Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization: Transport Engineering and Logistics

Report number: 2017.TEL.8142

Title: **Combining fixed-path and free-range AGV routing on a container terminal**

Author: D.L. Overbeek

Title (in Dutch) Het combineren van vaste-paden en free-range AGV routering op een container terminal

Assignment: Masters thesis

Confidential: no

Initiator (university): Ir. M.B. Duinkerken

Supervisor: Ir. M.B. Duinkerken

Date: June 20, 2017

This report consists of 88 pages and 6 appendices. It may only be reproduced literally and as a whole. For commercial purposes only with written authorization of Delft University of Technology. Requests for consult are only taken into consideration under the condition that the applicant denies all legal rights on liabilities concerning the contents of the advice.

Preface

This report is the result of my graduation project to complete the Master of Science Transport Engineering and Logistics at the TU Delft. During my master I discovered an interest in the simulation of logistical processes. I am glad that my supervisor Ir. M.B. Duinkerken made it possible to do a graduation project on this subject. Starting my project, I hoped to learn more on the simulation of logistic processes. Concluding this report, I can say that I certainly achieved this goal and learned a lot.

I would like to thank my supervisor Ir. M.B. Duinkerken for his enthusiasm of my proposal “to do something with AGV routing”, making this project possible and for his feedback and ideas during this project. Secondly, I would like to thank Dr. R.R. Negenborn for his advice during the official progress meetings.

Thanks to all family and friends who supported me during my study. In particular my mother, who supported me during all ups and downs and was determined on the fact that I would finish a master. My boyfriend Stefan, your enthusiasm for programming and all your hobby project proposals were sometimes overwhelming but also made me enthusiastic about programming. Thank you for teaching me the basics, your patience when I lost mine and your overall support during my study.

D.L. Overbeek
Delft, June 2017

Summary

Since the introduction of containers in 1955, container freight has grown enormously. This trend is expected to continue in the future, putting a pressure on container terminals to improve performances. Automation of equipment is required to increase the terminal performance whilst keeping operational costs under control. Some automation is already in operation at the terminal, think of quay cranes (QCs), stacking cranes (ASCs) and automated guided vehicles (AGVs). AGVs are in operation since 1993, their routing is the focus of this report.
Traditionally, the AGVs drive on a topology of paths from quay to stack and vice versa. This topology is designed such that traffic control is most reliable. The drawbacks of this routing type is the lack of flexibility and scalability. Traditional routing will not suffice in future growth scenarios.
A new type of routing is proposed in literature: free-range routing. This type does not require a topology of paths and is thus very flexible. A disadvantage is the risk of deadlocks, this will happen in crowded, chaotic zones such as near the cranes. This problem is not yet mentioned in literature, because (the few) free-range routing studies focus on routing performance instead of terminal performance. Often small AGV fleets are assumed that do not experience dense areas. The problem of crowds is mentioned and studied in robotics, a similar field of study.
Fixed-path routing performs quite well in crowded areas and therefore a mix of fixed-path and free-range routing is proposed: mixed routing. Fixed-path routing is applied near the cranes while free-range routing is applied in the remaining part of the terminal. Both terminal and routing performance of mixed routing are compared to the performances of free-range routing and traditional routing.

The system assumed is that of the waterside area of a container terminal. QCs and ASCs are on the boundaries of the system. AGVs transport containers from stack to quay, i.e. only the loading process is assumed. Cycle times of the cranes are modeled as stochastic behavior. A terminal model, including 4 QCs and 14 ASCs, is implemented in Borland Delphi including TOMAS. Each QC has four unloading spots, each ASC has one loading spot. The terminal model has three routing variants:

1. Traditional routing. A grid-like topology is assumed in which x-directional lanes are one-directional and y-directional lanes are bidirectional. Lanes consists of arcs between nodes. Each node can only be claimed by one AGV simultaneously. AGVs drive from node to node. Routing goes incrementally: a Dijkstra algorithm calculates each iteration its shortest, available path towards the destination.
2. Free-range routing. The potential field method (PFM) is chosen to be the most suited controller, this is a reactive method that does not plan. The nearby environment of the AGV is sensed for obstacles and those are avoided en-route. The main advantage of the PFM is its low computational burden, the main disadvantage is the risk of deadlocks.
3. Mixed routing. This routing type is a combination of the former two routing types: the basis is free-ranging, but a topology of paths is implemented near the QCs on which the AGVs are traditionally routed. Routing transfer points are used to switch routing types of the AGV.

The terminal model and all three routing types are verified and validated.

Experiments are performed to obtain the AGV costs per routing type. The minimum amount of AGVs needed to reach a 95% terminal performance threshold for different QC capacities is determined. Amount of AGVs is then converted to costs, which consist of fixed costs and variable costs. The results are explained with terminal and routing performance as a function of number of AGVs. Terminal

performance is quantified as handling efficiency at the QCs. Routing performance is quantified as the factor of realized travel times over minimal travel time. The amounts of AGVs in the model are varied from 4 to 23 and the QC capacity is varied from 30 moves/hr/QC to 70 moves/hr/QC, intermediate steps are 1 and 10, respectively.

It was expected that mixed routing combined the benefits of free-range and traditional routing, but results show that the new routing type does not live up to these expectations. Results must be explained in twofold: on the one hand for low QC capacities up to 50 moves/hr/QC and on the other hand for high QC capacities exceeding 50 moves/hr/QC.
At low QC capacities, few AGVs are required to fulfill the container infeed at the QCs and thus there is little to no congestion. The travel distances of the routing types are leading for the terminal performance. Free-range routing has the shortest paths and thus performs best, followed by mixed and traditional routing. Latter two routing types perform quite similar. In terms of costs, free-range routing is the cheapest option while mixed and traditional routing have similar costs that are more expensive than free-range. The costs of all routing types increase proportional as QC capacity increases. Routing performance showed little delay for mixed and traditional routing, free-range routing has increased delay as QC capacity decreases and more AGVs are in operation.
At high QC capacities, differences in performance are seen between the routing types. In terms of costs, mixed and free-range routing increases fast as QC capacity increases. The costs of traditional routing also increases, but at a slower rate. Free-range routing remains the cheapest type but traditional routing is slowly becoming competitive. The 95% threshold could not be reached when QC capacity exceeded 62 or 63 moves/hr/QC (mixed versus traditional and free-range routing, respectively). Clearly, congestion of free-range routing affected performance. Routing performance showed an increase in delay as more AGVs are in operation, many small deadlocks occurred. Delay decreases as QC capacity increases because traffic is more spaced across the terminal. Delay in traditional routing remains quite stable, alternative routes are immediately sought. Mixed routing shows unexpected delay: delay increases as more AGVs are in operation and QC capacity increases. It appeared that the free-range zone is too narrow and thus the problem of crowding is shifted towards the routing transfer points. This explains the vast increase in costs of mixed routing.

Free-range routing showed its potential as a routing type, but has some serious issues that must be resolved in order to fulfill future terminal throughput. Mixed routing is still advised as a routing type that might be beneficial. Unfortunately, the implementation as in this report appeared not the be the solution. It is recommended to develop a topology that does make use of the benefits of both free-range and traditional routing. For example by giving arriving AGVs priority over departing AGVs near the routing transfer point towards QC or by using bidirectional paths and multiple routing transfer points when heading towards the QC. Once proven as routing type, a new challenge lies in optimization of terminal interactions.

Summary in Dutch

Containervracht is enorm gegroeid sinds de introductie van de container in 1955. De verwachting is dat deze groei doorzet in de toekomst, daarbij een druk uitoefenend op container terminals die hun prestaties moeten verbeteren. Automatisering is nodig om de terminal prestatie te verhogen en operationele kosten in de hand te houden. Sommige automatisering is al geïmplementeerd in een terminal, denk aan kadekranen en opslagkranen. Automatisch geleide voertuigen (AGVs) zijn in operatie sinds 1993 en zijn onderwerp van dit onderzoek.
Oorspronkelijk rijden deze AGVs op een topologie van paden van kade naar opslag en vice versa. Deze topologie is zo ontworpen dat het verkeer op de meest betrouwbare manier kan opereren. Het nadeel van dit routeringstype is het gebrek aan flexibiliteit en schaalbaarheid. Traditionele routering zal niet voldoen in toekomstscenario's.
Een nieuwe routeringstype is voorgesteld in de literatuur: free-range routering. Dit type heeft geen topologie van paden nodig en is dus zeer flexibel. Een nadeel is het risico op deadlocks, deze zullen optreden in drukke, chaotische zones zoals bij de kranen. Dit probleem is nog niet genoemd in de literatuur, omdat de weinige free-range studies zich focussen op de routeringsprestatie in plaats van terminalprestatie. Het probleem van drukke zones is bestudeerd in het vergelijkbare veld van robotica. Vaste paden routering presteert goed in drukke zones en daarom is een mix van vaste paden en free-range routering voorgesteld: gemixte routering. Vaste paden routering is toegepast bij de kranen en free-range routering is toegepast in het overig gebied van de terminal. Terminal- en routeringsprestaties van gemixte routering worden vergeleken met de prestaties van free-range en traditionele routering.

Het aangenomen system is de waterkant van een container terminal. Kadekranen en opslagkranen staan op de grenzen van het systeem. AGVs vervoeren containers van opslag naar kade, alleen het laadproces is aangenomen. Cyclustijden van de kranen worden gemodelleerd met stochastisch gedrag. Een terminal model, inclusief 4 kadekranen en 14 opslagkranen, is geïmplementeerd in Bordland Delphi inclusief TOMAS. Elke kadekraan heeft vier uitlaad plekken, elke opslagkraan heeft één laad plek. Het terminal model heeft drie routerings-varianten:

1. Traditionele routering. Een roosterachtige topologie is aangenomen waarin de paden in x-richting één-directioneel zijn en de paden in y-richting zijn bidirectioneel. Paden bestaan uit bogen tussen knooppunten. Elk knooppunt kan door slechts één AGV tegelijkertijd worden geclaimd. AGVs rijden van knooppunt naar knooppunt. Routering gaat incrementeel: een Dijkstra algoritme berekend elke iteratie het kortste, beschikbare pad richting de bestemming.
2. Free-range routering. De potentiele veld methode (PVM) is gekozen als meest geschikt, dit is een reactieve methode die niet plant. De nabije omgeving van de AGV word gescand op obstakels en die worden onderweg vermeden. Het voornaamste voordeel van de PVM is haar lage rekenkracht die vereist is, het voornaamste nadeel is het risico op deadlocks.
3. Gemixte routering. Dit routeringstype is een combinatie van de vorige twee routeringstypen: de basis is free-range, maar een topologie van paden is geïmplementeerd bij de kadekranen waarop de AGVs traditioneel gerouteerd worden. Routering wissel punten worden gebruikt om te wisselen tussen routeringstypen.

Het terminal model en alle drie routeringstypen zijn geverifieerd en gevalideerd.

Experimenten worden uitgevoerd om de AGV kosten per routeringstype te bepalen. Het minimale aantal AGVs nodig om 95% terminalprestatie te behalen voor verschillende kadekraancapaciteiten

wordt bepaald. Het aantal AGVs wordt omgezet naar kosten, die bestaan uit de vaste kosten en variabele kosten. De resultaten worden uitgelegd met terminal en routering prestaties als een functie van aantal AGVs. Terminal prestatie is uitgedrukt als omslag efficiëntie bij de kadekranen. Routeringprestatie is uitgedrukt als de factor van gerealiseerde rijtijden over minimale rijtijden. Het aantal AGVs in het model wordt gevarieerd van 4 tot 23 en de kadekraancapaciteiten worden gevarieerd van 30 tot 70 handelingen/uur/kadekraan, tussenliggende aantallen zijn 1 en 10, respectievelijk.

De verwachting was dat gemixte routering de voordelen van zowel free-range als traditioneel routering zou combineren. De resultaten laten echter zien dat deze verwachting niet waar wordt gemaakt. De resultaten moeten enerzijds uitgelegd worden voor lage kadekraancapaciteiten tot 50 handelingen/uur/kadekraan en anderzijds voor hoge capaciteiten vanaf 50 handelingen/uur/kadekraan. Voor lage kadekraancapaciteiten zijn weinig AGVs nodig om in de container aanvoer te voorzien. Er is weinig tot geen vertraging. De rij afstanden van de routeringstypes zijn leidend voor de terminalprestatie. Free-range routering heeft de kortste paden en presteert het beste, gevolgd door gemixte en dan traditionele routering. De laatste twee types presteren vergelijkbaar. Qua kosten is free-range routering het goedkoopst terwijl gemixt en traditionele routering dezelfde kosten hebben die hoger liggen als free-range. De kosten van alle routeringstypen neemt proportioneel toe als kadekraancapaciteit toeneemt. Routeringprestatie laat weinig vertraging zien voor gemixt en traditioneel, free-range heeft meer vertraging als kadekraancapaciteit afneemt en er meer AGVs zijn. Voor hoge kadekraancapaciteiten zijn meer verschillen zichtbaar tussen de routeringstypen. Kosten nemen rap toe bij gemixte en free-range routering als kadekraancapaciteit toeneemt. De kosten van traditionele routering nemen ook toe maar in een langzamer tempo. Free-range routering blijft het goedkoopste type maar traditionele routering komt langzaam in de buurt. De 95% grens kon niet worden behaald zodra kadekraancapaciteit hoger werd dan 62 of 63 handelingen/uur/kadekraan (gemixt versus traditioneel en free-range routering, respectievelijk). Vertraging van free-range routering heeft duidelijk invloed op de prestatie. Routeringsprestatie laat een snelle toename in vertraging zien wanneer meer AGVs in operatie zijn, veel kleine deadlocks treden op. Vertraging neemt af als kadekraancapaciteit toeneemt, omdat het verkeer meer verspreid is over de terminal. Vertraging bij traditionele routering blijft stabiel, alternatieve routes worden onmiddellijk gezocht. Gemixte routering laat onverwachte vertraging zien: vertraging neemt toe naarmate meer AGVs in operatie zijn en kadekraancapaciteit toeneemt. Het blijkt dat de free-range zone te smal is waardoor het probleem van drukke zones verschoven is naar de routering wissel punten. Dit verklaart de snelle toename van vertraging in de kosten van gemixte routering.

Free-range routering heeft zijn potentie laten zien als een routeringstype, maar vereist nog wel het oplossing van sommige problemen. Gemixte routering is nog steeds geadviseerd als routeringstype dat voordelen ten opzichte van free-range kan brengen. Helaas is de implementatie van dit verslag niet de goede oplossing. Het wordt aangeraden een topologie te ontwikkelen die wel gebruik maakt van de voordelen van zowel free-range als traditioneel routering. Bijvoorbeeld door het gebruik van bidirectionele paden en meer routering wissel punten of door voorrang te verlenen aan arriverende AGVs over vertrekkende AGVs bij een routering wissel punt. Wanneer dit routeringstype goed functioneert resteert de optimalisatie van alle terminal interacties om zo aan de toekomstige vraag te voorzien.

List of abbreviations

AGV	Automated guided vehicle
ASC	Automated stacking crane
FCFS	First come first serve
FIFO	First in first out
KPI	Key performance indicator
PDL	Process description language
PFM	Potential field method
QC	Quay crane

Contents

Preface iii
Summary v
Summary in Dutch vii
List of abbreviations ix
List of tables xv
List of figures xvii
Chapter 1 - Introduction 1
1.1 Problem statement 1
1.2 Research question 2
1.3 Approach and structure of this report 2
Chapter 2 - System analysis 5
2.1 Terminal – physical system 5
2.2 Terminal – logistic system 7
2.3 KPI's of the system 11
2.4 Conclusions 12
Chapter 3 - Review of related literature 13
3.1 Routing on a traditional container terminal 14
3.1.1 Choice between fixed or dynamic routes 16
3.2 Free-ranging 17
3.2.1 Motion planning methods 18
3.2.2 Choice of method 23
3.3 Guidance of free-ranging AGVs 25
3.4 Conclusions 27
Chapter 4 - Terminal model 29
4.1 Model description 29
4.2 Control and interaction of subsystems in the terminal model 34
4.3 Implementation 37
4.4 Conclusions 38
Chapter 5 – Routing types within the terminal model 39
5.1 Traditional routing of AGVs 39
5.1.1 Description and assumptions of routing 39
5.1.2 Traditional routing PDL 40

5.1.3 Implementation 41
5.2 Free-range routing of AGVs 43
5.2.1 Description and assumptions of routing 43
5.2.2 Potential field routing PDLs 44
5.2.3 Implementation 45
5.3 Mixed routing 50
5.3.1 Description and assumptions of routing 50
5.3.2 Mixed routing PDL 51
5.3.3 Implementation 51
5.4 Conclusions 53
Chapter 6 - Verification and validation 55
6.1 Terminal model including traditional routing 55
6.1.1 Verification 55
6.1.2 Validation 57
6.2 Free-range routing 59
6.2.1 Verification 59
6.2.2 Validation 60
6.3 Mixed routing 61
6.3.1 Verification 61
6.3.2 Validation 62
6.4 Conclusions 62
Chapter 7 – Experimental plan 63
7.1 Experiment setups 65
7.2 Sensitivity analysis 67
7.3 Conclusions 70
Chapter 8 – Results and discussion 71
8.1 Costs of the AGV system 71
8.2 Terminal performance 73
8.3 Routing performance 75
8.4 Computational times 79
8.5 Conclusions 80
Chapter 9 - Conclusions and recommendations 81
9.1 Conclusions 81

9.2 Recommendations 84
References 85
Appendix A - Scientific research paper 89
Appendix B - Related literature (Chapter 3) 97
Appendix C - Terminal model (Chapter 4) 101
Appendix D - Routing types (Chapter 5) 103
D.1 Traditional routing PDLs and Usage of buffer places 103
D.2 Free-range routing PDLs and Usage of buffer places 106
D.3 Mixed routing PDLs and Usage of buffer places 108
Appendix E – Verification and validation (Chapter 6) 111
E.1 Traditional routing and terminal model 111
E.2 Free-range routing 120
E.3 Mixed routing 124
Appendix F – Data of results (Chapter 8) 129

