einfluss der Shuttles – $l = 40$ m

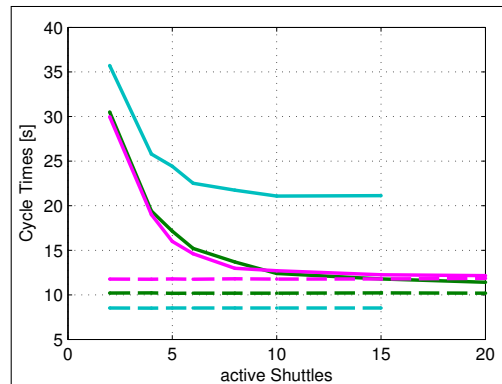


Abbildung 6.50: Einfluss der Shuttles – $l = 80$ m

Diese Folgerungen legen nahe, dass es für jede Kombination aus Ebenen, Länge und „aktiven Shuttles“ einen optimalen Betriebspunkt geben muss, bei dem die niedrigste Doppelspielzeit und damit die höchste Umschlagleistung erreicht wird.

6.5 Interpretation der Ergebnisse

In diesem Abschnitt sollen nun die Auswirkungen aller bisher besprochenen Effekte zusammengefasst und optisch vergleichbar gemacht werden. Nachdem in Abschnitt 6.3 die einzelnen Zyklen detailliert besprochen und in Abschnitt 6.4 die Variationen einzelner Parameter mit konstanten Rahmenbedingungen analysiert wurden, sollen in der Folge mehr-achsige dreidimensionale Grafiken eine Übersicht über einen sehr großen Wertebereich geben. Es wird jeweils nur der Parameter der „aktiven Shuttles“ konstant gehalten.

6.5.1 Entwicklung der Umschlagleistung

Als wichtiger Wert für die Leistungsfähigkeit eines fördertechnischen Systems wird die Umschlagleistung herangezogen. In der Folge wird nun in einer Reihe von Grafiken der Zusammenhang der Umschlagleistung mit den anderen Parametern dargestellt.

In Abbildung 6.51 werden die Daten, die auch in den vorigen Kapiteln herangezogen wurden, als dreidimensionale Grafik dargestellt. Die Parameter ‚Länge‘ und ‚Shuttles‘ werden vollständig aufgetragen, die Koordinatenerzeugung ist zufällig, die Übergabestelle befindet sich in Ebene 0 und für die „aktiven Shuttles“ wurde ein Wert von 10 festgelegt. Zur Übersicht wird in dieser Abbildung auch die Farbskala angezeigt, die für alle weiteren Grafiken der Umschlagleistung ebenfalls angewendet wird.

Aus dieser Grafik ist ersichtlich, dass die maximale Umschlagleistung bei sehr kleinen Systemen erreicht wird. Bei einer geringen Anzahl an Ebenen wirkt sich eine höhere Länge sehr negativ auf die Umschlagleistung aus, diese sinkt dabei um bis zu 50 % ab. Erhöht man die Anzahl der Ebenen auf einen Wert zwischen 20 und 40 so sinkt die Umschlagleistung zwar ab, jedoch liegt sie in einem Bereich zwischen 10 m und 60 m konstant über $320 \frac{1}{h}$. In den Randbereichen sinken die Umschlagleistungen deutlich ab.