List of tables

Table 1 - Central versus decentral control. 10
Table 2 - Equipment characteristics of the ECT Delta terminal. 11
Table 3 - Advantages and disadvantages of static versus dynamic routing. 16
Table 4 - Planning types of free-ranging methods. 18
Table 5 - Comparison of free-ranging methods on the criteria points. 24
Table 6 - Input parameters of terminal model. 32
Table 7 - Implemented standard input values of terminal model. 37
Table 8 - Potential field routing input parameters. 46
Table 9 – Implemented input parameters for potential field routing. 48
Table 10 - Verification input parameters. 56
Table 11 - Results of analytical and simulation models per element, traditional routing. 56
Table 12 - Comparison of results by Liu et al. [60] and terminal model. 58
Table 13 - Comparison of results by Bae et al. and terminal model. 58
Table 14 - Results of analytical and simulation models per element, free-range routing. 59
Table 15 - Results of analytical and simulation model per element, mixed routing. 61
Table 16 - Required number of replication per routing type. 64
Table 17 – Fixed input parameters of experiments. 64
Table 18 - QC capacity inputs of the experiments. 66
Table 19 - Varying input parameters of experiments. 66
Table 20 - Influence of container retrieval time on terminal and routing performance. 67
Table 21 - Influence of AGV speed on terminal and routing performance. 68
Table 22 - Influence of routing obstacle-avoidance parameters on terminal and routing performance, 16 AGVs. 68
Table 23 - Influence of routing obstacle-avoidance parameters on terminal and routing performance, 23 AGVs. 69
Table 24 - Number of AGVs required to reach 95% terminal performance. 72
Table 25 - Overview of studied literature. 99
Table 26 - Seed independence tests. 112
Table 27 - Consistency tests. 112
Table 28 - Average drive times and distances of AGV. 113
Table 29 - M/D/1 and simulation model results for ASC and QC systems. 113
Table 30 - Results of the M/D/1 model and deterministic simulation model for the container process. 114
Table 31 - Results of the M/D/1 model and deterministic simulation model for the ASC process. 114
Table 32 - Results of the M/D/1 model and deterministic simulation model for the QC process. 115
Table 33 - Results of the M/D/1 model and deterministic simulation model for the AGV process. 115
Table 34 - Results of the hand event tracing of the container process. 117
Table 35 - Results of the hand event tracing for the AGV process. 117
Table 36 - Equipment characteristics of Norfolk International Terminal. 118
Table 37 - Equipment characteristics of Norfolk International Terminal. 119
Table 38 - Input parameters as used by Bae et al. [9]. 119
Table 39 - Average drive times and distances of AGV. 120
Table 40 - Results of the M/D/1 model and simulation model for the container process, free-range routing. 120
Table 41 - Results of the M/D/1 model and deterministic simulation model for the ASC process. 121
Table 42 - Results of the M/D/1 model and deterministic simulation model for the QC process. 121
Table 43 - Results of the M/D/1 model and deterministic simulation model for the AGV process. 121
Table 44 - Results of the hand event tracing of the container process. 123
Table 45 - Results of the hand event tracing for the AGV process. 123
Table 46 - Average drive times and distances of AGV. 124

Table 47 - Results of the M/D/1 model and deterministic simulation model for the container process, free-range routing. 124
Table 48 - Results of the M/G/1 model, simulation model, M/D/1 model and deterministic simulation model for the ASC process. 125
Table 49 - Results of the M/D/1 model and deterministic simulation model for the QC process. 125
Table 50 - Results of the M/D/1 model and deterministic simulation model for the AGV process. 125
Table 51 - Results of the hand event tracing of the container process. Shown left. 127
Table 52 - Results of the hand event tracing for the AGV process. Shown right. 127
Table 53 – AGV lifetime costs for 95% terminal performance, corresponding AGVs and computational times. 129
Table 54 - Terminal performance for different QC capacities. 130
Table 55 - Routing performance for different QC capacities. 130

List of figures

Figure 1 – Future scenarios of throughput in port of Rotterdam [1] *1*
Figure 2 - Structure of this report. *3*
Figure 3 - Terminal layout [4]. *5*
Figure 4 - PROPER model of the container terminal system. *7*
Figure 5 - Import process of containers. *8*
Figure 6 - Export process of containers. *8*
Figure 7 - Three level planning model. *9*
Figure 8 - Control architectures [15]. *10*
Figure 9 - Terminal topologies. Left: Loop [7]. Center: Mesh [7]. Right: Tandem. *14*
Figure 10 – Example of a grid-like topology [9]. *15*
Figure 11 - Circular zone sizes [28]. *15*
Figure 12 - The use of semaphores [2]. *16*
Figure 13 - Cell decomposition methods [41]. Left: Exact cell decomposition. Middle: Approximate cell decomposition. Right: Adaptive cell decomposition. *19*
Figure 14 - Roadmap construction methods [41]. Left: Visibility graph. Right: Voronoi diagram. *20*
Figure 15 - Construction of total potential field [41]. a) obstacles in the field, b) attractive potential to target, c) repulsive potential of obstacles, d) total potential field of b and c, e) vectors representing the total potential field of d, f) contours of the total potential field and an example of a path. *20*
Figure 16 – Virtual force construction [14]. Left: Detection of obstacles. Right: Related virtual force model. *21*
Figure 17 - Virtual force method [44]. *21*
Figure 18 - Model predictive control [48]. *22*
Figure 19 - Arching and clogging effect [57]. Left: Initial crowd distribution. Right: Crowd during evacuation. *26*
Figure 20 - Lane formation [58]. *26*
Figure 21 – Terminal model topology. *30*
Figure 22 - Element interactions of the terminal model. *32*
Figure 23 - System and control cycle. *34*
Figure 24 - Control architecture of terminal model. *34*
Figure 25 - ASC control cycle. *35*
Figure 26 - AGV control cycle. *35*
Figure 27 - QC control cycle. *36*
Figure 28 - Animation in the GUI (traditional routed terminal). *38*
Figure 29 – Traditional routing terminal implementation in Delphi. *42*
Figure 30 - Representation of AGV in the PFM. *43*
Figure 31 – Left: Influence of beta on the avoidance trajectory. Right: Influence of beta on the speed profile. *48*
Figure 32 - Potential field routing terminal implementation in Delphi. *49*
Figure 33 - Mixed routing terminal topology, AGVs represented with orange. *50*
Figure 34 - Mixed routing terminal implementation in Delphi. *51*
Figure 35 - Hold positions for routing transfer point in mixed routing. *52*
Figure 36 - Interaction between elements in the container handling process. *56*
Figure 37 - Warm up and steady state of simulation run. *63*
Figure 38 - Lifetime costs of the AGV system. *71*
Figure 39 - Terminal performance for different QC capacities. *74*
Figure 40 – AGV reaching a QC unload spot between two occupied spots. Left: Trajectory of AGV. Right: Speed per iteration. *75*
Figure 41 - Deadlocks at QCs (red circles). *76*
Figure 42 - Snapshot of mixed routing at a QC capacity of 30 moves/hr/QC. *77*
Figure 43 - Snapshot of mixed routing at a QC capacity of 70 moves/hr/QC. *77*

Figure 44 - Routing performance for different QC capacities. *78*
Figure 45 - Computational times of 95% terminal performance. *79*
Figure 46 - GUI. *102*
Figure 47 - Crossings and buffer places used when AGV drives from ASC to QC. *105*
Figure 48 - Buffer places choices and crossing when driving towards ASC. *105*
Figure 49 - Waiting AGVs to be routed to an unloading spot. *106*
Figure 50 - Buffer place choices and crossings when driving towards ASC. *107*
Figure 51 - Buffer place choices and transfer points when driving towards ASC. *109*
Figure 52 - Model output. *111*
Figure 53 - Terminal performance as obtained by Liu et al. [60]. *119*
Figure 54 - Results from Bae et al. [9]. *119*
Figure 55 - Average waiting time per QC in verification run. *120*
Figure 56 - Average waiting time of AGV per QC. *124*
Figure 57 - Computational times for different QC capacities. *129*

Chapter 1 - Introduction

Since the introduction of the first container in 1955, container freight has grown enormously. Even during the recent economic crisis, container throughput in the port of Rotterdam grew 27% over the years 2006 to 2016 [1]. This growth is expected to continue in the future.
The port of Rotterdam investigated four future scenarios based on growth of world trade and speed of energy transition. The results are presented in Port Vision 2030 in which the situation in 2010 is chosen as starting point. Over the past years it appeared that the scenarios are somewhat optimistic. The most recent revised scenarios are presented in Figure 1. The port expects a growth in container throughput for all scenarios.

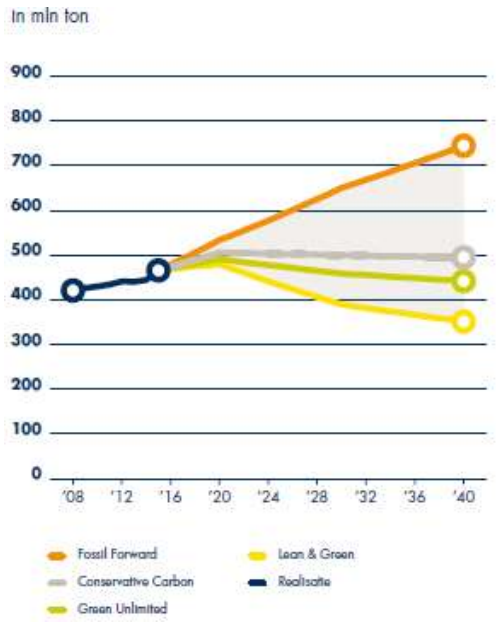


Figure 1 – Future scenarios of throughput in port of Rotterdam [1]

This expected growth puts a pressure on container terminals. In order to remain competitive with other terminals, throughputs must be increased by optimizing or redesigning terminal processes. Automation of equipment is required to increase the terminal performance whilst keeping operational costs under control. Nowadays, automated quay cranes (QCs), stacking cranes (ASCs) and vehicles are in operation, but the development has certainly not reached its end.

Automated guided vehicles (AGVs) are innovative vehicles that positively contribute to terminal performance. The faster they transport containers, the faster a vessel is handled and new ones can arrive. AGVs were introduced at first at the ECT Delta terminal in Rotterdam in 1993 and are subject to many studies to optimize their performance [2].
Traditionally, the AGVs drive on a topology of fixed, pre-determined paths to transport containers. The topology of these paths is designed such that routing is most reliable.

1.1 Problem statement

Routing on fixed paths is sufficient for now, but it might become a bottleneck for future growth scenarios, because it lacks flexibility and scalability. A new type of routing is needed to operate the AGVs.

Since some years, free-ranging AGVs are studied [3]. This is an application of a mobile robot, which is well studied in the robotics field. Free-ranging AGVs are not bound to fixed paths and thus have infinite route options. The travel distances are reduced because the AGV can drive in a straight line towards its destination. This improves the flexibility, throughput and costs of the AGV system.
Some drawbacks must be taken into account though. The required computational power of free-range methods is often high and there is a risk of congestions and deadlocks while avoiding obstacles. The advantage of free-ranging over fixed path routing can get lost if the controller has no proper way to deal with AGVs in crowded, chaotic zones. Crowded zones in a container terminal are near cranes.

Fixed-path routing has fewer problems with crowded zones and should therefore be combined with free-range routing such that the benefits of both routing types are used. It is suggested that fixed-paths are implemented near the cranes. This new routing type will be called mixed routing and is developed and tested throughout this report. Applying this new routing type, it is expected that the AGVs drive efficiently in crowded areas.
The suggested routing type is not studied in literature, the problems rising due to crowded areas are also not mentioned in literature. The few papers that study free-ranging AGVs are more focused on the development of a method and show their solutions with a small AGV fleet. Problems rising as a consequence of a large system are not studied. The robotics field does recognize the problem of navigating in crowds.

The goal of this research is to evaluate mixed routing in terms of terminal and routing performance. A comparison will be made to traditional (fixed-path) routing and free-range routing.

1.2 Research question

The following research question is proposed:
What is the effect of combined fixed-path and free-range AGV routing on the routing and terminal performance of an automated container terminal?

And the following sub questions:

1. What terminal operations take place in the considered part of an automated terminal and what are the key performance indicators of terminal and routing performance?
2. How is fixed-path routing (“traditional routing”) arranged in container terminals and what is its performance?
3. What are the implementation possibilities for free-range routing into a container terminal and what is its performance?
4. How can traditional and free-range routing be combined (“mixed routing”) and what is its performance?

1.3 Approach and structure of this report

This report consists of the following chapters:
Chapter 2 provides an analysis of the container terminal, both the physical and logistical system of a container terminal are explained. The considered system with its boundaries is described and common

key performance indicators are given. This chapter answers sub question 1.
Chapter 3 is devoted to the literature study. Literature is studied for three subjects: traditional routing in terminals that use AGVs, free-range methods that can be applied to AGVs in the system and solutions to guide AGVs in crowded zones. A free-range method will be chosen.
Chapter 4 describes the terminal model, which is a synthetization of the information gathered in chapter 2 and 3. The assumptions for the terminal model are stated and the elements and its interactions are explained. Also the implementation is explained. The routing part of the terminal model is however not part of this chapter, this is explained in Chapter 5. The model developed in this chapter is the basis of the models to answer sub questions 2,3 and 4.
Chapter 5 explains the three routing types and how they are implemented into the terminal model. The topology of the terminal and traffic control are discussed per routing type. This chapter is of importance for sub questions 2,3 and 4.
Chapter 6 describes the verification and validation of the terminal model and the routing types. Verification is needed to make sure the model does what it is intended to do. Validation checks if the model is correct to describe the system. This chapter is of importance for sub questions 2,3 and 4.
Chapter 7 explains the experimental plan that will be executed in order to answer the research question. This chapter is of importance for sub questions 2,3 and 4.
Chapter 8 processes the outcomes of the experiments into results. The key performance indicators are the indicators on which the results will be compared. A discussion of the results is part of this chapter. This chapter is of importance for sub questions 2,3 and 4.
Chapter 9 is the final chapter, the conclusions of the research are drawn and the research question and sub questions will be answered. Also further recommendations are presented.

The following figure shows an overview of the structure of this report:

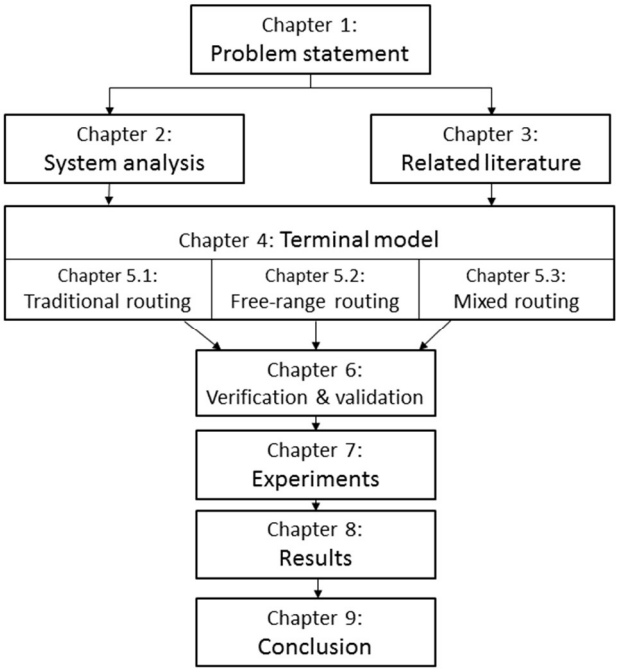


Figure 2 - Structure of this report.

Chapter 2 - System analysis

A container terminal is a hub where modal changes in container transport are made. In- and outgoing modes are for example deep-sea vessel, truck, train and barge. In a container terminal, containers are stored in a stack between two mode transfers.

The terminal can be analyzed from a physical and a logistical view. The physical view concerns the terminal area and its resources, Chapter 2.1. The logistical part concerns the processes that take place to operate the terminal, Chapter 2.2.

2.1 Terminal – physical system

Terminal areas

Figure 3 shows a typical container terminal layout. A terminal can be roughly divided into five areas, from top to bottom:

1. Water- or quayside — The vessels berth at the quay and containers are (un)loaded here.
2. Waterside transport — Containers are transported from the quay (stack) to the stack (quay). In an automated container terminal, this area uses automated vehicles.
3. Stack — Area where containers are stored until their next transport mode is ready to load.
4. Landside transport — Containers are transported from the stack to truck or rail mode or vice versa.
5. Landside — Area where equipment that travels over land is (un), such as trains or trucks.

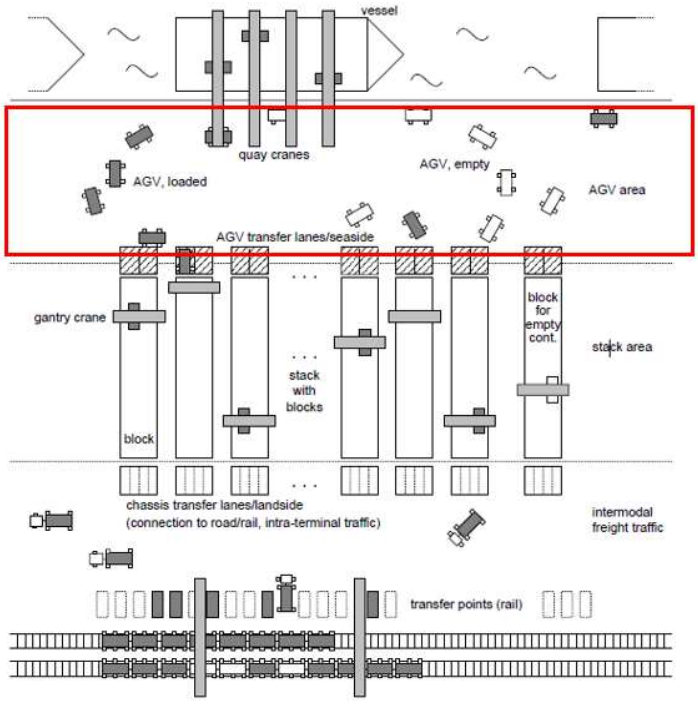


Figure 3 - Terminal layout [4].

System boundaries

The system in this report is focused on the waterside transport area, the boundaries are represented by the red rectangle in Figure 3. This area can be seen as a black box in which containers flow in and out. Containers can only enter or exit the system via a quay crane or an automatic stacking crane. The AGV system is entirely within this black box and thus the AGV system is assumed to be a closed system. The cranes have an open system, because they are located at the system boundaries. This report will model the behavior of these cranes to the extent that is relevant to the chosen system. Exact container locations are for example not taken into account. The exact implementation of the model and its assumptions will be later explained.

Equipment in the system

The equipment found within and on the system boundaries are:

6. Quay cranes at the quayside:
   Their function is to (un)load the containers (from) into the vessel and thus transfer the container. Quay cranes can be manually operated or automated. A spreader on this crane can lift a 45ft, 40ft, 30ft or one or two 20ft containers.
7. Automated guided vehicles (AGVs) at the transportation area:
   Their function is to transport containers from the quay to the stack or vice versa. An AGV is a passive vehicle [4]; it cannot perform lifts, it thus needs another piece of equipment to lift and pick up containers onto and from the AGV. An ALV (automated lifting vehicle) is similar to an AGV but has a lifting function, it is therefore an active vehicle [4]. ALVs are not considered in this report though.
8. Automated stacking cranes (ASCs) at the stack:
   Their function is to transfer the container from AGV to the stack or vice versa.

AGV navigation

The AGV system requires a topology and a positioning system. The topology consists of paths on which the AGVs travel. On a traditional routed terminal, the topology can be constructed out of different techniques [2], [5]: current-carrying wire, magnets or electric transponders. Wires (overhead or in the ground) were a popular technique in the early days of AGVs but since the late 80s, transponders and magnets are the new standard [5].
Different topologies are possible, for example mesh [6]–[8], grid-like [9], loop and tandem [10] (see Chapter 3). The paths in the topology can on its turn be unidirectional, where vehicles can only travel in one direction, or bidirectional, where vehicles can travel in both directions.

The positioning system makes sure that the controller knows the exact locations of the AGVs. A magnet ruler is mounted underneath each AGV and every time an AGV travels over a magnet, its position is recalibrated [5]. The transponder has a similar principle but uses an antenna instead of a magnet ruler. Free-ranging AGVs need more intelligence and sensors onboard. There are two types of free-ranging methods: with and without a planner.

2.2 Terminal – logistic system

The logistic processes can be categorized in three aspects: ships (order flow), containers (product flow) and equipment (resource flow). A PROPER model, introduced by The Delft System Approach [11], visualizes these aspects and their interrelations. Figure 4 shows the PROPER model for the assumed system.

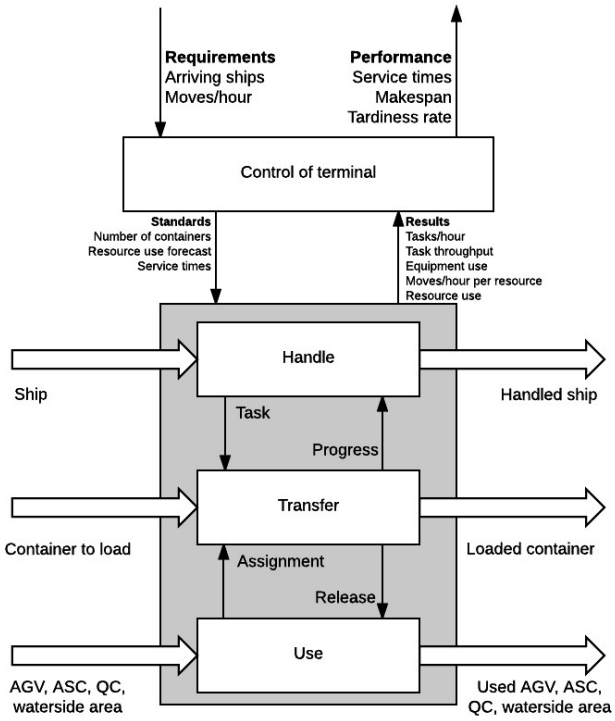


Figure 4 - PROPER model of the container terminal system.

The control system receives requirements from the environment that must be fulfilled. The output of the control system is the system its performance. The grey block is the operational system. Standards are send from the controller to the operational system and after operations, results are send back to the controller.

The operational system contains the three aspects:

1. The order flow. An order should be handled in the system. In the terminal system, a ship that must be (un)loaded with containers is the order. After completion, the ship is handled.
2. The product flow. In a terminal, the product is a container that must be (un)loaded into (from) a ship. There are two ways to fulfill the transfer functions: either transferring a container from stack to quay (export process) or vice versa (import process).
   Based on the Delft Systems Approach a process model is made from the import and export process in the system, Figure 5 and Figure 6 respectively. The rectangles represent the functions, the blue lines the container flow and the black lines the equipment flow. The white triangles represent waiting and the grey triangles represents being idle.
3. The resource flow. Resources in the system are the equipment: QCs, AGVs and ASCs, but also the waterside area of the terminal.

The interrelations between the aspects are:

- Task. When an order has arrived, tasks are made. These tasks concern the containers that must be handled. It can be quantified in service time per task.
- Release. Not all tasks can be released to the system at once, it must be dosed and the controller must make sure that the needed equipment is ready once a task pops up. It can be quantified by numbers of tasks per hour or task throughput per hour.
- Assignment. An ASC, QC and AGV are required to fulfill a task. Each type of equipment must be assigned to the task. It can be quantified by moves per hour per equipment type, idle rate per hour per equipment type or working rate per hour per equipment type.
- Progress. The progress of the order can be traced by updating the number of handled containers each time a task is completed. Quantify with number of handled containers per total number of containers.

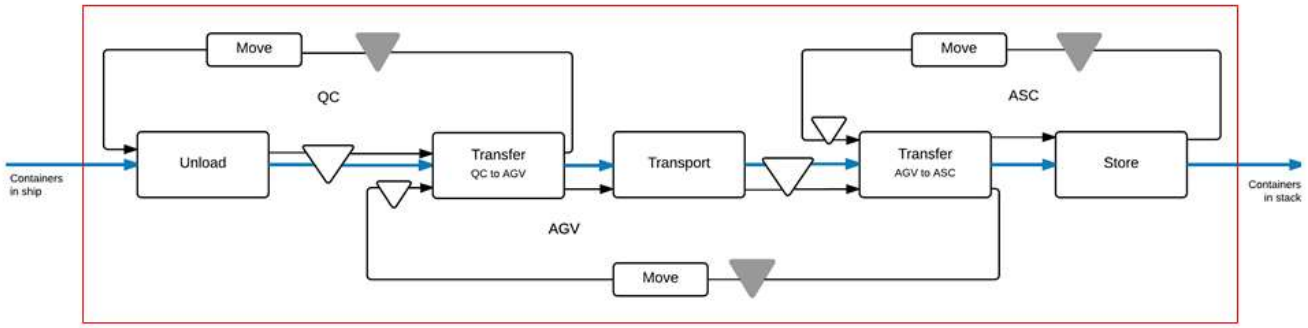


Figure 5 - Import process of containers.

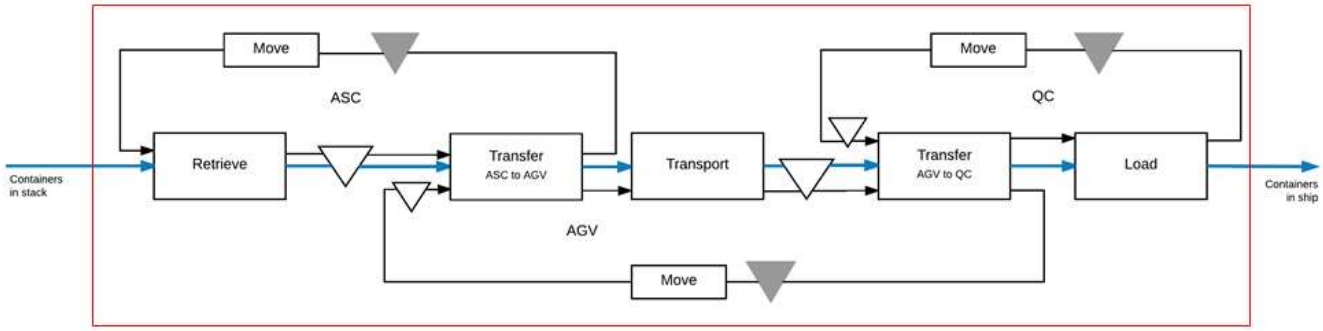


Figure 6 - Export process of containers.

Logistical processes at the terminal

The last important aspects of a container terminal are the logistical processes. These are needed to continue the operation of the terminal. There are three main processes:

1. Scheduling of jobs
   Its function is to schedule the incoming jobs (containers that must be transported) according to a load or stack plan. The desired arrival time per container, or container sequence, must thus be taken carefully into account.

2. Dispatching of jobs to AGVs
   Once the container schedule is made, equipment must be assigned that fulfill the jobs. QC and ASC is decided by the container origin or destination thus cannot be dispatched, so attention must be paid to the AGVs. Different strategies can be used for dispatching; one example is to choose the nearest AGV to the job its origin.
   A parallel strategy can be to operate the AGVs as single cycling or dual cycling. The former means that an AGV picks up a container at stack or quay crane, delivers it to quay or stack crane and then travels back without a container to its buffer place. Dual cycling entails that after a container is delivered, a new container can be loaded onto the AGV, thus the number of empty travels is reduced.
3. Routing of AGVs
   Routing concerns the actual route that an AGV should take in order to fulfill the job in a feasible way [12]. Depending on the topology, one or more routes can be taken to reach the destination. The terminal operator must determine the best route per AGV, taking into account other AGV path claims, arrival times at the cranes, objective of the transportation task etc. Routing is inherent to traffic control.

Planning levels

Logistical processes are decided for on three different planning levels [10], [13]: strategic, tactical and operational (Figure 7).

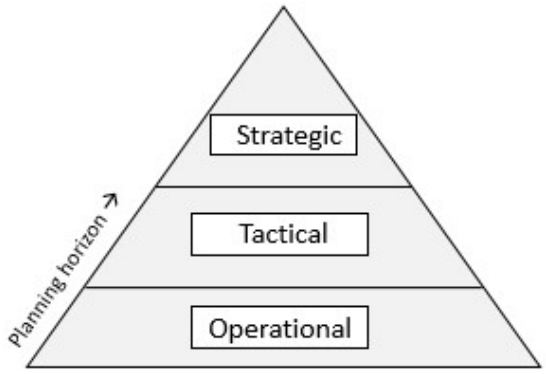


Figure 7 - Three level planning model.

Strategic level decisions are those for the long term (months to years): terminal topology, type of equipment, amount of equipment and the strategy of operations (scheduling, dispatching, routing). Tactical decisions are the numbers of equipment to assign based on forecast arrivals, positioning of the idle places, scheduling of jobs, strategy of operations and maintenance decisions. These decisions are based on mid-term planning horizon; days to months.
The operational decisions are routing per job, deadlock resolutions, congestion handling and crane assignment. This level operates at short term, day-to-day basis.

Traffic control is involved in all three levels. The topology on strategic level must be chosen simple with few intersections to ensure reliable traffic control. The tactical level must make sure that there are just enough vehicles and that the routes these vehicles take are such that no unnecessary conflict situations occur. On operational level, each AGV must chose a suitable path per job.

Traffic control

Traffic control is needed to avoid conflicts, deadlocks and collisions with obstacles. There are multiple ways to implement traffic control into the system, the next chapter discusses literature related to this subject. In general, control can roughly be divided into two types: central and decentral. Table 1 shows the differences between both [14].

Table 1 - Central versus decentral control.

	Architecture	**Advantages**	**Disadvantages**
Central control	One central agent to plan all trajectories.	• Easy to implement • Can reach optimization.	• Computational burden. • Entire system down if controller fails.
Decentral control	AGV controllers can communicate amongst each other, may or may not be supported by a central agent.	• Fault-tolerant. • Scalable.	• Not optimal solution of system.

Decentral control can be further divided into three types, shown as the three right types in Figure 8. The left type represents a central controller.

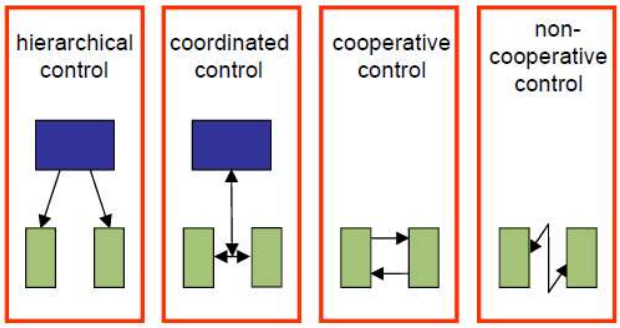


Figure 8 - Control architectures [15].

The second type, coordinated control, is the architecture implemented for fixed path, dynamic routing, for example on the ECT Delta terminal [15]. Within this architecture, a central controller is used, but AGVs can also communicate amongst each other and take simple decisions. Optimization and conflict resolution is done by the central controller.

2.3 KPI's of the system

At the waterside area of a container terminal, three types of equipment are in operation: AGVs, QCs and ASCs. Terminal performance relies on the one hand on their characteristics, but on the other hand on the strategic use of the equipment.

The following characteristics per equipment are relevant for the model in this report:

- AGV: speed, acceleration, deceleration, size, container capacity.
- ASC: cycle time (container retrieval time and AGV loading time), capacity per lift.
- QC: cycle time (AGV unloading time and put away time), capacity of spreader.

To assess the routing and terminal performance of a terminal and compare routing types later on, KPIs are necessary. Common KPI's in literature concerning automated container terminals are:

- Ship makespan. The total time needed to unload and load the ship (also called turnaround time).
- Waiting, idle or travel time or rate of equipment. The quay cranes must never be idle, else costly time is lost.
- Throughput of equipment. Tasks performed per hour.
- Distance or travel time per job. This should be minimized in order to deliver containers as fast as possible. Might not be optimal for entire system though.
- Average tardiness of jobs. Difference between actual task completion times and minimal time required.
- Operational costs The costs of equipment, combination of fixed and variable costs.

The aim of this research is to study the terminal and routing performance. Exact KPIs are chosen once the terminal model is developed, see Chapter 4.

ECT Delta case

To provide some feeling for characteristics and amount of equipment, the ECT Delta terminal is chosen. The ECT Delta terminal has a quay length of 3.6 km [16]. ECT Delta is the largest terminal of the ECT group, 263 ha of in total 476.1 ha [17]. ECT had a throughput of more than 7 million TEU in 2010.

Characteristics of available equipment [16], [18], [19] at this terminal:

Table 2 - Equipment characteristics of the ECT Delta terminal.

	AGV	ASC	QC
Number of equipment [-]	265	137	36
Max. speed (in corner) [m/s]	7 (3)	-	-
Acceleration [m/ s^2]	0.5	-	-
Size [m]	L x w: 14.7 x 3.1	-	-
Capacity per lift or ride [inch container]	20', 2x 20', 30', 40' or 45'	20', 30', 40' or 45'	20', 2x 20', 30', 40' or 45'
Cycle time [s]	-	198	160

The KPI's mentioned before are impossible to track down and could not be calculated based on the known figures as there must be made too much assumptions. Simulation models in literature mention their KPIs but these models are not specifically for the ECT Delta terminal.
The throughput of the ASCs and QCs of the ECT Delta terminal could be found in literature. The ASC cycle time is between 105 and 495 seconds with an average of 198 seconds [18]. A study by H. Schim van der Loeff [16] showed that the actual cycle time of a quay crane, 160 seconds, is 4 times lower than what is implied by the technical characteristics, this study is performed in 2007.

2.4 Conclusions

This chapter analyzed the system central to this report. The assumed system is the waterside area of a container terminal, the quay- and stacking cranes are on the boundaries. Within this system both physical and logistical processes take place.

Equipment found within this system are AGVs, QCs and ASCs. AGVs form a closed system while the QCs and ASCs are an open system in which containers enter or leave.
Each equipment type has its own sequence of physical actions. The AGV transports containers from quay to stack and vice versa. The QC and ASC have similar processes: containers are transferred between AGV and vessel or stack, respectively.

Containers can be imported or exported. Imported containers are unloaded from the vessel, transported to an ASC and then placed in the stack. Exported containers are retrieved from the stack, transported to a QC and then placed in the vessel.
Logistic processes in the system are scheduling, dispatching and routing. The first process schedules jobs, i.e. containers. Dispatching assigns equipment to each job, in particular AGVs that transport the containers. Routing concerns the actual route the AGV takes to reach its destination. Each of the logistical processes can be planned on a strategical, tactical and operational level.

Common KPIs to represent a similar system are ship makespan (turnaround time), throughputs, tardiness of AGVs and/or operational costs.

Chapter 3 - Review of related literature

The operation of AGVs require a threefold of logistical aspects: scheduling, dispatching and routing. They were briefly explained in the previous chapter. This report is focused on different AGV routing methods on an automated container terminal. Note that scheduling and dispatching are thus out of scope for this (literature) research.
This chapter discusses related literature to the subject of routing. Literature is sought in different databases, such as Scopus, WorldCat, Science Direct and the repository of TU Delft. Papers studying AGVs on automated container terminals are preferred, but other environments and fields are also explored. Think of AGVs in manufacturing environments and mobile robots (the term for AGVs in robotics). Articles are scanned on relevancy and quality.

Routing concerns the topology of paths and the assignment of paths (called routes) to transportation jobs. Routing is inherent to traffic control that ensures that the AGVs do not collide or end up in a conflict or deadlock. Deadlock is defined as "A situation where one or more concurrent processes in a system are blocked forever because the requests for resources by the processes can never be satisfied." [20].
Both the topology and assignment of routes can be done in a fixed or dynamic way. There are thus four aggregations:

1. Fixed paths and fixed routes.
   This type is also called static routing [13]. The topology of the terminal consists of a predetermined set of paths on which the AGVs can travel. The route to take from a start to end position is predetermined and only this route can be taken to reach the destination. This type of operation requires that all transportation jobs are known in advance such that each job can be scheduled. The first running AGVs at the ECT Delta terminal in Rotterdam used this type of routing.
2. Fixed paths and dynamic routes.
   Dynamic routes introduce flexibility to routing. The terminal its topology still consists of predetermined paths. For each job, a feasible route is chosen, there are thus multiple route options for a start to end point. During travel, the route may be changed. It is not required to know all jobs in advance and incremental planning is also an option [21]. An application fixed path, dynamic routing is found at the CTA terminal in Hamburg.
3. Dynamic paths and fixed routes.
   Dynamic paths means that there are no predetermined paths, the entire terminal area can be used to travel on. This combination is therefore not an option.
4. Dynamic paths and dynamic routes.
   This type is also called free-range routing. The entire terminal area can be used to travel on and there are an infinite amount of routes from a start to end point. Scheduling jobs in advance is possible, but is not required. There are currently no free-ranging AGVs in operation on container terminals.

This chapter discusses the traditional routing types, types 1 and 2, in section 1. Free-range routing, type 4, is discussed in section 2.

3.1 Routing on a traditional container terminal

Traditional terminals have a topology of paths on which the AGVs must travel. This topology is designed with traffic control in mind, the most reliable topology is chosen.

Routing is closely related to scheduling. Many algorithms are developed that try to optimize the transportation of jobs. The basis of these algorithms is often the vehicle routing problem (VRP), a problem in which multiple customers must be served exactly once. The vehicles that serve the customers, in this case AGVs, depart from a depot. To obtain a feasible schedule, this type of problem is extended with time windows: the vehicle routing problem with time windows (VRPTW). Each customer has a time window in which it must be served, the scheduler must assure this. Other problems are the pick-up and delivery problem with time windows (PDPTW) or dial-a-ride problem [22].
Basis to these problems are (extensions of) the Dijkstra algorithm that finds the shortest-paths in a network of nodes and arcs. Other routing objectives than shortest paths are also possible, think of fastest path or minimization of energy consumption.
For more information on routing algorithms of AGVs in terminals, the reader is referred to reviews [12], [13], [22], [23].

Traffic control is not complete when a topology is determined, other methods must assist the controller to ensure safe operation. The most well-known techniques are zones, time windows and semaphores. All are discussed briefly in this section. For summarizing reviews that provide an overview of traffic control studies, the reader is referred to [2], [3], [6], [9], [11].

Topology

Unidirectional lanes are often used to simplify the traffic control. Multiple lanes are placed parallel, in which half of them allow travel in one direction and the other half allow travel into the other direction. It is recommended to place the lanes facing the same direction next to each other in order to minimize braking [26]. Bidirectional paths are possible, but require more complex traffic control. A common conflict is that of two opposing AGVs wanting to travel on the same arc. Alternating and communication between AGVs are options to resolve such a conflict, the latter is most efficient [27].

Loop topology is the simplest topology, but AGVs travel lots of unnecessary distance. Tandem is a combination of loops, especially used in manufacturing environments [10]. Mesh, also called grid, is the most direct topology to reach a destination, but due to the many intersections a complex system. Another topology is a mix of loop and cross paths, in this report denotes as grid-like (Figure 10). This reduces complexity but has more flexibility than the loop topology. This type of topology is used at CTA Hamburg.