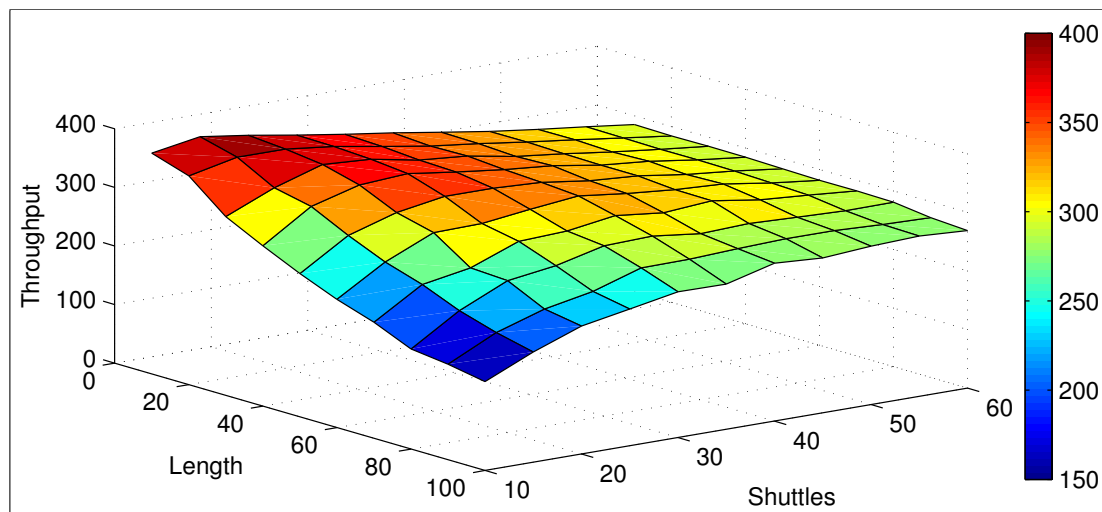


Abbildung 6.51: Umschlagleistung – $n_{as} = 10$

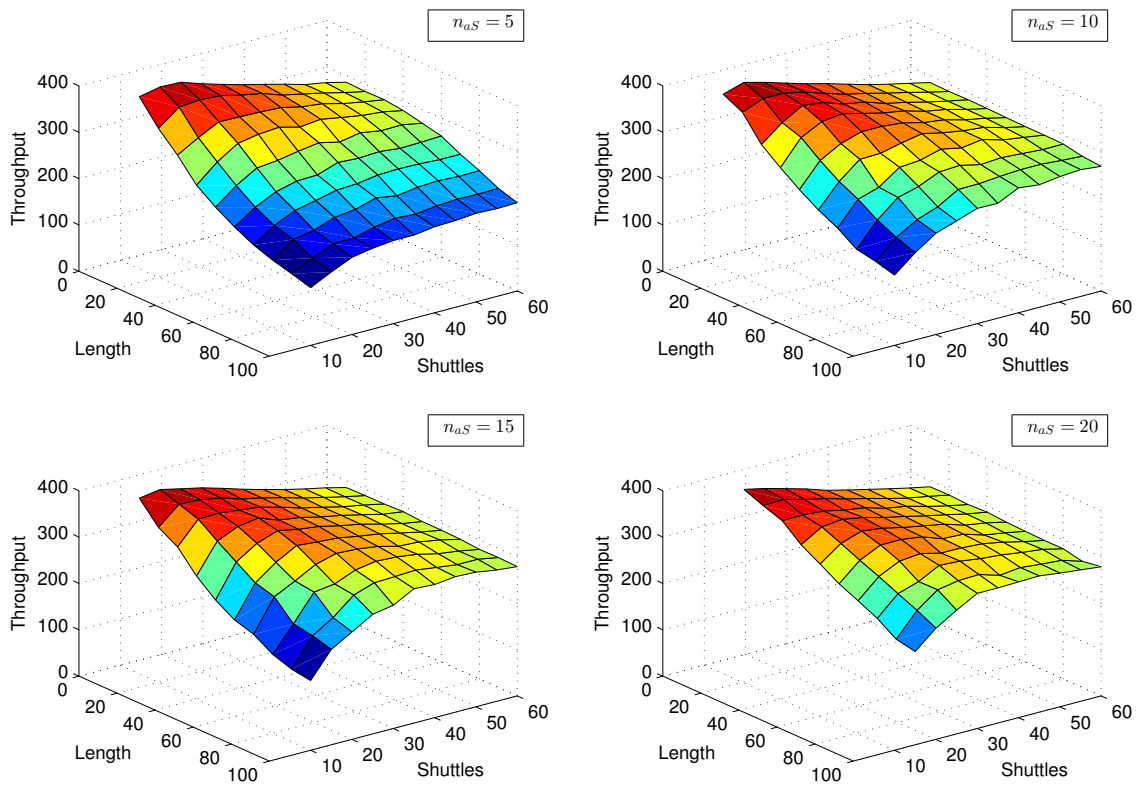


Abbildung 6.52: Umschlagleistung in Abhängigkeit der Länge und der Shuttles

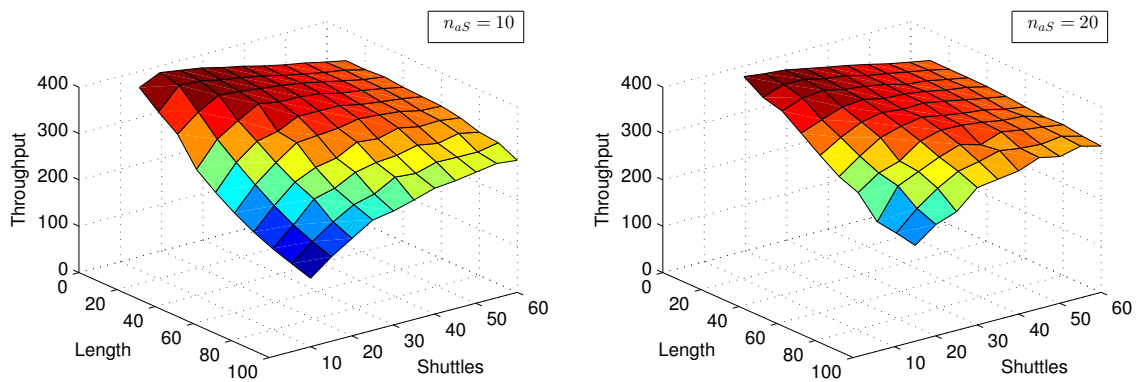


Abbildung 6.53: Umschlagleistung in Abhängigkeit der Länge und der Shuttles – Position der Übergabestelle in der Mitte des Liftes

In Abbildung 6.52 lässt sich die Auswirkung des Parameters „aktive Shuttles“ auf die Umschlagleistung vergleichen. Die Erhöhung von 5 auf 10 bewirkt eine massive Verbesserung der Leistung bei Systemen mit größerer Länge. Eine weitere Erhöhung wirkt sich nicht mehr so stark aus, kann aber immer noch leichte Optimierungseffekte erzielen. In der Realität entspricht die Erhöhung der „aktiven Shuttles“ der Anzahl der Aufträge, die vom System vorbereitet werden. Diese Anzahl variiert natürlich abhängig von der vorherrschenden Auftragslage des Systems und wird deshalb nicht während der gesamten Einsatzzeit konstant bleiben. Es ist zu erwarten, dass die Anzahl der Aufträge zu Stoßzeiten höher liegt als zu schwächeren Zeiten. Der hier ausgeführte Effekt würde also dazu führen, dass das System zu Stoßzeiten effizienter arbeitet als bei geringer Auslastung, da durch den höheren Gleichzeitigkeits-Faktor mehr Aufträge für die Auslagerung vorbereitet werden können.

Es lässt sich auch wieder feststellen, dass die Kombination aus einer großen Länge und einer geringen Anzahl der Ebenen zu äußerst niedrigen Umschlagleistungen führt und daher solche Systeme vermieden werden sollten, wenn eine hohe Umschlagleistung gefordert ist.

Vergleicht man nun die Graphen aus Abbildung 6.52 mit denen aus Abbildung 6.53, in der die Umschlagleistungen für Systeme, deren Übergabestelle sich in der mittleren vertikalen Position befinden, so lassen sich wieder die schon zuvor festgestellten Trends konstatieren: Während bei geringen Systemhöhen keine Verbesserungen zu erkennen sind, so wird deutlich, dass die Umschlagleistungen bei Systemen mit mehr Ebenen merklich höher sind als bei den Vergleichssystemen. Es ist also ratsam, bei höheren Systemen die Übergabestelle nicht am unteren Ende des Regals zu positionieren, sondern diese in eine höhere Ebene zu verlegen. Die Fördertechnik muss dann dafür sorgen, dass die Lagereinheiten aus dem Shuttlesystem abtransportiert werden bzw. in dieses gelangen können. Führt man diesen Gedanken weiter, wäre es bei hohen Systemen auch denkbar, mehrere Übergabepositionen einzurichten, um diese dann im Sinne des ‚Nearest Neighbour‘-Prinzips anzufahren.

6.5.2 Zusammenhang der Umschlagleistung und der Lagerplätze

Bis jetzt wurden nur die geometrischen Parameter ‚Länge‘ und ‚Höhe‘ und deren Auswirkung auf die Umschlagleistung besprochen. Nun ist es aber durchaus interessant zu vergleichen, wie sich die Umschlagleistungen bezogen zur Anzahl an vorhandenen Lagerplätzen im System verhalten, denn eine gewissen Anzahl an Lagerplätzen kann durch unterschiedliche Kombinationen aus Längen und Anzahl an Ebenen erreicht werden.

In Abbildung 6.54 sind für verschiedene Anzahlen der „aktiven Shuttles“ die Abhängigkeit der Umschlagleistung im Verhältnis zur Anzahl der Ebenen dargestellt. Die Anzahl der Lagerpositionen wird in Schritten von 1000 aufgetragen. Man sieht wieder, dass die durchschnittliche Umschlagleistung bei mehr „aktiven Shuttles“ ansteigt. Die Abbildungen stellen eine leicht in Richtung der höheren Anzahl der Shuttles abfallende Ebene dar, die bei einer gewissen Anzahl an Lagerpositionen stark nach unten weg knickt. An dieser Kante kann abgeschätzt werden, welche maximale Länge dieses System besitzen kann, ohne dass sich diese negativ auf die Umschlagleistung auswirkt. Der Anteil des ebenen Verlaufs erhöht sich immer weiter, je höher die Anzahl der „aktiven Shuttles“ wird. Wird z. B. bei $n_{aS} = 5$ noch eine Höhe von 30 Shuttles benötigt, um bei einer Anzahl von 5000

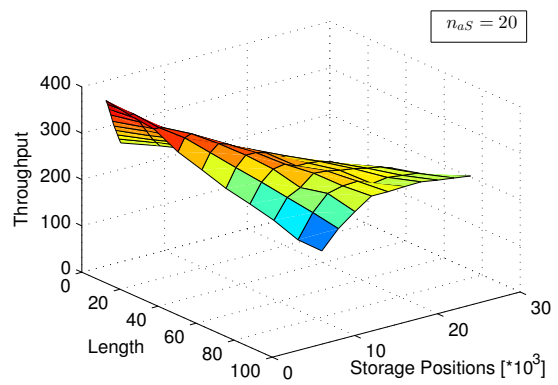
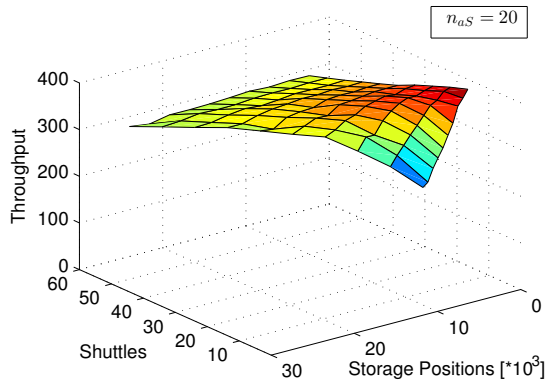
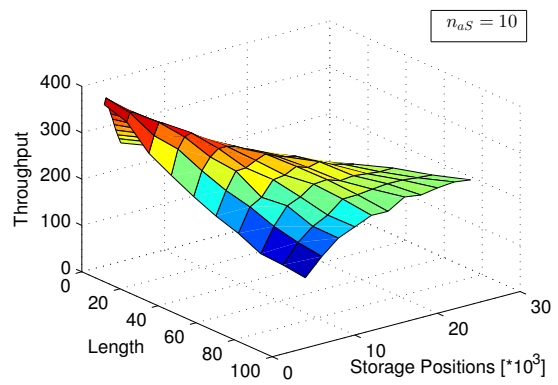
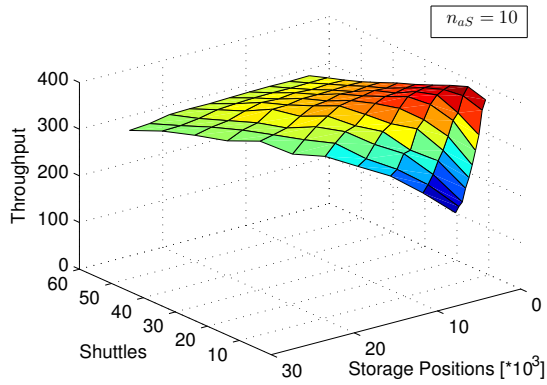
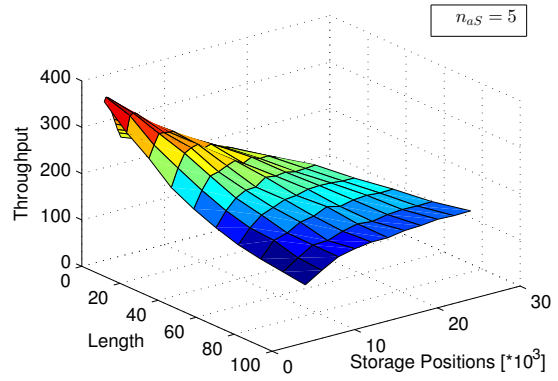
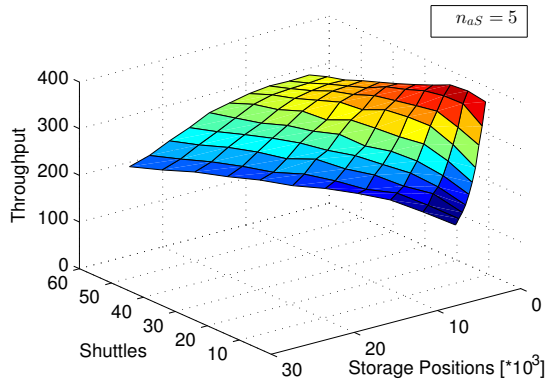


Abbildung 6.54: Umschlagleistungen in Abhängigkeit der Shuttles und der Lagerpositionen

Abbildung 6.55: Umschlagleistungen in Abhängigkeit der Länge und der Lagerpositionen

Lagerpositionen eine Umschlagleistung von über $310 \frac{1}{h}$ zu erreichen, so verringert sich diese auf zwischen 20 und 25 Ebenen bei 10 „aktiven Shuttles“ und auf 20 Ebenen bei 20 „aktiven Shuttles“. Dies entspricht einer Erhöhung der Länge von 40 m auf 60 m, die sich trotz Reduzierung der Ebenen nicht auf die Umschlagleistung auswirkt. Da die Anzahl der Ebenen und damit der eingebauten Shuttles ein Kosten-entscheidender Faktor bei der Auslegung eines Shuttle-Systems ist, ist diese Information sehr wichtig für die Auslegung eines Systems.

Daneben ist in Abbildung 6.55 der Zusammenhang der Länge mit den Lagerplätzen dargestellt. Hier sind dieselben allgemeinen Überlegungen zulässig wie zuvor besprochen. Auffällig hierbei ist, dass diese Fläche, im Gegensatz zur zuvor besprochenen Darstellung, die Form einer umgekehrten Rinne aufweist. Aus diesem Verhalten lässt sich also für jede Länge – über die Anzahl der Lagerplätze – die optimale Anzahl an Ebenen ermitteln.