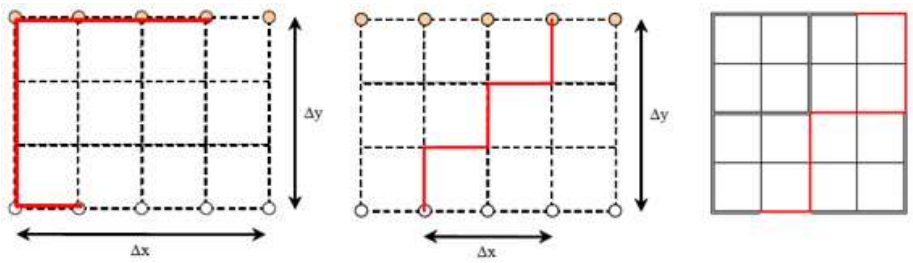


Figure 9 - Terminal topologies. Left: Loop [7]*. Center: Mesh* [7]*. Right: Tandem.*

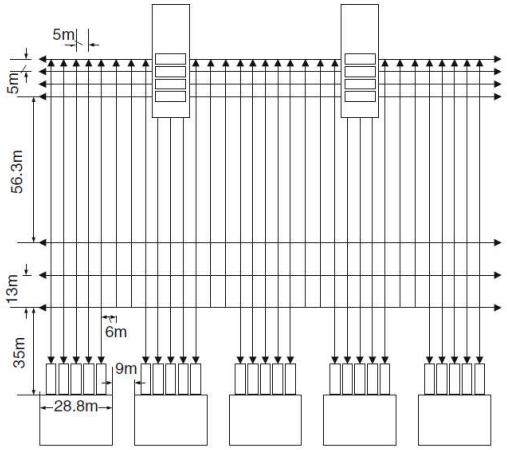


Figure 10 – Example of a grid-like topology [9].

Zones

Paths can be divided into zones. Claiming of a zone can be done by the use of time windows or semaphores, both are explained next. Synonyms for zones are occupation areas [8], [9] or cells [28]. There are two types of zones: static or dynamic [10]. The former type is where a path is divided into zones that remain fixed, this is the most common type. Dynamic zoning is not fixed and zone sizes change depending on traffic flow.

The size of the zones can be varied; either smaller than the AGV [8] or the AGV plus safety zone is one zone [2], [29]. Rectangles and circles are both options. The size of a zone is a trade-off between space utilization and computational power [28]. In general; the larger the zone, the less computational power required. However, especially in case of circles, precious space is unnecessary claimed. This is shown in the left figure of Figure 11. In general, the amount of zones in a cycle should be higher than the amount of AGVs in the cycle to prevent a deadlock [20].

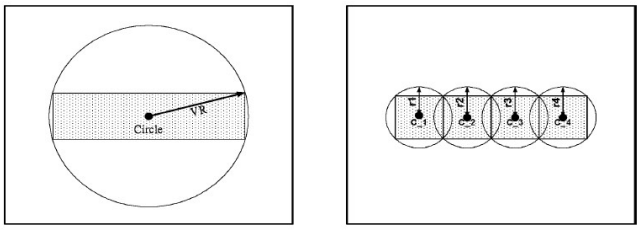


Figure 11 - Circular zone sizes [28].

If the next zone in which the AGV wants to travel is not free, the AGV waits in its current zone.
To prevent the AGV system to result in a deadlock, the following rules must be obeyed [8]:
1. A zone may be reserved by only one AGV simultaneously.
2. An AGV that is occupying zones, must try to reserve another zone.
3. An already reserved zone in a specific time window cannot be reserved by another AGV.

Time windows

Time windows are used in combination with zones, a zone is reserved for a specific time window. The size of this window is the zone traversal time plus a safety time, often 1-5% of the traversal time [30]. If a congestion or other conflict prevents the AGV from moving to its next zone, the time window reservation of the current zone is elongated. Time windows are regularly checked and updated.

Semaphores

Semaphores are a technique to control the amount of AGVs in an area [2], [31], [32]. The capacity of a semaphore can be binary or a positive integer [32]. Semaphores are controlled by the "wait and proceed" logic [31]. The "wait" procedure is activated when there are as many AGVs in the area as the semaphore capacity. No AGV may enter the area. The "proceed" procedure is activated when the capacity of the area is not reached, thus an AGV may enter the zone. The semaphore its temporary capacity is updated each time an event takes place, i.e. an AGV either leaves or enters the area.

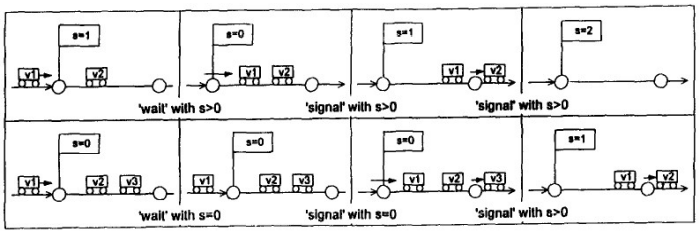


Figure 12 - The use of semaphores [2].

Areas can be controlled by multiple semaphores, but no more than one semaphore can control an arc.

3.1.1 Choice between fixed or dynamic routes

A choice must be made for the type of routing (fixed topology, fixed routes or fixed topology, dynamic routes) to implement as traditional routing in this report its model. The advantages and disadvantages per routing type are showed in Table 3 [13].

Table 3 - Advantages and disadvantages of static versus dynamic routing.

	Advantages	**Disadvantages**
Fixed topology, fixed routes	• Easy to implement • Less computational power	• Not flexible • Chances of congestions and deadlocks are high • Not robust • Load plan must be known in advance
Fixed topology, dynamic routes	• Flexible: routes can be changed on-the-go • More routes • Shortest routes can be chosen if free • Scheduling not necessary	• More complex system • Higher computational power • Not always optimal path per job (in case of incremental planning)

In terms of routing, the static version is most simple and computational efficient. There are no variations in routes, only traffic control must ensure that no collisions occur. The robustness of this type is bad, if an AGV breaks down on an arc other AGVs that use that arc cannot continue.
The second type is flexible, robust and the shortest path can be chosen per job (not guaranteed with incremental planning). Routing is however more complex, thus the computational burden will be higher.

The cycle or travel times of equipment on a container terminal exhibit stochastic behavior. It is nearly impossible to plan the entire container transportation sequence in advance; small disturbances easily jeopardize the minimization of the make span. Although static routing is easy, flexibility is a very valuable characteristic in a dynamic environment like a container terminal.
Fixed topology, dynamic routing is preferred over static routing in an automated container terminal. This type is also most applied in terminals and chosen for this research.

Overview of papers

For all the discussed papers, excluding reviews, a table is made to provide an overview (Table 25). The traffic control aspects, solution methods and objectives are classified.
It can be seen that all studies are focused on the minimization per individual trajectory instead of minimizing the make span. All papers explain their traffic control and most often use zones in combination with semaphores or time windows. A simulation to is used test their algorithms. AGV numbers up to 200 are tested, which is close to real life terminals [33].

3.2 Free-ranging

Fixed topology, dynamic routes routing is the current way of transport operation in automated container terminals. Although there is some flexibility in the system, future growth will require more capacity of the AGV system. A centralized system has limitations on its scalability and thus new routing methods must be explored. A similar field in robotics, those of mobile robots (other term for a free-ranging AGV), might lead to new opportunities for AGV routing in container terminals.
With free-ranging is meant; the AGV is no longer constraint to drive on fixed paths, the entire area may be used for transportation. A conflict-free trajectory is determined per job and the AGV is safely navigated to its destination. The different methods of free-ranging are discussed next, first the general advantages of free-ranging routing compared to fixed path, dynamic routing are stated:

- Direct trajectories to destination.
  Jobs can be completed faster which can lead to a minimization of makespan. This is an important KPI, literature is shifting towards optimization of the integration of terminal operations.
- Flexibility.
  The entire transportation area can be used, thus there are infinite options to reach a destination. Also in case of rerouting there are plenty of options.
- Wear and maintenance.
  Because trajectories can be shorter, the AGV wears less fast. This also benefits the maintenance costs.
- Number of AGVs needed.
  A reduction of AGVs might be possible, which benefits the investment and operational costs.

However, there are also some drawbacks. Think of:

- Completeness.
  Not all free-ranging techniques are guaranteed to find a solution (a route to the destination), this is indicated with the term completeness.
- Time complexity.
  The controller is more complex. For each new job a feasible trajectory must be chosen, thereby avoiding other AGVs. Complexity increases computational burden. Time complexity differs per chosen method.

No free-ranging AGVs are in operation on a real-life container terminal, literature is slowly exploring the opportunities in this field. Mobile robots are studies quite extensively since some decades, many control approaches are used and reviewed [34]–[37].

3.2.1 Motion planning methods

To travel to a destination, the AGV needs a trajectory planner and a controller. The planner decides which route the AGV should take in order to reach its destination in a safe and fast way. The controller tracks and corrects the AGV to make sure it travels on the planned trajectory.
There are three types of planners depending on the time horizon of the planner and the estimated time of arrival (ETA):

- Complete route planning: time horizon ≥ ETA
  The start and end positions of the AGV are known. The planner takes into account other AGVs their trajectories and based on that information plans a feasible route. The advantages of complete route planning is that the route is optimal at the moment of departure and the destination is guaranteed to be reached, no deadlocks occur. Drawbacks are the stochasticity of the system (the route may become invalid due to disturbances [21]), the inadequacy to respond to unknown events [35] and the computational inefficiency [38]. This is a global approach.
- Incremental planning: time horizon < ETA
  A small part of the route is planned per iteration, depending on the time horizon or segment distance of the method. When the AGV reached its temporary destination, the new route part is determined. Advantages of incremental planning are that the response time of the algorithm is shorter and less failures of the algorithm occur [21]. Disadvantage is that the overall optimal trajectory is not taken into account.
- No route planning: time horizon < ETA
  This type is called a reactive approach. The direction of motion towards the destination is determined and the AGV starts heading in that direction. Each iteration, the vicinity of the AGV is checked for (moving) obstacles and direction of the AGV may be changed in order to avoid collisions. This type of planner cannot guarantee that deadlocks will not occur. The main advantage of this method is its computational efficient approach [39]. The main disadvantage is the risk of deadlocks. This is a local approach.

Traffic control is inherent to a free-ranging planner and controller. A feasible trajectory is one that does not collide with obstacles and avoids other conflicts. Traffic control is dealt with depending on the free-ranging method used.

In general, free-ranging methods are divided into classic methods and intelligent methods [34]. Classic methods are cell decomposition, roadmap, potential field and virtual force. Intelligent methods are often based on behaviors found in biology. Another emerging technique is Model Predictive Control (MPC), in which each AGV is seen as an agent that can take decisions thereby having the option to consult other agents in the system.

Table 4 - Planning types of free-ranging methods.

	Complete route planning	Incremental planning	No route planning, only controller
Cell decomposition	x	x	
Roadmap	x	x	
Potential field			x
Virtual force			x
DEFT	x		
ACO			x
MPC		x[1]	

[1]MPC plans the complete route to destination, but this route is updated every time step

Classic methods

Cell decomposition

In this method, the transport area is divided into non-overlapping cells. A route is planned following a sequence of adjacent cells, the route can be planned entirely or incrementally. A cell can be larger or smaller than the AGV. Dijkstra and the A* algorithm are commonly used algorithms to find the shortest path. Algorithms can search forwards, backwards or both (they meet in the middle). These algorithms can be adapted to include the use of time windows and/or the claiming of multiple nodes at once [30], [40].

Cell decomposition can be divided into exact decomposition, approximate decomposition and adaptive decomposition [34]. Exact decomposition uses the obstacles as starting point for decomposition. Multiple methods exists within exact decomposition, for example trapezoidal decomposition [41]. In this method, a vertical or horizontal line is drawn from each node of the obstacle to the area boundary, see Figure 13.

Using approximate decomposition, cells have a predefined size and shape. The area is made up of a grid of cells, this grid is laid virtually over obstacles. A cell can be partly or entirely occupied by an obstacle or the cell is free. Cells of the latter case can be claimed by the path planner. Depending on the neighborhood geometry, a cell has 4,8 or 16 reachable neighbor cells [28].

Adaptive decomposition also makes use of cells with a predefined size and shape. Partly occupied cells are however split into four cells such that the free parts of the original cell can now be used to travel on. The splitting into four cells continues until there are no partly occupied cells in the grid. See Figure 13.

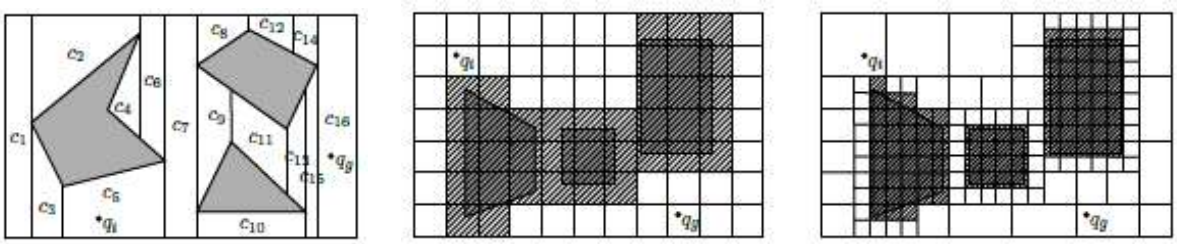


Figure 13 - Cell decomposition methods [41]. *Left: Exact cell decomposition. Middle: Approximate cell decomposition. Right: Adaptive cell decomposition.*

Roadmap

This method constructs a map with paths between free areas on which the AGV can travel [34]. The path planner uses these paths to calculate the fastest route for the AGV. This can be done in a complete route or incremental way. Routing of this method depends on the (modified) path planning algorithm that is chosen.

There are many methods to construct the paths, two examples that are briefly explained are the visibility graph and Voronoi diagram [34], [41]. In the visibility graph, straight lines are drawn between each node in the area that do not cross an obstacle. Obstacles are often somewhat enlarged to prevent the AGV from grazing against the obstacle.

A Voronoi diagram is constructed by drawing a line between two nodes, the line is at the exact middle of two nodes. Each closed, surrounded node is a cell. See Figure 14.

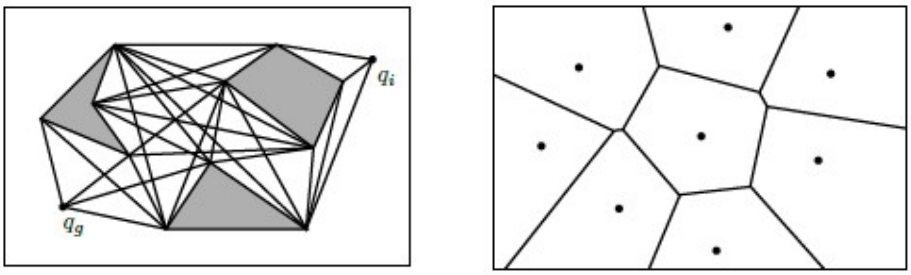


Figure 14 - Roadmap construction methods [41]. *Left: Visibility graph. Right: Voronoi diagram.*

Potential field

In the potential field method, AGVs are positively charged and move towards negatively charged destinations, while they repel from the positively charged obstacles [14]. It thus has no planner. The total potential field is made from superposition of all individual fields: a uniform flow field, a field where the goal is represented as a sink and a field where each obstacle is represented as a source. Figure 15 shows an example of the construction of the total potential field. A drawback of the potential field method is that it can suffer from local minima in which the AGV gets stuck. This can be avoided by using a harmonic field in which no local minima occur [42] or an escape force [41]. Potential field method is a popular control technique in robotics as it yields good results and is computationally efficient [35].

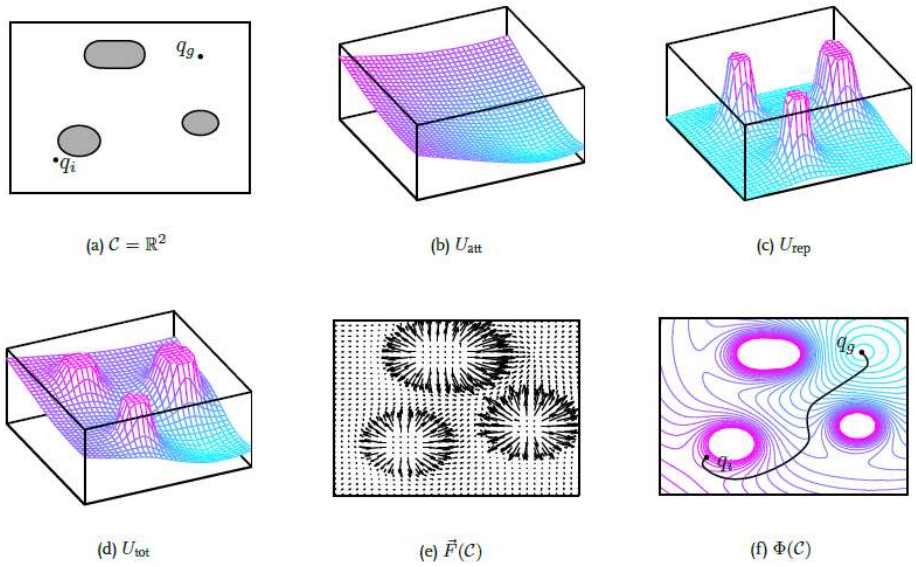


Figure 15 - Construction of total potential field [41]. *a) obstacles in the field, b) attractive potential to target, c) repulsive potential of obstacles, d) total potential field of b and c, e) vectors representing the total potential field of d, f) contours of the total potential field and an example of a path.*

The construction of a virtual force is a common implementation of the potential field method. A case study based on a container terminal using AGVs is presented by Choi and Zhu [14], [43]. The construction of the virtual force is shown in Figure 16. The results were satisfactory, but they emphasize that the control parameters are determined empirically and could benefit from an analytical model.

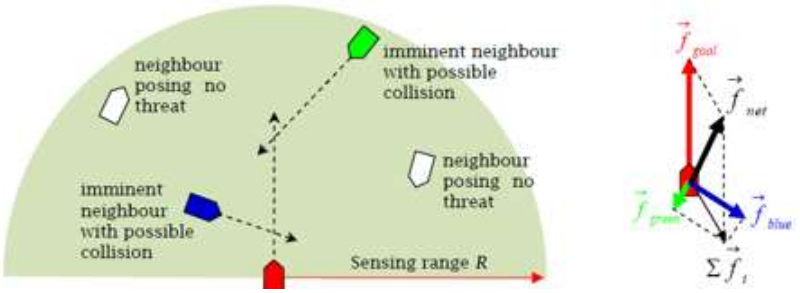


Figure 16 – Virtual force construction [14]. *Left: Detection of obstacles. Right: Related virtual force model.*

Virtual force

Originally, the virtual force method is based on both a certainty grid and the potential field method. The area is divided into cells and the robot scans its neighborhood. If an obstacle is detected, the certainty of the cell on which the obstacle is detected is increased by one. This results in a grid of certainty numbers, the higher the number, the more certain it is that there is an obstacle. All directions are translated into forces corresponding to the certainty of the cells. Those are combined into a resultant repulsive force and the resultant of this force and the attractive force is the direction in which the AGV will travel.

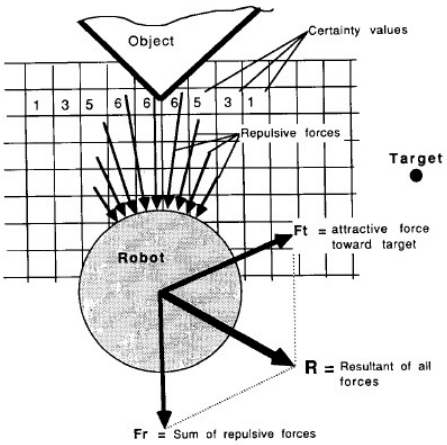


Figure 17 - Virtual force method [44].