Ein Beispiel: Bei einer Anzahl von 5 „aktiven Shuttles“ und einer Länge von 60 m kann die Anzahl der Ebenen in einem Bereich zwischen 30 und 55 liegen, wobei in allen Fällen Umschlagleistungen von ungefähr $260 \frac{1}{h}$ erreicht werden. Dieses Verhalten resultiert daraus, dass die Kurve in diesem Bereich schon sehr flach ist, sich das System also nicht in einem optimalen Bereich befindet. Betrachtet man nun dieselbe Ausgangssituation bei 10 „aktiven Shuttles“, so lässt sich ein tatsächliches Maximum bei einer Anzahl von 35 Ebenen und einer Umschlagleistung von $325 \frac{1}{h}$ bestimmen. Erhöht man die „aktiven Shuttles“ weiter auf 10, so bleibt die Umschlagleistung für 35 Ebenen annähernd konstant. Das Maximum verschiebt sich aber auf 30 Ebenen bei einer leicht gestiegenen Umschlagleistung von $335 \frac{1}{h}$. Analog zu den ausgeführten Beispielen lässt sich nun für jedes beliebige System eine Untersuchung durchführen und mit der Auswertung relativ weniger verschiedener Graphen ein anschauliches und vergleichbares Ergebnis produzieren.

6.5.3 Kritische Annahmen und Optimierungspotential der Simulation

Nachdem nun die Ergebnisse der verschiedenen Simulationen ausführlich diskutiert wurden, soll nochmals die Entstehung der Daten kritisch hinterfragt werden.

• Aktive Shuttles

Es zeigt sich über alle Betrachtungen hinweg, dass der Parameter „aktive Shuttles“, der in dieser Simulation eingeführt wurde, einen großen Einfluss auf die Ergebnisse hat. Dieser Parameter existiert aber in dieser Form nicht in einem realen System, da hier die Auslastung von der aktuellen Auftragslage abhängt und mit hoher Wahrscheinlichkeit nicht über einen langen Zeitraum konstant bleibt. Die Simulation kann deshalb nur Tendenzen und Abhängigkeiten zeigen, jedoch kein reales System mit realen Auslastungen eins zu eins abbilden.

Optimierung:

Um diesen Effekt abzubilden wäre die Implementation einer Verteilungsfunktion notwendig, die während einer Simulation die Anzahl der gerade aktiven Shuttles variiert. Um bessere Ergebnisse zu erhalten, wären aber Daten für das Lastkollektiv eines realen Systems notwendig.

• Koordinatenerzeugung

Ein weiterer entscheidender Faktor, der sich direkt auf die Ergebnisse auswirkt, besteht in der Annahme, dass alle Koordinaten zufällig generiert werden und keine Optimierung der Reihenfolge vorgenommen wird. Diese Annahme rechtfertigt sich vor allem bei der Auslagerung, da hier auch in der Realität die Sequenzierung eine entscheidende Rolle spielt und das Potential für geringfügige Modifikationen eher niedrig ist. Bei einer Einlagerung hingegen ist das Einsparungspotential größer, da – nach dem Prinzip des ‚chaotischen‘ Lagers – eine Einlagerung an jeder freien Lagerposition denkbar ist. In einem realen System bedeutet das, dass Wartezeiten im Einlagerungszyklus nur sehr selten auftreten, da der Lift nicht wartet bis der Puffer geleert wird, sondern gleich in die nächste Ebene mit leerem Puffer fährt. Ebenso wird wahrscheinlich in einer Phase hoher Auslastung eine Lagerposition in den vorderen Bereichen des Ganges bevorzugt vergeben und hintere Positionen erst in Phasen geringer Auslastung aufgefüllt.

Optimierung:

Um die Doppelspielzeiten in den Bereichen, in denen schon konstante Wartezeiten auftreten, weiter zu optimieren, wäre es möglich, die oben beschriebene Vorgangsweise zu implementieren. Da Wartezeiten im Einlagerungszyklus seltener auftreten und auch weniger hoch sind als diejenigen im Zwischenzyklus, sollte der Einfluss auf die mittleren Doppelspielzeiten aber eher gering ausfallen.

• Priorisierung

Um die Simulation in einem großen Wertebereich stabil zu halten, hat in der Funktion $[Define\ Next\ Step]_f$ das Leeren des Einlagerungspuffers eine höhere Priorität als die nächste Auslagerung, da das Auslagern bei gefülltem Puffer möglicherweise zu einer Deadlock-Situation führen kann, die damit endet, dass die Einlagerung überhaupt nicht ausgeführt werden kann. Diese Annahme wirkt auf den ersten Blick unzulässig, da ja die Auslagerung und deren Sequenzierung die nach außen wirksamen Systemparameter sind, wohingegen die Einlagerung nur im Hintergrund geschieht. Betrachtet man aber nun das System genauer, so stellt man fest, dass viele Auslagerungsvorgänge gleichzeitig stattfinden, aber nur eine oder relativ wenige Einlagerungen. Dies hat zur Folge, dass die Wahrscheinlichkeit sehr hoch ist, dass die Einlagerung nach dem Beenden einer Auslagerung ausgelöst wird. Zu diesem Zeitpunkt befindet sich das Shuttle schon an der Übergabestelle und muss sich für die nächste Auslagerung wieder in den Gang hinein bewegen. Nun wird das Shuttle mit der einzulagernden Einheit beladen und bewegt sich zum Lagerort für die Einlagerung. Dabei wird einerseits die Totzeit des Shuttles, also die Zeit in der es leer zur Position der nächsten Auslagerung gefahren wäre, genutzt, andererseits aber die Zeit für die nächste Auslagerung um die Übergabezeiten und die Fahrzeit zwischen den beiden Positionen verlängert. Eine ausschließliche Verschlechterung tritt nur im seltenen Fall ein, dass zwei Einlagerungen hintereinander ausgeführt werden, da zusätzliche Totzeit auftritt.

Optimierung:

Um das Auftreten von Deadlock Situationen zu vermeiden und gleichzeitig die Sequenzierung zu priorisieren, wäre die Implementation komplexer Entscheidungs-Algorithmen notwendig: So müsste im Zeitpunkt der Auslösung von $[Define\ Next\ Step]_f$ festgestellt werden, ob die nächste Auslagerung in einer der nächsten Zyklen benötigt wird oder ob Zeit bleibt, die Einlagerung sofort durchzuführen. Die Simulation wurde nicht im Hinblick

auf solche Entscheidungen entworfen, daher ist es auch im Nachhinein – bei Beibehaltung der anderen Funktionen – fast unmöglich, solche zu implementieren, ohne das gesamte Simulationskonzept zu ändern.

• Zeitpunkte der Überprüfung

Es wurde schon in vorherigen Kapiteln (vgl. Unterabschnitt 5.2.4 und Unterabschnitt 5.3.3) darauf hingewiesen, dass bei Ausführung der Funktion $[Define\ Next\ Step]_f$ die aktuelle Simulationszeit – und nicht die Endzeit des letzten Prozesses – als Ausgangszeitpunkt für neu startende Prozesse festgelegt ist. Dadurch dauern Hintergrundprozesse länger, als sie es eigentlich – aufgrund der Logik – dürften. Dieser Effekt kann sich auf die mittleren Doppelspielzeiten auswirken und tritt vor allem bei einer niedrigen Anzahl von „aktiven Shuttles“ und großen Wartezeiten auf.