This methods suffers from some drawbacks. Lots of repulsive data is simplified into one force vector and thus detailed information on nearby obstacles is lost. Also a unstable, oscillation motion occurs in narrow corridors. A new, improved method based on virtual force is developed; the vector field histogram method [45]. The area is divided into a polar histogram. Based on this histogram, the (safe) direction closest to the target is chosen.

It should be mentioned that over the years the virtual force method got rid of the certainty grid and became part of the potential field implementation methods that use virtual force construction.

Intelligent methods

Many motion planning methods in robotics are developed that are heuristics or approaches based on biology. Examples are social force method, ant colony optimization, particle swarm optimization, genetic algorithm, neural network and fuzzy logic [3], [35]. Intelligent methods dominate the field of robotics

since 2000, classical methods are less used since then [35]. The social force method and ant colony optimization method are briefly explained.

DEFT

Human and walking behavior is often studied and modelled in the field of robotics, DEFT is inspired by pedestrian behavior. The DEFT (dynamic, evasive, free-range trajectories) controller is an example in which an automated container terminal with AGVs is used as a case study [3], [7]. The DEFT controller consists of a planner and controller. The planner plans the complete route within the time frame of expected departure and arrival. The corresponding coordinates are claimed per time step. The planner minimizes a cost function, including costs for time pressure, obstacles, speed of movement and time window arrival [3]. Collisions are avoided using social force. The controller ensures that the trajectories are executed as planned, the controller is an optimal controller. Individual trajectories are optimized. A drawback of this controller is the high computational power it requires.

Ant colony optimization (ACO)

ACO is inspired on the ability of ants to find the shortest path. Ants deposit pheromones while walking, other ants follow the strongest scent of pheromones that represents the shortest path [46]. Adaptions are easily made when obstacles appear. The controller lets the ants follow the strongest pheromones, no route is planned beforehand. Hamzheei et al. [47] developed an ACO algorithm to operate in a block environment. The application of ACO in container terminals is questionable, with respect to new constructed paths and traffic control.

Model predictive control (MPC)

A new emerging technique in trajectory planning is model predictive control. The procedure is as follows [48]: every time step, the new states of the system are determined from the measurements that are provided to the controller. The expected disturbances per time step are updated and the control problem is solved to generate the optimal actions. These actions are returned to the system.
The main advantage of MPC is that every time step, the most recently known disturbances and states are considered. Each agent, i.e. AGV, has a MPC controller. MPC needs a planning algorithm that determines an optimal and feasible trajectory [36] for the time horizon ahead. It is a distributed control system. MPC requires a numerical solver to solve the control problem, the solver can either be of the type linear programming or non-linear programming [48].

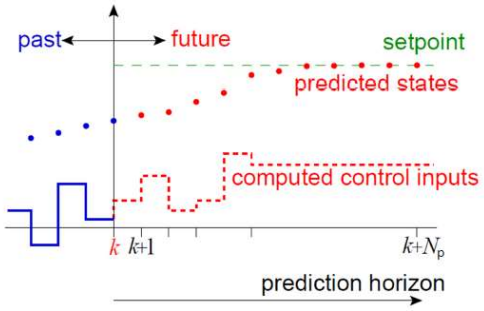


Figure 18 - Model predictive control [48].

An example of MPC implemented in a case of AGVs at an automated container terminal is the work of Xin et al. [49]–[52]. The control commands are translated to a mixed integer linear programming (MILP) problem and the solver solves this problem sequentially. Constraints are given to obstacles so that they

are included in the MILP problem. The controller is a hierarchical controller composed of multiple layers, both continuous and discrete dynamics are incorporated. The objective was to optimize the terminal operation, not individual trajectories. A drawback of this controller is the high computational power.

Overview of papers

Free-ranging studies are focused on collision avoidance rather than on deadlock avoidance (Table 25). Deadlocks will however be an issue when the AGV density increases. Free-ranging studies show a trend towards minimizing makespan. All studies of AGVs on automated container terminals use simulation as their solution method. Often, small AGV fleets are used which is not representative for real-life terminals.
Virtual force, social force and MPC are the only methods with a case on an automated container terminal.

3.2.2 Choice of method

In general, control methods in robotics are evaluated on the following three points [34]:

- Completeness
  A controller is complete when a solution exists and the controller guarantees to find this solution. With solution is meant: it is possible for the AGV to reach its destination. An incomplete controller does not guarantee to find a solution if one exists.
- Optimality
  Optimality guarantees that the lowest-cost path is found. A non-optimal controller cannot guarantee that the solutions are optimal. The controller can also be optimal only in some sense, when sometimes optimal paths are found [41].
- Time and algorithm complexity
  Time complexity concerns the calculation time of the controller to find a solution, this is depending on the complexity of the algorithm. Time complexity can be stated into a lower and upper bound [41], this is out of scope for this research however.

Other important criteria points that the controller must fulfill are:

- Flexibility
  The controller must be flexible in its routing. In case of a conflict, a new route should be quickly determined.
- Scalability
  It must be possible to add AGVs into the system such that future terminal growth is no problem.
- Safety
  The AGVs must operate in a safe manner. Collisions may not occur.
- Robustness
  The controller must be able to deal well with moving objects and uncertainty. The ability to keep operating if a controller stops working, an AGV breaks down etc.
- Time horizon
  A suitable time horizon must be chosen to ensure the correct planning type. In case of incremental planning, the time horizon must yield the correct balance between computational complexity and solution quality.

Each method is briefly reviewed on each of these criteria points, see Appendix B. The reviews are based on knowledge gained from literature, most often robotics related literature.

Evaluation table

Each method is evaluated on the criteria points (Table 5). Assumed is that the methods are properly designed such that the balance between evaluation points is optimal.
Roadmap and virtual forces are not shown as they are equal to cell decomposition or potential field, respectively.

The scores are plus if the evaluation point is positively fulfilled, minus if the evaluation point is negatively fulfilled.

Table 5 - Comparison of free-ranging methods on the criteria points.

		Complete	**Optimal**	**Time and algorithm complexity**	**Flexibility**	**Scalability**	**Safety**	**Robustness**
Complete route	Cell approximate decomposition	Yes	Yes	+	+	+	++	-
	DEFT	Yes	Yes	-	+	+	+	-
Incremental	Cell approximate decomposition	No	No	+	+	+	++	-
	MPC	No	No	-	++	++	+	+
No planner	PFM	No	No	++	++	++	+	+
	ACO	No	No	-	+	-	+	-

Each method has its own pros and cons, but ACO is the least suitable method of all. Newly constructed paths are not guaranteed to reach its destination and on a large scale, the computational burden is high [53].
The cell decomposition scores well on everything, except robustness. The method relies on a central controller that should respond quickly and certainly cannot fail.
The DEFT controller is more optimal as it plans an entire route at once. Safety is guaranteed with implementations of the social force model. Drawback is the high computational power it requires.
MPC scores well on all points except time and algorithm complexity, this method is computationally intensive because all actions should be determined and revised at each time step. A distributed system is assumed, thus scalability is a plus.

It is decided to choose the potential field method as free-ranging routing type because it has the benefit of being a little intensive, robust and flexible method. Its drawbacks, local minima and risk of deadlocks, are possible to improve without much effort needed. The algorithm is extended with escape forces that should take care of the drawbacks.

3.3 Guidance of free-ranging AGVs

The newly suggested routing type, mixed routing, comes forth from the drawbacks of free-ranging AGVs in crowded zones. Problems arising from multiple AGV robot navigation in crowded areas is mentioned and studied in the field of robotics [54]–[56]. It may therefore be expected to occur on container terminals.

In order to obtain smooth navigation between multiple AGVs, organized behavior is studied in the robotics field. The suggested mix of fixed-path and free-range routing is one solution to organize the AGV flow in crowded zones. Other solutions as suggested in literature are briefly explained next.

Bypassing crowds

Ferrera et al. [54] propose an algorithm that detects crowds and resolves conflicts. When a crowd is detected, the AGV is steered counterclockwise around the crowd and other AGVs are assumed to follow this behavior. The idea is clever but not applicable in this report because the destination of the AGVs lays inside the crowd. A distinction could be made for AGVs that are driving away from the crowd and ones that must enter the crowd, but this would require too much algorithm complexity.

Reserved region

Pallottino et al. [55] propose the use of a reserved region for each AGV. The reserved region is shaped as a circle and is as large as the turning radius of the AGV including safety distance. Some traffic rules are suggested that are applied to all AGVs, such as the fact that only right turn moves can be made. Their solution is tested for 70 AGVs and did not result in deadlocks or collisions. The applicability in this report is questioned due to the large radius of the reserved region.

Checkpoints

Bähr et al. [56] propose the use of checkpoints to organize AGV flows. These checkpoints must be visited by the AGVs and are placed such that an ingoing and outgoing flow is established. This prevents deadlocks. In the case of a deadlock, the AGV can be routed towards a checkpoint.

Semaphores

Bähr et al. [56] propose the application of a semaphore controlled zone around the destination, this reduces traffic in this zone and thus the risk of deadlocks and conflicts. A variation is to divide the destination into two zones with each a checkpoint to be visited and thus creating a route.

Self-organizing behavior

Other valuable insights are gained from human self-organizing behavior in crowded situations, often taken as inspiration in the robotics field. Three common self-organizing behaviors are: arching and clogging, the faster-is-slower effect and lane formation [57].

Arching and clogging

Arching and clogging is seen when pedestrians face a narrow opening. All pedestrians walk towards the opening, the density at that section increases, the pedestrians go one by one through the opening in a non-orderly fashion (Figure 19). This type of behavior will not be seen at a container terminal as it is a non-existent situation there.

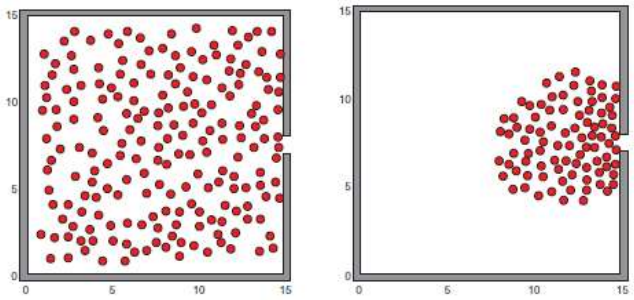


Figure 19 - Arching and clogging effect [57]*. Left: Initial crowd distribution. Right: Crowd during evacuation.*

Faster-is-slower effect

Pedestrians have a desired walking speed. In crowded situations, it is impossible to maintain this speed and the pedestrian must slow down. In case of an evacuation and pedestrians try to maintain their desired speed, the crowd can get clogged and as a result the evacuation time is slower than when people decreased their speed. AGV speed should be optimized such that the container handling times are minimized.

Lane formation

Lane formation is the effect that is seen when crowds intersect, see Figure 20. When the crowds first meet, small lanes are formed that are later aggregated into wider lanes. Lane formation reduces the friction between pedestrians [58]. This type of self-organizing behavior could be applied to AGVs at a container terminal but will perhaps be more beneficial in the middle of the transportation area than near the cranes. The AGVs near the cranes reach at some point their destination and thereby disturb the lane flow.

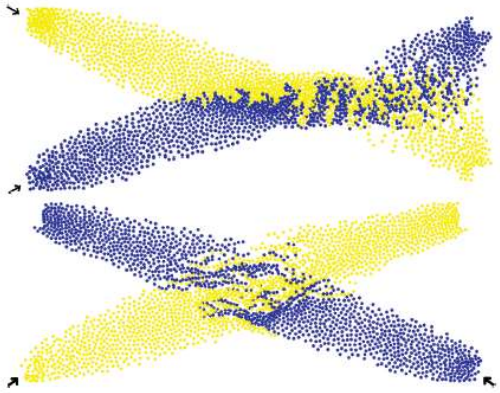


Figure 20 - Lane formation [58].

3.4 Conclusions

This chapter provides background and related work to the research question of this report. The research question required three subjects to study; routing on a traditional container terminal using AGVs, routing of free-ranging AGVs and guidance of multiple AGVs in crowded zones.

Traditional routed terminals have paths that are fixed, the routes to take can be fixed or dynamic. In the former case the exact same route is taken each time from A to B. If multiple routes can be chosen, it is called dynamic routing. The latter case is more flexible and robust.
The model of the traditional routed terminal will use fixed path, dynamic routing because of the benefits over static routing. The topology will look like a grid to offer route flexibility. Other assumptions of this routing type as implemented in this report is presented in Chapter 5.1.

Free-ranging is a method that uses dynamic paths and dynamic routes. The entire terminal area can be used to travel on and a feasible trajectory is decided for each job. A free-ranging AGV requires a planner and a controller. The planner can plan the entire route, part of the route or only a direction (no route). The controller checks the AGV its position during travel and corrects if needed. Global methods (ones that plan) have a centralized control architecture, local methods (ones that do not plan) have a decentralized control architecture.
There are several free-ranging methods. Often the distinction between classical and intelligent methods is made. Classical methods are cell decomposition, roadmap, virtual force and potential field. Intelligent methods are often inspired on biology: social force and ACO are briefly explained. A new emerging technique is that of MPC, in which each AGV is an agent that can negotiate with another agent. Different control architectures are possible.
The free-range methods are rated on criteria points: completeness, optimality, time and algorithm complexity, flexibility, scalability, safety and robustness. Concerning the development of a free-ranging controller, the method of potential field will be used. The PFM is proven to yield good results. It scores well on all the criteria points, however completeness of this methods requires some special attention. This is a local method, no planner will be implemented. Implementation of the PFM is presented in Chapter 5.2.

The PFM is sensitive to deadlocks which will occur in crowded zones. The problem of crowded zones is recognized and studied in the field of robotics, but not specifically mentioned for AGVs in container terminals. Some solutions in robotics are: checkpoints, semaphores or self-organizing behavior as seen with pedestrians. The suggested mix of fixed-path and free-range routing is not studied in literature, but it is expected that this guidance of AGVs in crowded zones benefits the performance.

Chapter 4 - Terminal model

The system in this report, a container terminal, is a complex dynamic system. A discrete-event simulation model of the terminal will be made to assess its performance. The terminal model will be developed in Borland Delphi including the TOMAS tool. TOMAS is a tool that tracks each event and provides useful statistical information on the elements.
The objective of the model is to evaluate its terminal and routing performance such that the best routing type can be determined on.

This chapter presents the terminal model with the general routing component. Detailed information on each routing type (traditional routing, free-range routing and mixed routing) is presented in Chapter 5.

4.1 Model description

The interacting system of QCs, ASCs and AGVs is modelled. The AGV system is a closed system, the cranes have an open system and their behavior is not modeled into full detail. The assumptions will make clear which details are included or excluded.

Assumptions

The following assumptions are made for this research and are partly based on the studied literature:

- Waterside area is considered as the system, in particular the area between quay cranes and stack.
- Only loading of containers into a vessel is assumed. To compare the performance of the three routing types it is not necessary to simulate both loading and unloading, therefore it is chosen to not implement unloading due to the complexity that comes along with it.
- Only AGVs are used to fulfill the transportation function. Research is focused on routing and traffic control of AGVs. The size of an AGV is 14.7 x 3.1 m [19]. The AGV itself is assumed as a point mass, no kinematic constraints are taken into account.
- Load plans, and thus scheduling, are out of scope for this report. At initialization, containers are randomly placed along the stacks. One dispatching strategy is chosen (explained later).
- Once a container is delivered at a QC, the AGVs heads towards a buffer place but may be reassigned to a new job.
- Exact locations of the containers in stack or vessel are out of scope. These are covered with stochastic cycle times of the quay and stacking cranes.
- Container (un)loading time on (off) an AGV is taken as a constant variable for the QC and ASC.
- AGV travel speed is taken as a constant, equal for all AGVs. No distinction is made between corner speed and “normal”, straight ahead speed. Acceleration and deceleration of the AGV are included.
- Each QC has four unloading positions.
- Each ASC has one loading position. Waiting of multiple AGVs at an ASC will not occur, because only the loading process is assumed.
- Idle AGVs are parked in the middle lanes of the topology, the buffer zone (see Figure 21).
- QC and ASC legs are not assumed, thus also not as obstacles. Other static obstacles in the terminal area are not assumed.
- No door correction moves are assumed.
- Breakdowns of equipment are not assumed.

Topology of the model

The ASCs are located at the bottom, each with one unloading spot (see Figure 21). The y-location of the ASCs is zero. The QCs are located at the quay, top of the figure. There are four unloading spots at each QC.

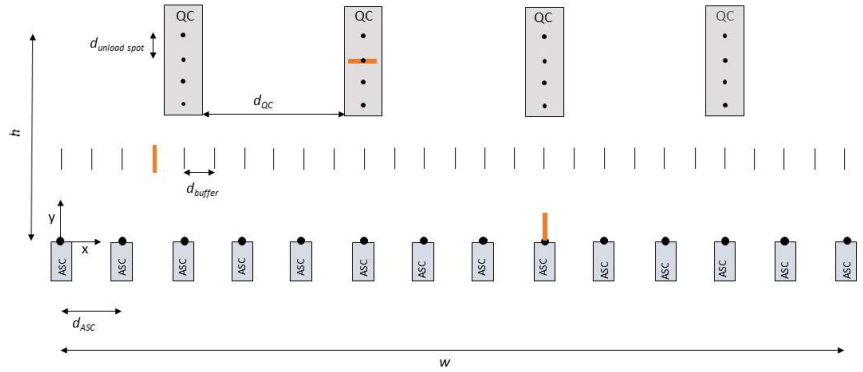


Figure 21 – Terminal model topology.

The AGVs are initialized in the middle lane at buffer locations (more detail on exact locations per routing type will be given in Chapter 5), from where they are called for a job.
The container generator places each container in its corresponding stack, containers leave the system when placed on the vessel. Stack and vessel are not shown in Figure 21.

PDL of equipment

The process cycle of each equipment type is briefly explained in Process Description Language (PDL).

ASC procedure
Repeat
  If number of containers in stack > 0
    Choose container
    Request AGV
    Retrieve container in stack (bring container to transfer point)
    Wait until AGV is at ASC
    Load container onto AGV
    Activate AGV

The ASC procedure starts with the fact that there are containers to handle. A container with QC destination that has the smallest handling queue is chosen. A container is placed in the handling queue once the ASC starts handling and leaves this queue once it is delivered to the QC.
An AGV is requested to transport the container. Then, the ASC cycles to the container in the stack, picks it up and brings the container to the transfer point (where the AGV waits). The container is loaded onto the AGV and the AGV may continue.
Two parameters are used to describe the stacking crane behavior: loading time and container retrieval time. The latter represents the time of the stacking crane to travel to the container, pick it up from the stack and bring it to the transfer point.

AGV procedure
Repeat
Wait for job to handle
ROUTING: *drive to ASC, explained in Chapter 5*
Wait until ASC is ready and AGV is activated
ROUTING: *drive to QC, explained in Chapter 5*
Wait until QC is ready and AGV is activated
ROUTING: *drive to buffer while no container is assigned to AGV, explained in Chapter 5*

Once a container is assigned to an AGV, i.e. the AGV is dispatched, the AGV drives to the ASC. After the container is loaded onto the AGV, the AGVs gets a sign from the ASC that it may continue. The AGV drives towards the QC, where the container is unloaded. Once the container is unloaded, the AGV drives to the buffer but may be reassigned to a new container and thus it will travel to an ASC.