Optimierung:

Um diese Fehlerquelle zu beseitigen, sind mehrere Möglichkeiten denkbar. Eine Lösung wäre es, für jeden Fall, der eintreten kann, eine eigene Routine zu implementieren. Diese Vorgehensweise würde aber zu einer sehr großen Zahl an Bedingungen führen und damit die Nachvollziehbarkeit und eine spätere Modifikation erschweren. Ein einfacherer Ansatz wäre es, eine geringe Unschärfe zuzulassen und während der Überprüfung nicht nur den Anfangs- und Endzeitpunkt zu betrachten, sondern ab einem gewissen Intervall in inkrementellen Schritten mehrere Überprüfungen durchzuführen. Dadurch würde das Gerüst der Simulation nicht verändert werden, aber die Abweichungen von den tatsächlichen Zeiten würden sich nur mehr im Größenbereich der Schrittweite bewegen.

• Der Doppelspielzyklus

Die größte Annahme, die getroffen wurde, bezieht sich auf die Funktionsabfolge an sich. Da den Doppelspielzeiten besondere Bedeutung beigemessen werden soll, wurde das System so erstellt, dass es ausschließlich in der Lage ist Doppelspielzyklen zu simulieren. Daher wurde für den Aufbau des Systems die Variante mit einem Lift, der hintereinander Ein- und Auslagerungen durchführt, ausgewählt. Tatsächlich aber arbeiten die Systeme meist mit zwei Liften, sodass keine echten Doppelspielzyklen mehr zustande kommen, sondern nur eine Abfolge von Einzelzyklen. Diese sind prinzipiell voneinander unabhängig und werden nur auf Grund der Auslastung der Shuttles miteinander verbunden. Es wäre denkbar, die vorliegende Simulation durch Erweiterung der Funktionen und Entitäten auch im Hinblick auf eine solche Ausgangslage auszurichten.

6.6 Ergebnisse verschiedener Simulationen im Vergleich

Zeitgleich mit dieser Diplomarbeit entstand an der Technischen Universität Wien die Master-Arbeit von CASTELLS I SANABRA [Cas13], die sich ebenfalls mit dem Thema Shuttle-Regal-Systeme beschäftigt. Durch die Abstimmung der Parameter, die in beiden Simulationen Anwendung gefunden haben, konnte eine Vergleichbarkeit der Ergebnisse erreicht werden.

Vergleich der Simulationen

Die beiden Arbeiten unterscheiden sich durch die jeweilige Funktionsweise der Simulation. Castells greift auf das Softwarepaket ARENA zurück, welches die eigene Programmiersprache SIMAN verwendet, um diskrete Prozesse darzustellen. Die Software stützt sich dabei auf Flussdiagramme, die die logischen Abfolgen wiedergeben. Hierbei wird auf vorgefertigte Gruppen zurückgegriffen, welche ihrerseits angepasst werden können. Dieses Vorgehen ermöglicht es, sich auf die Modellierung der Prozesse zu konzentrieren, während das Grundgerüst der Simulation, also insbesondere die Zeitpunktabfrage, automatisch vom systemeigenen Prozesssimulator erledigt wird. Bei Verwendung von ARENA lassen sich die Bewegungen der einzelnen Systemkomponenten grafisch darstellen und somit auf dem Bildschirm verfolgen [Cas13, S. 26-28].

In der vorliegenden Arbeit wurde dem Erstellen eines eigenständigen Grundgerüsts der Simulation der Vorzug gegeben. Jeder Schritt ist logisch nachvollziehbar, von mir implementiert und dokumentiert, sodass das Einarbeiten in die wichtigsten Funktionen relativ wenig Zeit in Anspruch nehmen sollte, vor allem weil keine zusätzliche Kenntnis einer Simulationssprache notwendig ist. Die ARENA-Simulation, in ihrer derzeitigen Form, hat Schwächen bei der Darstellung von größeren Systemen, braucht also sehr lange, um eine einzige Systemkonfiguration zu simulieren (allein bis zu 7 Stunden, um den Vorlauf der Simulation abzuschließen). Die MATLAB-Simulation hingegen bietet – dank ihrer vereinfachten Annahmen – die Möglichkeit, Daten für sehr viele unterschiedliche Systemkonfigurationen zu berechnen [vgl. Cas13, S. 93 f.]. Die Berechnungszeit für ein System mit 1500 simulierten Zyklen beträgt lediglich zwischen 45 s und 60 s, ohne dass hierfür auf einen leistungsstarken Rechner zurückgegriffen werden muss. In der MATLAB-Simulation gibt es keine Beschränkung der Systemgrößen, einzig eine sehr hohe Anzahl an „aktiven Shuttles“ bei einer gleichzeitig geringen Anzahl an Ebenen erhöht die Wahrscheinlichkeit einer Deadlock-Situation.

Neben den grundlegenden Unterschieden in der Methodik der Simulationserstellung ergeben sich auch Unterschiede in der Darstellung der Prozesse:

Bei dem auch in der ARENA-Simulation in drei Einzelzyklen geteilten Doppelspielzyklus wird die Transferzeit zwischen Puffer und Lift nicht zum Zwischenzyklus, sondern direkt zum Auslagerungszyklus gerechnet. Das hat zwar auf das Endergebnis keine direkten Auswirkungen, macht aber deutlich, dass zwei unabhängig voneinander entwickelte Simulationen, bei gleicher Ausgangssituation, unterschiedliche Definitionen enthalten können.