Parameters belonging to the AGV process are speed, acceleration, deceleration, unloading time and load time.

QC procedure
Repeat
Request a task (AGV with container at an unloading point to handle)
Wait for AGV with container to be assigned
Unload container from AGV
Activate AGV
Put away container in vessel

The QC places a request that it is available to handle a container. It remains idle until an AGV has arrived at one of its unloading spots. Once the container is unloaded, the AGV may continue and the QC places the container in the vessel. In case of a queue at the QC, the AGVs (and thus containers) are handled FCFS.

Two parameters are needed to describe the quay crane behavior: unloading time and put away time. The latter also includes the spreader of the quay crane travelling back to the transfer point.

Equipment interaction

Interaction between the elements is necessary to fulfill a container cycle (Figure 22). Each element (i.e. equipment) procedure was explained just before, focus is now on the interaction moments.

Once a container is chosen at the ASC, a request goes out to assign an AGV to this container. This request is handled by the RequestAGV procedure (Appendix C). This procedure dispatches AGVs and activates the chosen AGV.
The ASC can only load the container on an AGV when it has arrived, this is checked with the location of the AGV. After loading, the AGV is reactivated by the ASC.
Once a QC is created (at initialization) or done handling a container, it places a request to handle a new container. These requests are handled by the RequestTask procedure (Appendix C), the QC is reactivated once a request is granted. The AGV is reactivated by the QC once the QC finished unloading.

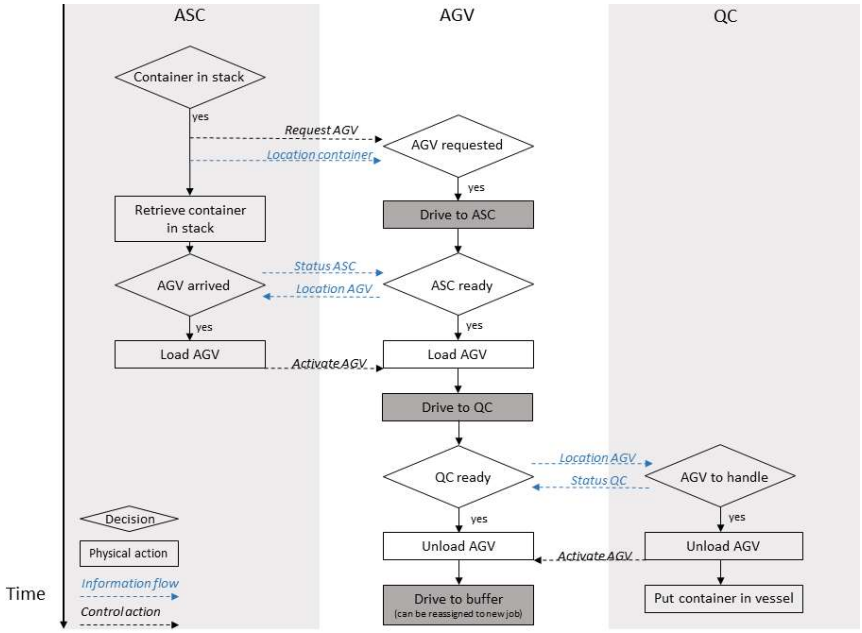


Figure 22 - Element interactions of the terminal model.

Routing functions are represented with the grey blocks. Each routing type has a different implementation, as will be explained in Chapter 5. The rest of the diagram remains intact.

More detail on the control of the model and its interactions is given in Chapter 4.2.

Input of the model

The exact container location in the stack and vessel are not included and are imitated with stochastic behavior of the cycle times of the quay and stacking cranes. Stochastic behavior is modeled as a normal or empirical distribution, QC or ASC respectively. The model requires the following input variables:

Table 6 - Input parameters of terminal model.

	Parameter	Type of value	Unit
Topology	Size of terminal (w x h)	Constant	m
	d_{ASC}	Constant	m
	d_{QC}	Constant	m
	d_{buffer}	Constant	m
	$d_{unload\ spot}$	Constant	m
Operational	Number of AGVs	To be varied	-
	Speed of AGV	Constant	m/s
	Acceleration of AGV	Constant	m/s^2
	Deceleration of AGV	Constant	m/s^2
	Unload time of AGV at QC	Constant	s
	Put away time of QC	Normal distribution	s
	Load time of AGV at ASC	Constant	s
	Container retrieval time of ASC	Empirical distribution	s
	Simulation time	Constant	s
Costs related	Purchase of 1 AGV	Constant	€
	Operational costs of 1 AGV per hr	Constant	€

The number of AGVs will be varied during the experiments to determine the optimal number needed in the system. Chapter 7 explains the experiments.

KPI's

To assess the performance of the three routing types, comparisons are made based on the determined KPI's. The KPI's should be chosen such that it best reflects the routing and terminal performance. The routing objective in the model is to choose the shortest path to the destination. The KPI's chosen are:

- **Costs of the AGV system**
  The routing type with the best balance between performance and costs should be chosen. Only costs of the AGVs are assumed and are calculated with:

$$AGV\ costs\ [€] = \#_{AGVs} \cdot c_{fixed} + \#_{AGVs} \cdot c_{variable} * t_{operating\ hours} \tag{4.1}$$

- **Terminal performance: container handling efficiency**
  Terminal performance is related to realized QC capacity, the QCs should handle containers continuously. Therefore the amount of handled containers is divided by the theoretical amount of containers that could be handled by continuous QC operation.

$$Handling\ efficiency\ [-] = \frac{Handled\ containers\ in\ simtime}{\frac{Simtime}{Cycle\ time\ of\ QC} \cdot \#\ QCs} \tag{4.2}$$

- **Routing performance: travel time delay**
  The AGV has two travel components per container: travelling from its location at the moment of container assignment to a stack and travelling with container from stack to QC. With container assignment is meant the moment that the AGV is dispatched to an ASC.
  For each container job, the travel time of both components is determined and the corresponding minimal travel time of both components. The routing objective is to choose the shortest path, thus with minimal travel time is meant the travel time of this shortest path (no obstacles assumed). After completion of the simulation, data from all jobs is converted to the following outputs:
  - 95% confidence interval of realized travel times from assignment location to stack
  - 95% confidence interval of realized travel times from stack to QC
  - 95% confidence interval of minimal travel time from assignment location to stack
  - 95% confidence interval of minimal travel time from stack to QC

  The realized travel times are summed and the minimal travel times are summed. The realized travel times are divided by the minimal travel times to obtain a delay factor that can be compared to other routing types.

$$Travel\ time\ delay\ [-] = \frac{Realized\ travel\ time\ to\ ASC + realized\ travel\ time\ to\ QC}{Minimal\ travel\ time\ to\ ASC + minimal\ travel\ time\ to\ QC} \tag{4.3}$$

4.2 Control and interaction of subsystems in the terminal model

A basic system and control cycle is shown in Figure 23. The controller has a control objective that it wants to fulfill. Based on the objective, actions are decided upon that the system must execute. The system has a state that is updated each time step. The state is influenced by the actions taken and disturbances on the system. Measurements can be made from the system and are send to the controller. The controller on its turn decides which actions should be taken.

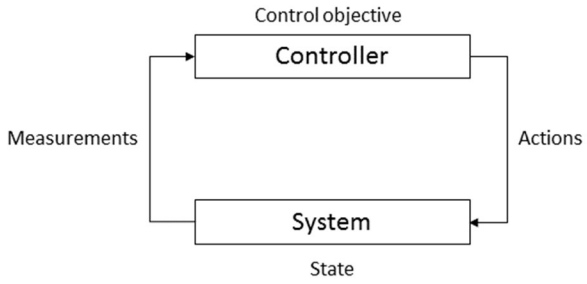


Figure 23 - System and control cycle.

The terminal model has multiple subsystems (or elements): the main subsystems are those of the ASCs, QCs and AGVs. All three are implemented as an object that has its own procedure. A container cycle with interaction of processes from these subsystems is explained briefly before.

The terminal model is of the type cooperative control (Figure 8). The control architecture of the terminal model is as shown in Figure 24. There are multiple QCs, ASCs and AGVs. Each element of the same follows the same procedure, thus QC1 follows the same procedure as QC3. Each element has its own system and control cycle. The controllers of the elements must ensure that the processes are executed in the correct order. There is not an overall controller per element type and no supervisory controller. Information is exchanged between the elements, but not between sub-elements (for example: not between AGV1 – AGV2, but between AGV-QC). Information is gathered from the container that is being handled.

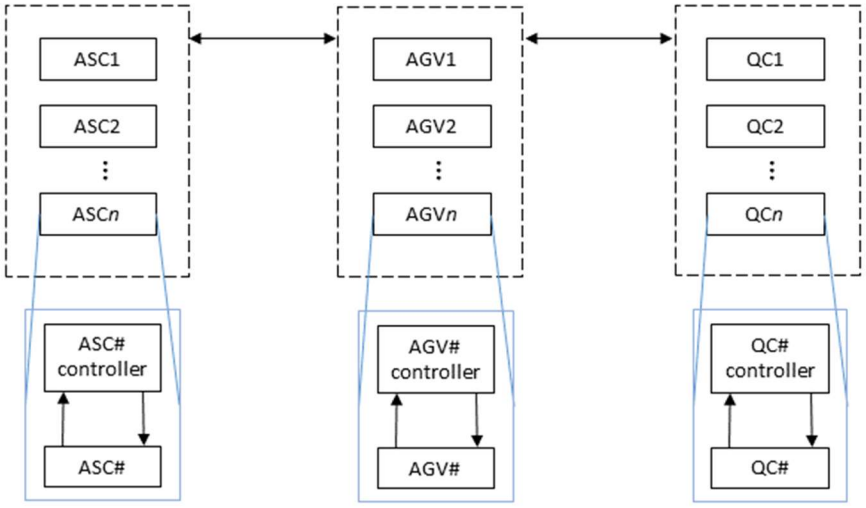


Figure 24 - Control architecture of terminal model.

The terminal model is implemented as a discrete event simulation. This means that the simulation revises each element at each event to see if a new steps can be taken in a procedure. During the simulation, the ASC procedures are visited sequentially (first ASC1 will be checked for new steps, then ASC2 etc.). The same applies to the AGV and QC procedures. AGVs waiting at a QC are unloaded in order of arrival (FIFO).

The control cycle of an ASC element is as follows:

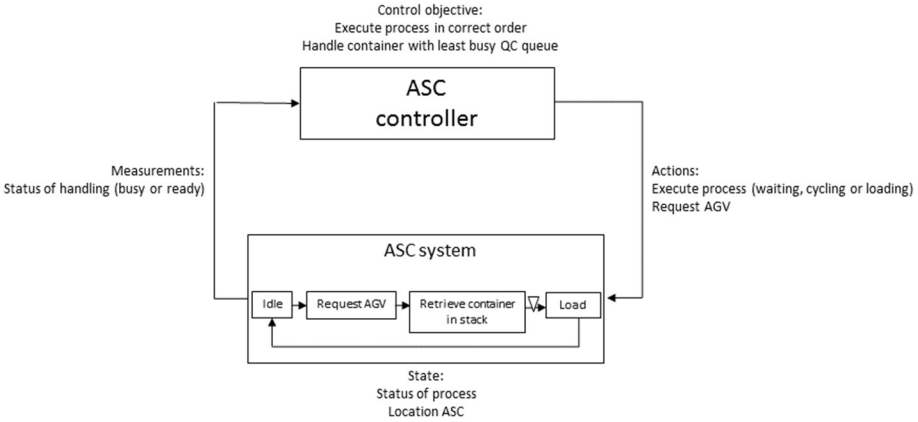


Figure 25 - ASC control cycle.

The controller must make sure that the ASC process is executed in the correct manner. If there are multiple containers waiting in the stack, the controller decides which container to handle first. The one with a QC destination that has the smallest handling queue is chosen.
Dispatching of AGVs is not part of this controller, the ASC only places a request for an AGV.
The controller requires (sends) information from (to) an AGV element, the QC handling queue and containers-in-stack-queue. Also information is send to the AGV request handler.

The control cycle of an AGV element is as follows:

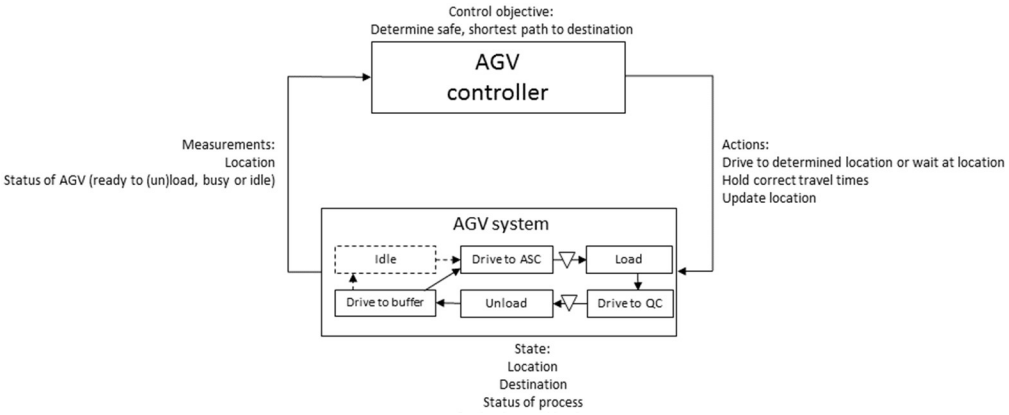


Figure 26 - AGV control cycle.

The AGV controller decides upon the path to take, either node by node or direction (depending on routing type implemented). Also the controller makes sure that the process is executed in the correct order. The actions of the controller are the next driving move (either node or direction) and how long

the simulation must hold (time = distance/speed) before the locations are updated. The system its state consist of the location, destination and status of the process. The measurements of the system are location and status of the process.

Information requested from other elements is if the AGV is requested and if so, where it should drive to. In case of waiting at cranes, information on reactivation is needed.

The control cycle of an QC element is as follows:

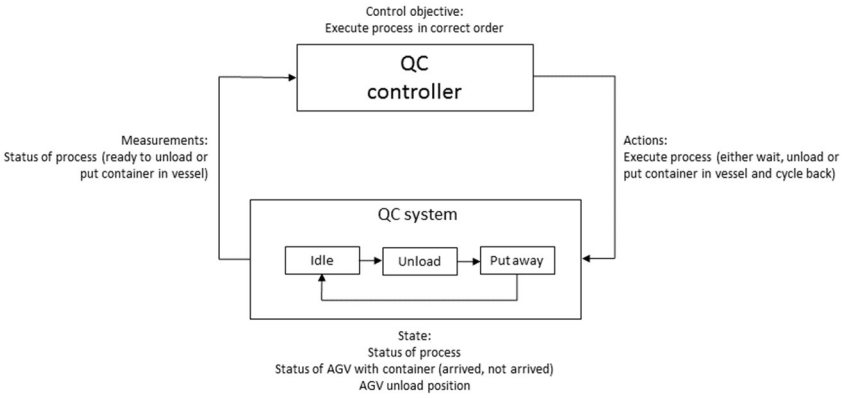


Figure 27 - QC control cycle.

The controller makes sure that the process is executed correctly. Actions depend on the progress of the QC. In case of unloading, the AGV must be arrived. The system its state consists of the status of the process, and status and location of the AGV. Measurements of the controller are the status of the process to decide upon the next action.
Information is exchanged with an AGV element: has the AGV arrived and is unloading completed.

Besides the shown control cycles, there is a request function for the ASCs and QCs. The ASCs make a request for an AGV. The QCs make a request to handle an AGV.

4.3 Implementation

More information will now be presented on the implementation of the terminal model into Delphi.

Input data of the model

The terminal model has a fixed topology, the corresponding input values are presented in Table 7. A reference case is assumed, based on current operation characteristics. Its values are presented in the table. Costs related parameters are fixed.

Table 7 - Implemented standard input values of terminal model.

	Parameter	Value	Unit
Topology	Size of terminal (w x h)	416 x 108	m
	d_{ASC}	32	m
	d_{QC}	96	m
	d_{buffer}	Depends on routing type	m
	$d_{unload\ spot}$	12	m
Operational	Number of AGVs	To be varied	-
	Speed of AGV	4	m/s
	Acceleration of AGV	1	m/s^2
	Deceleration of AGV	1	m/s^2
	Unload time of AGV at QC	18	s
	Put away time of QC	N(120,30)[1]	s
	Load time of AGV at ASC	10[1]	s
	Container retrieval time of ASC	Empirical distribution[2]	s
	Simulation time	Determined later	s
Costs	Purchase of 1 AGV	282000[3]	€
	Operational costs of 1 AGV per hr	4.50[3]	€

Output of the model

The model output consist of numerical output, graphs and an animation. These are presented in the Graphical User Interface (GUI). More information on the GUI is presented in Appendix C.

Numerical outputs are the amount of handled container per AGV per hour, the total amount of handled containers and the moves/hr/QC. Also the mean, 95% confidence interval and standard deviation of AGV cycle time, drive time to ASC, drive time to QC and delay of the latter two are given.

There are four graphs. The first graph shows the amount of handled containers in time for ASC and QC. The second graph shows the mean waiting time of AGVs per QC (once they are arrived at an unloading spot). The third graph shows the travel time delays of driving to ASC and QC (both with and without waiting at the buffer zone for a QC). The fourth graph shows the average amount of AGVs driving and the amount of containers waiting to be handled. All graphs are a function of simulation time.

A 2D animation is made to visualize the driving of the AGVs. The animation is updated every second and displays the current positions of the AGVs. Each routing type requires a change in topology: the traditional routing has a topology of paths, free-ranging requires no path topology and the mixed requires partly a topology of paths. An example of the animation is as follows (example from traditional routed terminal):

[1] Input as reported by Ottjes and Veeke [59].
[2] Distribution as reported by Vis and Harika [18].
[3] Input as reported by Henesey, Davidsson and Persson [63].

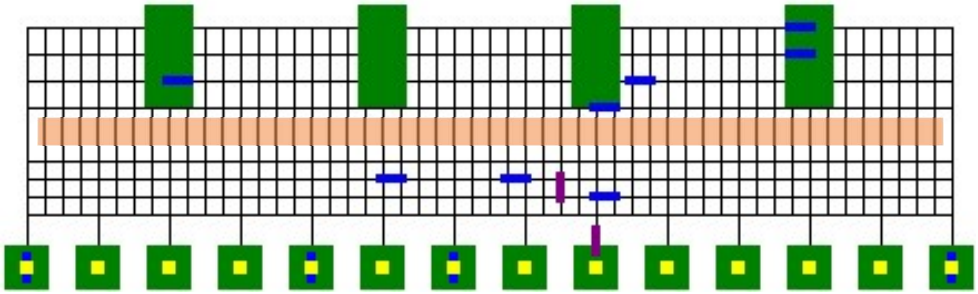


Figure 28 - Animation in the GUI (traditional routed terminal).

The ASCs are represented each by a green square at the bottom of the animation. A stack that has containers to handle is colored by a yellow square at its ASC. The QCs are the rectangles at the top of the animation. Numbering of equipment (ASC and QC) starts from the left side of the figure.
AGVs that are idle are colored yellow (not shown) and are positioned at the middle, vertical lanes (in the orange zone). AGVs are initially blue, but when its first job is completed and the AGV is assigned directly to a new job, the AGV turns purple.