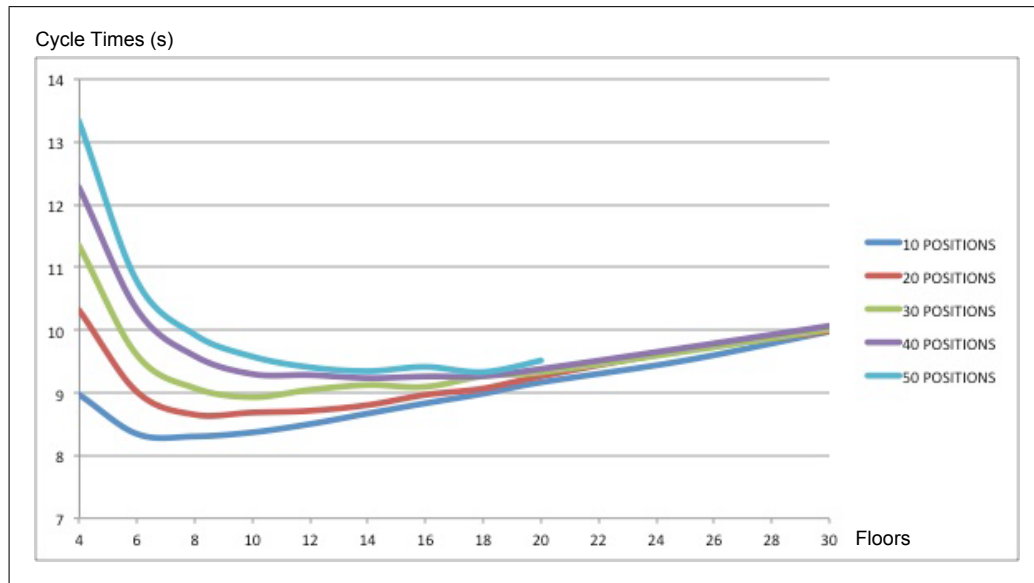
Bedeutender ist der Unterschied in der Logik der Einlagerung: Während in der MATLAB-Simulation die Koordinaten für die Einlagerung bindend sind und nur im Falle einer Deadlock-Situation abgeändert werden (vgl. Unterabschnitt 6.5.3: Koordinatenerzeugung), wird in der ARENA-Simulation die realitätsnähere Variante herangezogen: Der Lift führt keine Einlagerung in einer Ebene durch, in der der Einlagerungspuffer bereits gefüllt ist. Dieses Vorgehen hat zur Folge, dass keine Wartezeiten im Einlagerungszyklus auftreten und dieser ebenso konstant ist wie der Auslagerungszyklus. Allerdings können nun aber, falls alle Einlagerungspuffer gefüllt sind, Wartezeiten zwischen den Zyklen auftreten. Außerdem haben die Wartezeiten im Einlagerungszyklus (wie in Unterabschnitt 6.5.3 schon erläutert) einen wesentlich geringeren Einfluss auf die Doppelspielzeit als die Wartezeiten im Zwischenzyklus.

Ein deutlicher Unterschied liegt in der Festlegung der Priorität von Aus- und Einlagerung. Während bei der MATLAB-Simulation die Einlagerung priorisiert wird, wird bei der ARENA-Simulation, auch aufgrund der im vorherigen Absatz geschilderten Annahme, die Auslagerung priorisiert. Castells führt aber auch Untersuchungen mit einem sogenannten „Shuttle Parameter“ durch, um von der strikten Priorität abzuweichen und ein logisches Modell zu erarbeiten, welches zur Systemzeit entscheiden kann, ob eine Einlagerung oder Auslagerung durchzuführen ist [siehe Cas13, S. 82-91].

Die ARENA-Simulation arbeitet hauptsächlich in einem Bereich kleiner geometrischer Abmessungen und muss daher berücksichtigen, dass bei kleinen Längen der trapezförmige Verlauf im Zeit-Geschwindigkeits-Diagramm nicht erreicht wird. Der Weg, der benötigt wird, um einen solchen Verlauf zu gewährleisten, wird mit 3572 mm bestimmt. Dies entspricht einer zurückgelegten Anzahl von mehr als 11 Ebenen [siehe Cas13, S. 36]. Da bei der MATLAB-Simulation hauptsächlich höhere Systeme berechnet werden, bei denen die durchschnittliche Anzahl der durchquerten Ebenen über 11 liegen, wird die Ungenauigkeit bei kleinen Systemen in Kauf genommen.

Vergleich der Ergebnisse

Abschließend sollen nun die Ergebnisse der beiden unterschiedlichen Simulationen hinsichtlich der Relevanz der Zahlenwerte verglichen werden. Castells' Daten (Abbildung 6.56) zeigen den Verlauf der Doppelspielzeiten abhängig von der Anzahl der Ebenen bei verschiedenen Längen. Die Längen werden hier über die Anzahl der Lagerpositionen definiert; damit entsprechen 50 Positionen einer Systemlänge von 25 m. Der untere Bereich der geometrischen Abmessungen ist sehr genau aufgelöst, da eine Schrittweite von 2 Shuttles pro Simulation gewählt wurde. Diesbezüglich entspricht der qualitative Verlauf den dargestellten Ergebnissen in der vorliegenden Arbeit: Bei einer kleinen Anzahl von Ebenen treten bei steigenden Längen immer höhere Wartezeiten auf. Die den beiden Simulationen zugrunde liegende Logik lässt – trotz geringfügiger Unterschiede – dieselben Tendenzen erkennen. Zu beachten ist, dass Castells sein Hauptaugenmerk auf einen anderen Wertebereich richtet, das hat auf die Berechnung der Liftzeiten großen Einfluss; insofern lassen sich die Ergebnisse der beiden Arbeiten nicht direkt quantitativ vergleichen.

Abbildung 6.56: ARENA: Doppelspielzeiten bei konstanter Länge¹

In den Abbildungen 6.57 und 6.58 sind nun Ausschnitte aus beiden Ergebnissen dargestellt, die den vergleichbaren Wertebereich der Simulationen zeigen. Man sieht, dass Werte der MATLAB Simulation am Anfang signifikant höher sind. Dies lässt sich durch die zuvor geschilderten unterschiedlichen Annahmen (S. 81 f.) hinreichend erklären. Der steile Anstieg im ersten Bereich ist bedingt durch die geringere Auflösung der Simulation, da nur eine Schrittweite von 5 Ebenen realisiert wurde.

Bei beiden Simulationen beginnen sich die Verläufe der Kurven bei circa 18 Shuttles anzugleichen. Bei einer Höhe von 30 Shuttles unterscheiden sich auch die beiden Doppelspielzeiten in beiden Simulationen nur mehr unwesentlich, da der Einfluss der Annahme eines konstant trapezförmigen Verlaufes der Geschwindigkeit in diesem Bereich vernachlässigbar klein wird.

Es lässt sich erkennen, dass die beiden Simulationen einander nicht widersprechen, sondern, im Gegenteil, einander ergänzen: Mit der MATLAB-Simulation kann schnell ein Überblick über das tendenzielle Verhalten eines Systems erzielt werden, indem eine Vielzahl von unterschiedlichen Systemkonfigurationen simuliert wird. Sollten Werte für ein System mit geringer Shuttle-Anzahl interessieren, scheint es sinnvoll, weiterführende Untersuchungen unter Benutzung der ARENA-Simulation durchzuführen.

¹[Cas13, S. 76]

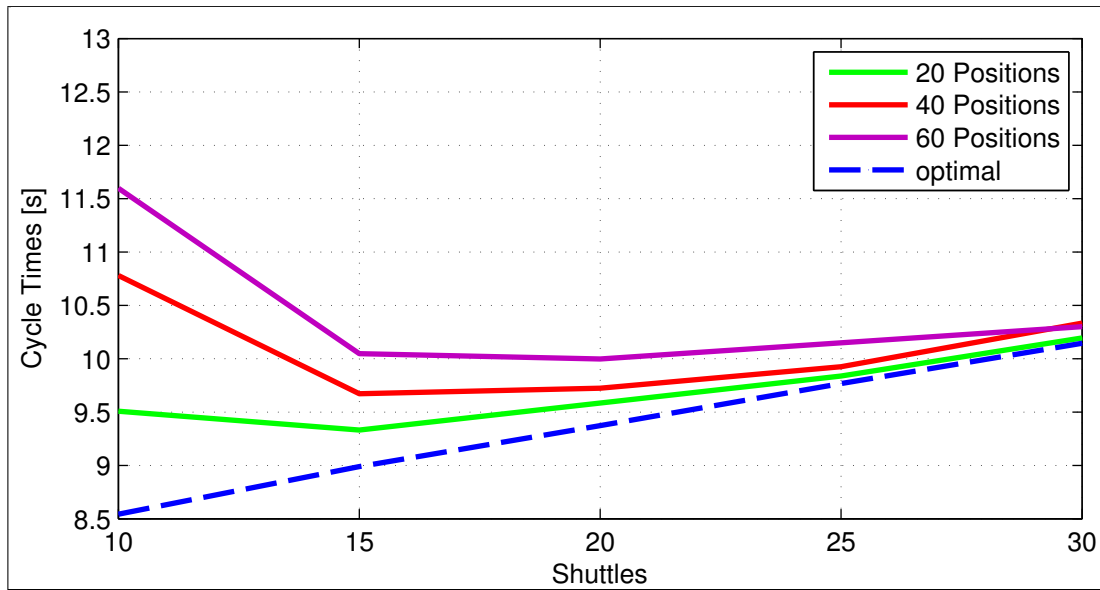


Abbildung 6.57: Vergleich der Simulationen: MATLAB

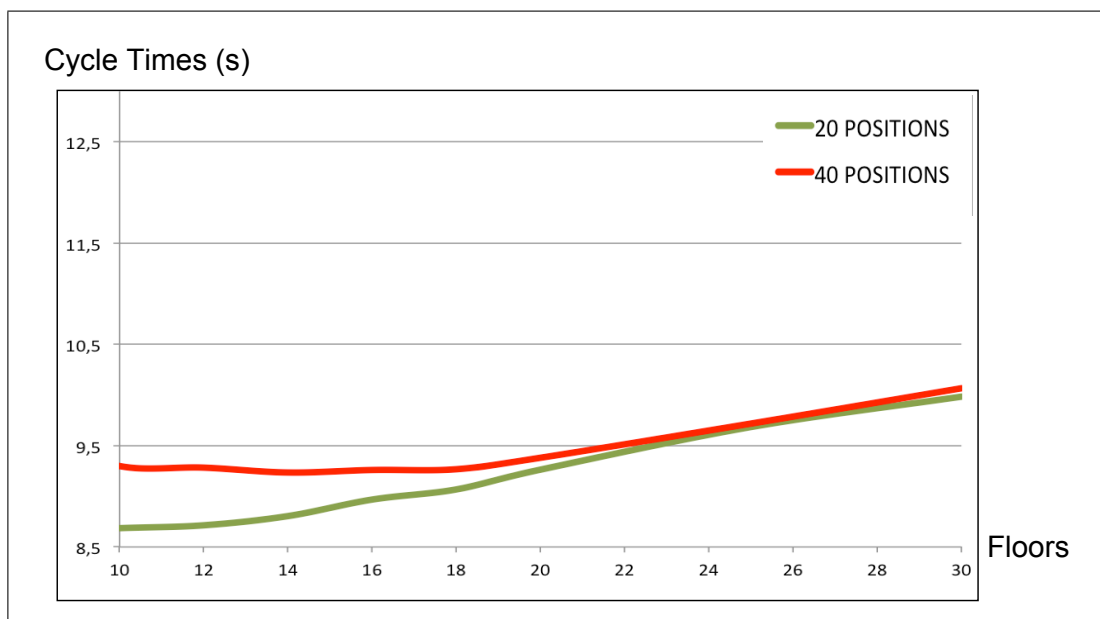


Abbildung 6.58: Vergleich der Simulationen: ARENA²

²[Cas13, S. 80]

7 Zusammenfassung und Ausblick

Das Ergebnis dieser Arbeit stellt eine voll funktionsfähige Simulation in Form eines MATLAB Programmes dar, das abgeleitet vom Prozessmodell, die gestellten Anforderungen erfüllt. Durch die Implementierung einer grafischen Benutzeroberfläche sollen eine einfache Untersuchung verschiedenster Systeme und eine „quasi intuitiv“ zu bedienende Analyse der daraus resultierenden Daten ermöglicht werden, ohne dass dafür erweiterte Kenntnisse der verwendeten Programmumgebung vorausgesetzt werden.

Aus wissenschaftlicher Sicht lassen sich aus der Analyse der durch die Simulation gewonnenen Daten viele Effekte, die bei einem Shuttle-Regal-Systems auftreten können, darstellen sowie deren Entstehung begründen. Betrachtet man bei einer ausgewählten Systemkonfiguration einzelne Doppelspielzyklen – bestehend aus Einlagerungs-, Zwischen- und Auslagerungszyklus –, so lassen sich folgende Schlüsse ziehen:

Die mittlere Doppelspielzeit eines Systems setzt sich aus einem konstanten und einem variablen Anteil zusammen. Der konstante Anteil besteht einerseits aus den Übergabezeiten, die in jedem Zyklus auftreten, und andererseits aus der durchschnittlichen Fahrzeit des Liftes, die nur von der Höhe des Systems und den kinematischen Größen abhängt. Durch die Optimierung der genannten Systemparameter lässt sich eine Senkung des konstanten Zeitanteils und damit eine Steigerung der Umschlagleistung erreichen.

Der variable Anteil ist erst dann von Bedeutung, wenn Wartezeiten in einer signifikanten Anzahl von Zyklen auftreten und damit die durchschnittliche Doppelspielzeit erhöht wird. Sowohl im Einlagerungszyklus als auch im Zwischenzyklus können, bedingt durch die Logik des Systems, solche Wartezeiten auftreten. Bei einer ungünstigen Systemkonfiguration nehmen diese einen Großteil der mittleren Doppelspielzeit ein, wobei die im Zwischenzyklus auftretenden Wartezeiten den entscheidenden Anteil ausmachen. Dadurch hat eine Optimierung der Systemparameter wenig bis keinen Einfluss mehr auf die Umschlagleistung des Systems.