4.4 Conclusions

Due to the stochastic complexity of the system, the system will be modelled as a discrete-event simulation. There are four elements in the model: QC, ASC, AGV and container. The former three each have a procedure to handle the container. The container element acts as an information source for the procedures. Only the loading process is assumed, thus transport from stack to quay. KPIs of the system are AGV costs, handling efficiency and travel time delay.

Each of the elements, except the container, has a system and control cycle. An AGV communicates with a QC or ASC, a QC will not communicate with an ASC or vice versa. The controllers ensure correct execution of the procedures. Measurements are done to the status of the system. To determine upon control actions, information from the own and another system can be used. Actions can be physical, for example “load”, or of control nature, for example “activate AGV”.

The terminal model is implemented in Borland Delphi including TOMAS. Inputs of the model concern the topology, operations or costs. Outputs are numerical data, graphs and an animation.

Chapter 5 – Routing types within the terminal model

Routing is the main research focus in this report. Three algorithms are to be developed that each fulfill a different routing type. These algorithms are implemented in the terminal model (Chapter 4).
The first type of routing is routing as found on a traditional terminal, the second type uses free-range routing by the potential field method. The third type is a mixed version of traditional and free-range routing.

5.1 Traditional routing of AGVs

The traditional routing topology and execution are based on literature. Different cases and ideas are merged to form the routing model. The main motivation for the implemented topology is based on the study of Bae et al. [9]. This section describes the assumptions, execution and implementation of routing into the terminal model.

5.1.1 Description and assumptions of routing

The following assumptions are made:

- Paths in x-direction are one directional. In total there are 8 horizontal paths: 2 sets of 4 lanes. In each set, travelling on the top two lanes is restricted to the right while the bottom 2 lanes are restricted to left. Advised by Li et al. [26]. Paths in y-direction are bidirectional.
- AGVs travel from node to node. Nodes are dimensionless and can either be occupied by an AGV or the node is free. Each node is controlled by a semaphore with capacity 1 to keep track of the occupancy.
- Route planning goes incremental using a Dijkstra algorithm to find the shortest path.
- Idle AGVs are parked at a vertical lane in the middle of the terminal, one out of three methods as described by Vis [22]. Dispatching to a parking place is later explained. The parking places are also used for AGVs waiting for their QC spot to be available.
- Acceleration is always compensated in the first travelled arc of the acceleration, also when the acceleration distance is longer than the arc distance between two nodes. Only accelerations to full speed are assumed.
- Deceleration is always compensated after the AGV has come to a stop. The extra travel time due to deceleration is compensated at once. Only decelerations to a standstill are assumed.

Traffic control is arranged with semaphores, each node has a capacity of 1.
A Dijkstra algorithm calculates the shortest path to the AGV its destination. The AGV drives to the first node on that path. Once arrived on that node, the Dijkstra algorithm determines the next step to take. This iterating continues until the destination is reached. This way, the current node capacities are constantly up to date.
Planning thus goes incrementally, but final destination is considered each iteration. If the node ahead is blocked by another AGV, the current AGV waits at this current node until the next node is free or the AGV takes a different route.

Acceleration and deceleration

Acceleration and deceleration are implemented into the model. Because the model plans incrementally, it is not always possible to hold the correct drive times. This happens for example when the acceleration or breaking distance is larger than the distance between current node and next node. Also, deceleration is detected and corrected afterwards.

Drive times per iteration are calculated as follows:

- AGV drives constantly
  Drive time is the distance of the iteration divided by the AGV speed.

$$t = \frac{distance(current\ node, next\ node)}{v} \tag{5.1}$$

- AGV accelerates and $d_{acc} > d_{current\ node,\ next\ node}$
  The acceleration distance is compensated for in the first iteration of the acceleration. It is assumed that the AGV can keep accelerating until full speed is reached. The time spend on the remaining part (acceleration distance minus current iteration distance) of the acceleration distance is calculated and subtracted from the total acceleration time.

$$t = \frac{2d_{acc}}{v} - \frac{d_{acc} - d_{current\ node,next\ node}}{v} = \frac{d_{acc} + d_{current\ node,next\ node}}{v} \tag{5.2}$$

- AGV accelerates and $d_{acc} <= d_{current\ node,\ next\ node}$
  The acceleration distance is subtracted from the current iteration distance. The remaining part is driven at full speed. The other part is spend at accelerating.

$$t = \frac{d_{current\ node,next\ node} - d_{acc}}{v} + \frac{2d_{acc}}{v} = \frac{d_{acc} + d_{current\ node,next\ node}}{v} \tag{5.3}$$

- AGV decelerates
  The entire deceleration time is compensated after the AGV has come to a stop. Because it is not possible to look into future actions, a stop is noticed after it occurred. The compensation consist of the difference in drive time while driving at full speed and decelerating. The entire deceleration distance is compensated at once.

$$t = \frac{2d_{dec}}{v} - \frac{d_{dec}}{v} = \frac{d_{dec}}{v} \tag{5.4}$$

5.1.2 Traditional routing PDL

There are three driving moments in the model: driving from buffer to ASC, driving from ASC to QC and driving from QC to buffer. All three are implemented separately because of some small differences. The basics are however similar and is shown next.

Traditional routing procedure
Receive destination
While current location of AGV is not destination
  Get temporary destination
  Drive to temporary destination
  Update node claims and current location
  ASC->QC: If at buffer place and destination is occupied
    Standby

Routing goes incrementally and thus each iteration, the AGV can drive to an adjacent node. The temporary destination is determined with a Dijkstra algorithm (Appendix D).

The drive time to the temporary destination depends on the status of the AGV (accelerating, decelerating or driving at full speed) and the distance of the current location to temporary destination. The correct drive time is calculated with the function getDriveTime (Appendix D).

AGVs travel from node to node and each node they occupy must be claimed such that other AGVs cannot claim the node. When the AGV moves, new nodes must be claimed and old nodes must be released. When the AGV arrived at the temporary destination, its location is updated.

Differences between routing moments are found in the receiving of a destination:

- Driving from buffer to ASC
  Each container has a stack number assigned. When a container is assigned to the AGV, it retrieves this information from the container and convert it to coordinates. The destination of the AGV is the location of the container.
- Driving from ASC to QC
  The container that is picked up at the ASC has a QC destination. This destination is also the destination of the AGV. But note that each QC has four unloading spots and one must be chosen. The function getCraneSpot (Appendix D) checks which spot is free and that node is assigned. If all spots are occupied, the spot that will be soonest available is chosen.
- Driving from QC to buffer
  The AGV must be assigned a buffer location where it can go idle. A buffer location that has the shortest travel time is chosen. Some buffer places are prohibited or discouraged to park on to ensure evenly spaced crossings for the AGVs. A buffer place is chosen with the function getBufferNode (Appendix D).

5.1.3 Implementation

The terminal model implementation as discussed in Chapter 4.3 is applicable. The topology of paths is constructed of arcs between nodes. It is assumed that the distance between nodes is 8, 12 or 24 meters (Figure 29) to reduce algorithm complexity while travelling in a safe manner. The path planning algorithm requires that the first 3 nodes of the path are free, this gives enough distance to make a stop from full speed:

$$s_{min} = vt + \frac{1}{2}at^2 = \frac{v^2}{2a} = \frac{4^2}{2 \cdot 1} = 8\ m \tag{5.5}$$

There are 53 vertical crossings in the topology and thus 53 parking places, these are located at y = 60 and Δx = 8.

Because the AGV its length exceeds the distance between two nodes in most cases, each AGV claims two nodes simultaneously. These are denoted as front and back node. Starting from the front node, the next temporary destination (i.e. adjacent node) is determined. If a new temporary destination is found, the back node claim is released and updated with the front node claim. The new front node claim is the temporary destination node. If the AGV cannot travel further, the current claims remain.

The terminal animation looks as follows:

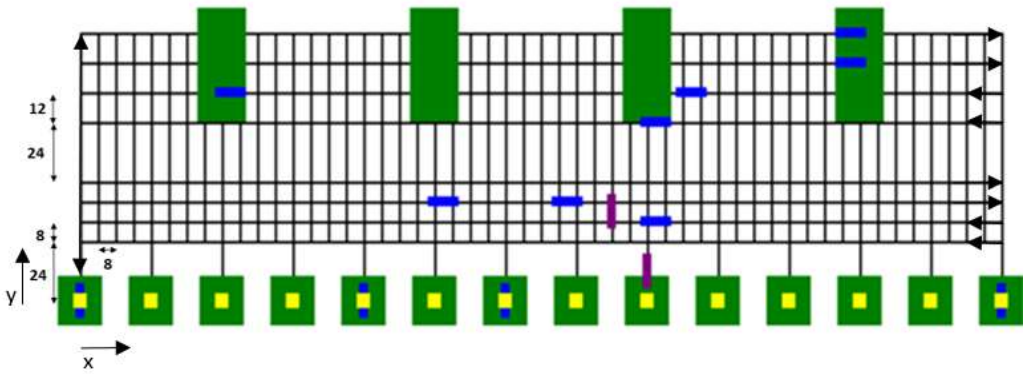


Figure 29 – Traditional routing terminal implementation in Delphi.

AGVs that are idle are colored yellow (not shown) and are positioned at the middle, vertical lanes. AGVs are initially blue, but when its first job is completed and the AGV is assigned directly to a new job, the AGV turns purple.
Speed of the AGVs is fixed at 4 m/s, unless otherwise specified by the user. This is roughly the average speed of maximum corner and straight ahead speed of AGVs running at the ECT Delta terminal [19]. Acceleration and deceleration of the AGV are 1 m/s^2.

Initialization

When starting the model, the AGVs are placed evenly spaced onto the buffer places at location (x,60). The empty places receive weight factor infinite so that there are evenly spaced passages for the AGVs during simulation. The five places closest to each QC receive a weight factor to discourage parking at those places.

Hold – when all QC unload spots are occupied

When an AGV travels towards a QC it has received an unloading spot. This spot might be occupied by another AGV. When the AGVs travels across a buffer place, a procedure is initiated that checks if its unloading spot is available. When not available, the AGV is held at the buffer place until the spot is available. This prevents the AGV from swarming next to the QC.

Usage of buffer places

Buffer places are used as parking place where AGVs are idle or as a crossing. All situations and range of chosen buffer places per QC are explained into more detail in Appendix D.

5.2 Free-range routing of AGVs

Free-ranging does not require a topology of paths on which the AGVs must travel. This way, infinite route options are possible. The potential field method is chosen as the routing method to implement, see Chapter 3.2. This is a reactive method, AGVs heads towards its destination and avoids obstacles on its way.

5.2.1 Description and assumptions of routing

The following assumptions are made for the potential field method:

- AGVs are represented as a circle. Goals (quay crane spots and ASCs) and obstacles (other AGVs) are also represented as a circle. Each circle has a radius and spread (Figure 30). The former denotes that either the goal is reached or absolute no-go distance (goal versus obstacle, respectively). The spread denotes vicinity in which the AGV its speed is reduced depending on distance.
- Goals have an attractive force while obstacles have repulsive forces.
- An escape force is implemented when the AGV is too close to an obstacle. No deadlock detection and avoidance intelligence is implemented.
- The potential field is updated each second.
- The (un)loading spots at the QC and ASCs do not require specific unloading positions of the AGVs (AGV location must be near (un)loading spot).
- Acceleration and deceleration differ each iteration depending on obstacle and goal distance.

The topology of the terminal is as in Figure 21.

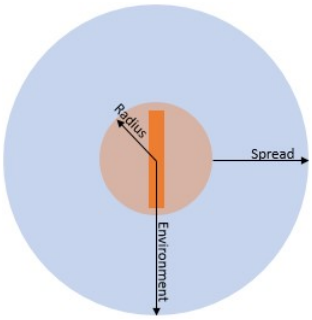


Figure 30 - Representation of AGV in the PFM.

When the AGV must travel towards a destination, its potential field for that moment is created. The destination (goal) has an attractive force in both x and y direction, the strength depends on distance and angle. Each AGV other than the current AGV is an obstacle and each has a repulsive force, the strength of repulsion depends on the distance and angle between obstacle and AGV. The forces are combined into a resultant repulsive force in both x and y direction. The sum of the attractive and repulsive force is the direction in which the AGV will travel:

$$F_{resultant} = \sqrt{{F_{x,total}}^2 + {F_{y,total}}^2} \tag{5.6}$$

$$\theta_{resultant} = \tan^{-1}\left(\frac{F_{y,total}}{F_{x,total}}\right) \tag{5.7}$$

Each second, a new direction and velocity is determined for the AGV. After an iteration, the new location of the AGV is updated.

Distance is calculated as the Euclidean distance:

$$d(AGV, goal) = \sqrt{(x_{goal} - x_{AGV})^2 + (y_{goal} - y_{AGV})^2} \tag{5.8}$$

An escape force is implemented if the AGV is inside the radius of an obstacle, the obstacle then has a large repulsive force.

5.2.2 Potential field routing PDLs

The AGV has three drive moments: from buffer to ASC, from ASC to QC and from QC to buffer place. All three are implemented separately because of some small differences. The basics are however similar and is shown next.

PFM routing procedure
Receive destination
While location of AGV is not destination
    Calculate speed and direction
    Hold one second
    Update location of AGV
    ASC -> OC: If near QC and AGV has no unload spot
        Standby

The only difference between the three routing components is found in the receiving of a destination:

- Driving from buffer to ASC
  Each container has a stack number assigned. When a container is assigned to the AGV, it retrieves this information from the container and convert it to coordinates. The destination of the AGV is the location of the container.
- Driving from ASC to QC
  The container that is picked up at the ASC has a QC destination. This destination is also the destination of the AGV. But note that each QC has four unloading spots and one must be chosen. The function getCraneSpot (Appendix D) checks which spot is free and that node is assigned and claimed. If all spots are occupied, the AGV travels to the lowest QC unload spot. Close to its destination, the AGV is held until an unload position is available.
  Note the difference with traditional routing: the unload node must be free, in comparison with traditional routing where this was not obligated. Traditional routing could not claim an unload spot yet, because then this node could never be reached (routing sees the node as occupied).
- Driving from QC to buffer
  The AGV must be assigned a buffer location where it can go idle. A buffer location that has the shortest travel time is chosen. Some buffer places are prohibited or discouraged to park on to ensure evenly spaced crossings for the AGVs. A buffer place is chosen with the function getBufferNode (Appendix D).

Each second the potential field is updated and a new direction and speed must be determined for the AGV. The function CalculateDirection determines the speed and direction to head to.

CalculateDirection
Calculate attractive force (function: FindDestination)
Calculate resultant repulsive force (function: AvoidObstacles)
Add attractive and resultant repulsive force
Calculate direction
Calculate speed

CalculateDirection calls function FindDestination to determine the attractive force.

FindDestination
Calculate distance between AGV and destination
Calculate angle between AGV and destination
If distance < radius of goal
    AGV has arrived, no attractive force
Else if radius of goal <= distance <= environment of goal
    Attractive force depends on distance to goal and scaling parameters
Else if distance > environment of goal
    Attractive force is maximal

CalculateDirection calls function AvoidObstacles to determine the repulsive force.

AvoidObstacles
For all obstacles (i.e. AGVs)
    Calculate distance between AGV and obstacle
    Calculate angle between AGV and obstacle
    If distance < radius obstacle
        Escape force
    Else if radius obstacle<= distance <= environment obstacle
        Repulsive force depends on distance to obstacle and scaling parameters
    Else if distance > environment obstacle
        Repulsive force is zero
    If AGV drives close to terminal boundaries
        Add repulsive force in opposite direction

5.2.3 Implementation

The terminal model implementation as discussed in Chapter 4.3 is applicable.
There are 27 buffer places, distance between two places is 16 meters. This is larger than with traditional routing (Δx=8m) because of two reasons: the AGVs can enter and park at every angle of the buffer place and second, to ensure smooth driving (no escape force activated) of the AGV when the two places near the AGV are occupied.

Determination of routing parameters

Table 8 shows the routing parameters that must be determined. This is done empirically.

Table 8 - Potential field routing input parameters.

Parameter	Type of value	Unit	Explanation
radiusGoal	Constant	m	Radius of goal
radiusObstacle	Constant	m	Radius of obstacle
spreadGoal	Constant	m	Environment of goal in which speed is gradually reduced
spreadObstacle	Constant	m	Environment of obstacle in which repulsive force is gradually increased
reduced radiusObstacle	Constant	m	Near a QC, the radius of obstacles is reduced so that the destination can be reached
Reduced spreadObstacle	Constant	m	Near a QC, the environment of the AGV is reduced
Alpha	Constant	-	Scaling factor for goal attractive force
Beta	Constant	-	Scaling factor for obstacle repulsive forces

Attractive force

There are three statements to decide upon the attractive force.
First statement:

$$if\ d(AGV, Goal) < radiusGoal \quad (5.9)$$
$$deltaX = 0$$
$$deltaY = 0$$

If this statement is true, the goal is reached and does not exert an attractive force. The (un)loading locations require precision and therefore it is decided to make radiusGoal = 1.

Second statement:

$$If\ radiusGoal \leq d(AGV, Goal) \leq (spreadGoal + radiusGoal) \quad (5.10)$$
$$deltaX = alpha * (d(AGV, Goal) - radiusGoal) * \cos(\theta_{AGV,Goal})$$
$$deltaY = alpha * (d(AGV, Goal) - radiusGoal) * sin(\theta_{AGV,Goal})$$

If this statement is true, the attractive force is linearly decreased as the AGV approaches its destination.

The parameters are determined with the third statement:

$$If\ (d(AGV, Goal) > (spreadGoal + radiusGoal) \quad (5.11)$$
$$deltaX = alpha * spreadGoal * \cos(\theta_{AGV,Goal})$$
$$deltaY = alpha * spreadGoal * \sin(\theta_{AGV,Goal})$$

If this statement is true, the attractive force is maximal. It is decided that the speed is maximal at a distance > 10 and thus spreadGoal = 10 – radiusGoal = 9 m.
The maximum speed of the AGV is assumed 4 m/s. One iteration lasts 1 second and thus the attractive force should be:

$$F_{att} = \sqrt{deltaX^2 + deltaY^2} = 4 * 1 = 4. \quad (5.12)$$

Assuming the following situation: there are no obstacles, an AGV drives in x-direction towards a goal on the right. The function should thus deliver deltaX = 4 :

$$deltaX = \text{alpha} * \text{spreadGoal} * \cos(0) = \text{alpha} * \text{spreadGoal} = 4 \quad (5.13)$$
$$deltaY = 0$$

Now alpha can be determined:

$$alpha = \frac{v_{max}}{spreadGoal} = \frac{4}{9} \tag{5.14}$$

Repulsive force

Similar to the attractive force, the repulsive force is determined by three statements.
The first statement:

$$\begin{aligned} &if\ d(AGV, obstacle) < radiusObstacle \\ &deltaX = -sign(\cos(\theta_{AGV,obstacle})) \cdot \infty \\ &deltaY = -sign(\sin(\theta_{AGV,obstacle})) \cdot \infty \end{aligned} \tag{5.15}$$

An escape force is activated when the AGV is within the radius of the obstacle. The obstacles, i.e. (moving) AGVs, require a radius of 16 meters. This is based on the extreme situation that two AGVs drive in full speed towards each other [14]:

$$r_{AGV} = \frac{{v_{max}}^2}{a_{max}} = \frac{4^2}{1} = 16\,m \tag{5.16}$$

Even though escape force is infinite, the final proposed speed is limited to 4 m/s in each iteration.