Aus der Analyse des Systemverhaltens bei verschiedenen geometrischen Parametern lassen sich folgende Tendenzen feststellen:

Durch die Anzahl der Ebenen (die der Anzahl der Shuttles entspricht) und deren Höhe, wird die Gesamthöhe des Systems bestimmt. Dieser Parameter beeinflusst, wie oben geschildert, den konstanten Anteil der mittleren Doppelspielzeit und legt dadurch die niedrigste mögliche Doppelspielzeit des Systems fest, also jenen Wert der zustande kommt, wenn der Lift zu 100 % ausgelastet ist und somit keine Wartezeiten auftreten.

Dieser optimale Zustand lässt sich aber nur bei einem bestimmten Wertebereich der Längen erreichen. Jeder Anzahl der Ebenen kann ein kritischer Längenwert zugeordnet werden, bei dessen Überschreitung Wartezeiten auftreten. Diese steigen bei noch größeren Längen immer weiter an, sodass die mittleren Doppelspielzeiten ansteigen und damit die Umschlagleistung immer weiter sinkt.

Um den Effekt der simultan ablaufenden Hintergrundprozesse, also die Bewegung der Shuttles, darstellen zu können, wurde der Parameter der „aktiven Shuttles“ eingeführt. Dieser Parameter steht für die Anzahl der Auslagerungsaufträge, die in der Simulation gleichzeitig vorbereitet werden, und bleibt für die gesamte Simulation einer Konfiguration konstant.

Ein niedriger Wert der „aktiven Shuttles“ bedeutet eine geringere Auslastung des Systems. Dadurch werden große Kapazitäten des Systems nicht genutzt und die kritische Länge sinkt auf einen geringen Wert; das führt im Umkehrschluss zu hohen Wartezeiten.

Wird die Anzahl der „aktiven Shuttles“ aber erhöht, so wird eine höhere Auslastung des Systems dargestellt. In diesem Betriebszustand besticht das Shuttle-Regal-System durch eine hohe Umschlagleistung, da durch die Entkoppelung der Horizontal- und der Vertikalbewegung die Fahrzeiten der Shuttles keinen Einfluss mehr auf die Doppelspielzeiten haben.

Zusammenfassend lässt sich feststellen, dass die Umschlagleistung von vielen Parametern abhängt. Lässt man die Möglichkeit der Veränderung der Systemparameter außer Acht, so lässt sich folgender Zusammenhang beobachten: Die größte Umschlagleistung wird in Systemen von geringer Höhe und Länge erreicht; die niedrigste hingegen bei Systemen von geringer Höhe, aber großen Längen. Erhöht man die beiden geometrischen Parameter, so ist der negative Einfluss der Länge größer als jener der Höhe – dieser Effekt kann aber durch eine hohe Anzahl an „aktiven Shuttles“ kompensiert werden.

Die Möglichkeit eines schnellen qualitativen Vergleichs vieler verschiedener Systeme ist die große Stärke der vorliegenden Simulation. Das wird durch die optimierte Simulationsgeschwindigkeit, die nicht vorhandenen Restriktionen der geometrischen Parameter und die integrierte grafische Aufbereitung der Daten erreicht. Bei der Auswertung der Daten ist folgendes zu beachten: Durch die qualitative Analyse lässt sich einerseits ein Bereich ohne Wartezeiten von einem Bereich mit ansteigenden Wartezeiten unterscheiden. Während die quantitativen Aussagen im Bereich keiner oder geringer Wartezeiten mit Castells' Referenzergebnissen übereinstimmen, scheinen sie im Bereich gehäuft auftretender Wartezeiten tendenziell höher auszufallen, als nach den Resultaten von „Simulation of cycle and throughput of Shuttle Storage Systems“ zu erwarten ist.

Das diesen Erkenntnissen zugrunde liegende Programm soll in Zukunft als Basis für weitere Untersuchungen dienen, um ähnliche in der Praxis eingesetzte Systeme untersuchen zu können. Durch die Adaptierung einzelner Funktionen sollte dies bei Beibehaltung des generellen Simulationskonzeptes problemlos möglich sein. Interessant wäre hierbei insbesondere die Untersuchung von Systemen mit zwei Liften, die entweder jeweils nur Einlagerungen oder Auslagerungen durchführen oder je nach Bedarf beide Prozesse unterstützen. Ebenfalls wäre die Umsetzung von Systemen mit mehrfach tiefer Lagerung von Interesse.

Abschließend bleibt zu hoffen, dass die in dieser Arbeit gewonnenen Erkenntnisse zum besseren Verständnis der Shuttle-Regal-Systeme beitragen und den Anstoß zu weiteren Forschungen auf diesem Gebiet liefern.

Appendix

Literatur

- [AF09] ARNOLD, D. und FURMANS, K.: *Materialfluss in Logistiksystemen*. 6. Aufl. VDI. Heidelberg [u.a.]: Springer, 2009 (siehe S. 34).
- [BPM11] BPMB BERLINER BPM-OFFENSIVE: *BPMN 2.0 - Business Process Model and Notation - Poster*. 2011. URL: http://www.bpmb.de/images/BPMN2_0_Poster_DE.pdf (besucht am 26. 10. 2013) (siehe S. 10).
- [Cas13] CASTELLS I SANABRA, A.: „Simulation of cycle and throughput of Shuttle Storage Systems“. Master Thesis. Technische Universität Wien, 2013. (unpublished) (siehe S. 53, 81–84, 86).
- [FR12] FREUND, J. und RÜCKER, B.: *Praxishandbuch BPMN 2.0*. 3. Aufl. München und Wien: Carl Hanser Verlag, 2012 (siehe S. 8, 9).
- [Hei11] HEIDENBLUT, V.: „Kompaktlagersysteme im Vergleich. Pro und Contra ausgewählter Technologien“. In: *f+h* 5 (2011), S. 222–226 (siehe S. 1, 2).
- [Hep11] HEPTNER, K.: „Shuttles unter der Lupe“. In: *dhf* 5 (2011), S. 17–19 (siehe S. 3).
- [Kna12a] KNAPP: „Multichannel-Lösung für den E-Commerce“. In: *dhf* 10 (2012), S. 24–26 (siehe S. 3).
- [Kna12b] KNAPP: „Shuttle-Lager unterstützt Multichannel-Vertrieb. Ausgewogenes Zusammenspiel von manuellen und automatisierten Prozessen“. In: *f+h* 9 (2012), S. 38–40 (siehe S. 3).
- [OMG11] OMG: *Business Process Model and Notation (BPMN)*. 2011. URL: <http://www.omg.org/spec/BPMN/2.0/PDF> (besucht am 26. 10. 2013) (siehe S. 8).
- [Ste12] STEIN, U.: *Programmieren mit MATLAB*. 4. Aufl. München: Carl Hanser Verlag, 2012 (siehe S. 7).