The second statement:

$$\begin{aligned} &if\ radiusObstacle \leq d(AGV, obstacle) \leq (spreadObstacle + radiusObstacle) \\ &deltaX = -beta\big(spreadObstacle + radiusObstacle - d(AGV, obstacle)\big)\cos(\theta_{AGV,obstacle}) \\ &deltaY = -beta\big(spreadObstacle + radiusObstacle - d(AGV, obstacle)\big)\sin(\theta_{AGV,obstacle}) \end{aligned} \tag{5.17}$$

The repulsive force increases linearly from distance = spreadObstacle + radiusObstacle towards distance = radiusObstacle. With the third statement, spreadObstacle is determined. Afterwards the parameter beta is determined (see next).

The third statement:

$$\begin{aligned} &if\ d(AGV, obstacle) > (spreadObstacle + radiusObstacle) \\ &deltaX = 0 \\ &deltaY = 0 \end{aligned} \tag{5.18}$$

It is chosen that the obstacle does not have a repulsive reaction if the AGV is at a distance > 36. Thus spreadObstacle = 36 – radiusObstacle = 20.

Assume a situation where goal and obstacle are at the exact same position. It is desired that the goal and repulsive forces are balanced such that the AGV slows down and comes to a stop when distance is 16 m. At distance = 16 m, the goal has an attractive force of 4, thus the repulsive force should be:

$$deltaX = -beta * spreadObstacle * \cos(0) = -beta * spreadObstacle = -4 \tag{5.19}$$

And thus:

$$beta = \frac{v_{max}}{spreadObstacle} = \frac{4}{20} = \frac{1}{5} \tag{5.20}$$

It appears however, that using this beta leads to a last minute obstacle avoidance, something that may start a bit earlier. Figure 31 shows the influence of beta on the trajectory of the AGV, beta as calculated is represented with the red line. Beta is somewhat increased to start the obstacle avoidance somewhat sooner, the blue line. It can be seen that decreasing beta leads to oscillating behavior near the obstacle, the yellow line.

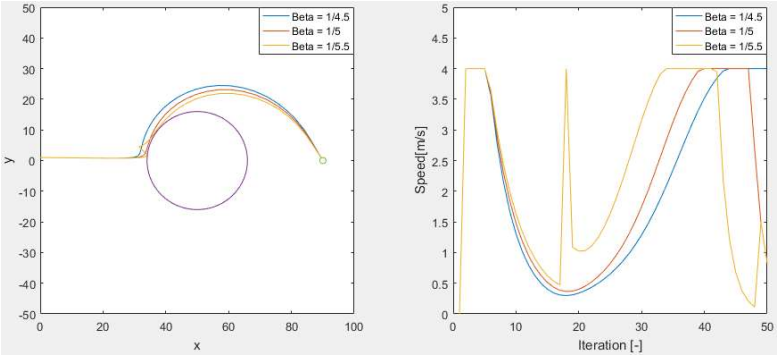


Figure 31 – Left: Influence of beta on the avoidance trajectory. Right: Influence of beta on the speed profile.

Reduction of parameters

The QC unloading spots are close to each other, 12 meters is assumed. Four AGVs cannot simultaneously wait at or reach a QC if the radius of an obstacle is 16 meters. Therefore the radius and spread of obstacles are decreased if the obstacle does not move. The radius of the still standing obstacle is reduced to 10 meters and the spread to 10 meters. This way, the moving AGV can travel towards its QC unloading spot.

The implemented routing parameters are presented in Table 9.

Table 9 – Implemented input parameters for potential field routing.

Parameter	Value	Unit
radiusGoal	1.00	m
radiusObstacle	16.00	m
spreadGoal	90.00	m
spreadObstacle	20.00	m
reduced radiusObstacle	10.00	m
reduced spreadObstacle	10.00	m
Alpha	0.44	-
Beta	0.20	-
Speed of AGV (max.)	4.00	m/s

Terminal animation

The terminal animation looks as follows:

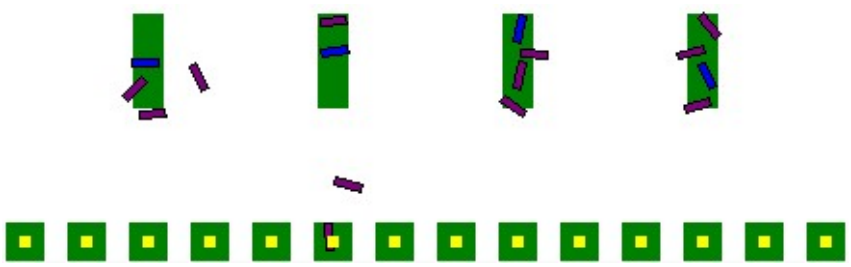


Figure 32 - Potential field routing terminal implementation in Delphi.

AGVs that are idle are colored yellow (not shown) and are positioned at the buffer places. AGVs are initially blue, but when its first job is completed and the AGV is assigned directly to a new job, the AGV turns purple.

Initialization

When starting the model, the AGVs are evenly spaced onto the parking places at location (x,45). The empty places receive weight factor infinite so that there are evenly spaced passages for the AGVs during simulation. The 3 places closest to each QC receive a weight factor to discourage parking there.

Hold – when all QC unload spots are occupied

When an AGV travels towards a QC it might or might not have received an unloading spot, when a spot is available or all spots are occupied, respectively. When no spot is received, the AGV travels to the lowest unloading spot and is held when it is closer than 30 meters to this destination. The AGV waits here until an unload spot is assigned.

The unload spot (i.e. node) is thus only claimed by one AGV and is already claimed when the AGV drives towards this node (note the difference with traditional routing).

Usage of buffer places

Buffer places are used as parking place where AGVs are idle or as a crossing. All situations and range of chosen buffer places per QC are explained into more detail in Appendix D.

5.3 Mixed routing

The third routing type is chosen to be a mix from the traditional routing and free-range method. Near the QCs is a busy area that requires guided behavior. The traditional routing is suited to arrange guided behavior and is thus implemented near the QCs. The rest of the terminal is less crowded and benefits from free-range routing.

5.3.1 Description and assumptions of routing

Mixed routing is a combination of traditional and free-range routing as were implemented in Chapter 5.1 and Chapter 5.2, respectively. Their assumptions are also valid here. New assumptions are:

- Free-range routing using the potential field method is chosen as basis routing type.
- Near each QC is a topology of paths on which the AGVs travel to and from the QC. The paths in x-direction run through each QC unload spot are one directional to the right. The paths in y-direction are one directional up at the left of the QC and one directional to down at the right of the QC. A bidirectional path connecting all buffer places runs below the QCs.
- The buffer places are reached by traditional routing (via paths). Idle AGVs are parked at such a place.
- There are four routing transfer points to transfer from free-range to traditional routing (traveling towards the QC), one at each left side of the QC. Transfer from traditional routing to free-ranging can take place at each other buffer place.
- Each routing transfer point has a capacity of one.
- Free-ranging is not allowed in the QC areas where traditional routing is applied.

The topology of mixed routing is as follows:

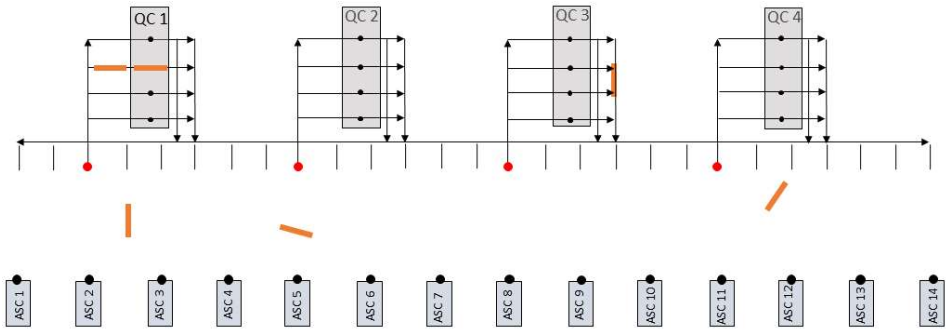


Figure 33 - Mixed routing terminal topology, AGVs represented with orange.

Free-ranging is applied in the bottom side of the terminal. The upper part, starting above the buffer places, uses traditional routing.

When an AGV travels from an ASC towards a QC, it must first go to the routing transfer point, the red dots in Figure 33. Only one AGV can be on the routing type transfer point.
Once arrived at the transfer point, the routing type is switched to traditional routing arranged with semaphores. The AGV travels node by node to its destination.
Traditional routing is applied until the AGV crosses a buffer place, at that point the switch to free-ranging is made again.

5.3.2 Mixed routing PDL

There are three drive moments:

1. Routing from buffer place or QC to ASC
   Driving from buffer place to ASC is arranged by free-range routing. When the AGV is reassigned while driving towards the buffer, it is traditional routed. A new buffer place is calculated that contributes to the shortest path to the ASC. That buffer place then functions as routing transfer point to free-ranging.
2. Routing from ASC to QC unload spot
   This consists of both traditional routing and free-range routing. First the AGV free-ranges from the ASC towards the routing transfer point. There, the AGV uses traditional routing to reach the QC unload spot.
3. Routing from QC unload spot towards buffer
   This is arranged by traditional routing.

The traditional and free-range routing PDLs are similar to the ones explained in Chapter 5.1.2 and Chapter 5.2.2, respectively.

Receiving of a destination per driving moment is similar as in traditional routing (Chapter 5.1.2).
The only difference is found in the assignment of an unload spot when the AGV drives from ASC to QC: the function getCraneSpot (Appendix D) checks which spot is free or soonest available and that node is assigned and claimed. This claim is released when unloading is completed. If all spots are occupied, an available waiting spot is chosen (to the left of each unload spot). If all waiting spots are occupied, the AGV waits at the routing transfer point. The difference with traditional routing is that an unload spot can be chosen that has an available waiting place.

5.3.3 Implementation

The terminal model implementation as discussed in Chapter 4.3 is applicable. A path topology near each QC is implemented. The paths along each unload spot is 40 meters long, with nodes at each 8 meters so that multiple AGVs can be on a path.
The distance between two buffer places is 16 meters to ensure smooth traveling from a buffer place towards the ASC (no escape forces are activated). There are 27 buffer places, but four buffer places are restricted to function as buffer and may only be used as routing transfer point from free-range to traditional routing.

The terminal animation is as follows:

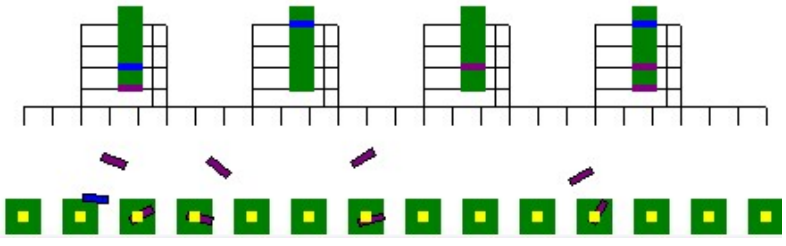


Figure 34 - Mixed routing terminal implementation in Delphi.

The implemented paths can be seen at the QCs, each unload spot has one horizontal path, the buffer places are connected by one bidirectional horizontal path.

AGVs are initially blue, but when reassigned they turn purple. Idle AGVs are yellow and parked in a buffer place (not shown).

As traditional routing is applied: the AGV speed is assumed 4 m/s and both acceleration and deceleration are assumed 1 m/s^2. Note that when free-ranging is applicable, the potential field method determines the speed and acceleration, deceleration. Speed is limited to 4 m/s. Potential field parameters are equal to the ones presented in Table 9.

Initialization

When the model is started, the AGVs are placed at the buffer places from left to right, except for the routing transfer points. The buffer places that function as routing transfer points (from ASC to QC) receive weight factor infinite to prohibit parking.

Approaching a routing transfer point (free-range to traditional routing)

When the AGV travels from ASC to QC, it must travel to the routing transfer point to switch to traditional routing before it can reach the QC. When the AGV is at a distance ≤ 20 m from this point, the AGV enters a queue that handles the approaching AGVs of that point. There are four of such queues, one for each QC routing transfer point.
A request procedure then checks if the transfer point is free. If so, the AGV can travel towards the point. Else, the AGV is held at its current position until the point is free. There are four of such request procedures, one for each QC.
In case there are multiple AGVs waiting to approach the transfer point, the queue is handled FIFO and only one AGV can proceed at the time.

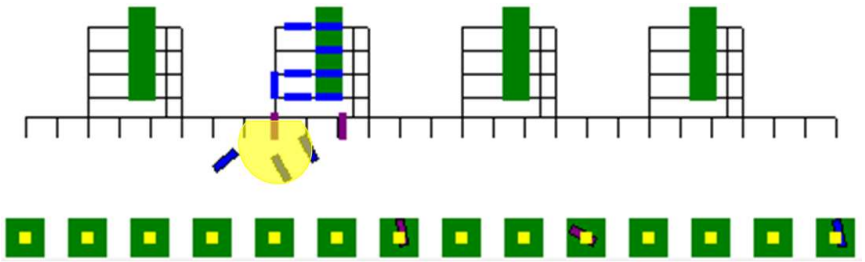


Figure 35 - Hold positions for routing transfer point in mixed routing.

Figure 35 shows the example for QC2, the yellow zone. There are two AGVs waiting to be routed to the transfer point.

Hold – when all QC unload spots are occupied

An AGV receives an unloading spot when it is at the routing transfer point. There, a function determines which unload spot can be assigned or which waiting spot is available (next to the unload spot). When all (waiting) spots are occupied, the AGV is held at the routing transfer point.
The unload spot (i.e. node) is thus only claimed when the AGV is actually at this location.

Usage of buffer places

Buffer places are used as parking place where AGVs are idle or as a crossing. All situations and range of chosen buffer places per QC are explained into more detail in Appendix D.

5.4 Conclusions

This chapter presents the three routing types: traditional routing, free-range routing and mixed routing.

Traditional routing requires a topology of paths. It is chosen to implement a combination of one-directional (in x-direction) and bidirectional (in y-direction) paths. The path topology is composed of nodes and arcs between nodes. Routing of the AGV goes from node to node. The nodes are controlled by semaphores, each node has a capacity of 1 and thus can be claimed by one AGV simultaneously. Routing goes incrementally, each iteration the shortest available path towards destination is searched via a Dijkstra algorithm and the first node on that route is chosen as temporary destination. Once arrived, a new route is searched.

The PFM is chosen as routing method for free-range routing. The PFM is a reactive method that does not plan. Each AGV is represented as a circle which consists of the radius (the actual AGV) and spread (the environment to take into account). The destination has an attractive force while each obstacle has repulsive forces. Repulsive forces are stronger when the obstacle is near. The resultant of the repulsive forces and the attractive force determine the speed and direction of the AGV. The potential field is constructed each second and thus direction and speed are updated each second.

Mixed routing is a combination of the previously presented traditional routing and free-range routing. The latter is chosen as basis and a path topology is implemented near the QCs. The paths along the QC unload spots are one-directional to reduce complexity. When travelling towards a QC, the AGV must visit a routing transfer point where the routing type is switched from free-range towards traditional routing. One such point is available for each QC. When travelling from QC towards ASC, multiple routing transfer points can be used.

Chapter 6 - Verification and validation

Before the model can be used for experiments, the model itself should be tested to see if it does what it is intended to do. This testing requires on the one hand verification, “is the model right?”, and on the other hand validation, “is it the right model?”.
The terminal model including traditional routing is verified and validated in Chapter 6.1. Free-range and mixed routing are verified and validated in Chapters 6.2 and 6.3, respectively.

6.1 Terminal model including traditional routing

This section verifies and validates the terminal model (Chapter 4) and traditional routing (Chapter 5.1).

6.1.1 Verification

Verification is based on the situation that AGVs must travel back to the closest buffer place before they can be called for a new job. It is assumed that there are 40 AGVs present in the system. Traffic control is off in the simulation, thus driving times will not be delayed due to congestions. The simulation stops when 50000 seconds are passed. Container inter arrival time is modeled by a Poisson distribution with an average of 60 seconds.

The following verification methods are applied. Results are briefly stated.

Checklist

A list of general checks that the model must fulfill is presented in Appendix E. The model passed all the checks.

Run-time visualization

Four graphs are chosen as output of the simulation model. The amount of handled containers (graph 1) increases linearly in time, which is expected. Waiting and active elements (graph 2 and 4) reach a steady state, also expected. Travel time delay is zero which is expected as traffic control is off. The graphs are shown in Appendix E.
An animation shows the location of the AGVs. All AGVs travel in the correct order (buffer -> ASC -> QC -> buffer). Containers pop up at the ASCs and disappear once they are loaded onto the AGV.

Test run verification

A seed independence check and a consistency check is done. The first check is to ensure that the chosen seeds produce results in the same range as other seeds would do. The second check should show that when the capacity of a system is doubled, its utilization is halved or vice versa.
Both checks showed no irregularities. See Appendix E for the results.

Verification with analytical results

Queuing theory models are mathematical models of the system with which the simulation model can be verified. A single server model should be used since each crane has its own queue. Arrival time is exponentially distributed; service times of the QC and ASC are assumed deterministic. A M/D/1 model is used to evaluate waiting times at the ASCs and the QCs.
The following test set is used for the mathematical models:

Table 10 - Verification input parameters.

Parameter	Value	Unit
Number of QCs	4	-
Number of ASCs	14	-
Number of AGVs	40	-
ASC container retrieval time	200	s
ASC load time	10	s
QC unload time	18	s
QC put away time	120	s
AGV speed	4	m/s
Container inter arrival time	60	s
Simulation time	50000	s

Other terminal model inputs are shown in Table 7, acceleration and decelerations are not assumed. More information on (the results of) the M/D/1 model is provided in Appendix E.

The following figure shows the entire process of interaction:

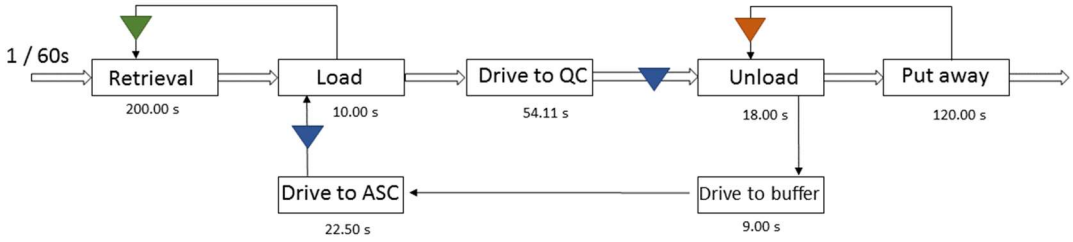


Figure 36 - Interaction between elements in the container handling process.

Each block in the figure represents a function to fulfill, the mean time it takes is shown below the block. The triangles represent waiting times. Waiting times are calculated with the M/D/1 model and are: waiting time of container in stack and waiting time of AGV with container at QC.

Each element (container, ASC, QC and AGV) has a cycle time. The components of a cycle time consist of deterministic data, waiting time data from M/D/1 model or variable data (travel time depends on distance). Each element cycle and its components are presented in Appendix E.
The following table shows the cycle times per element and its utilization. The container flows through the system and thus has no utilization.

Table 11 - Results of analytical and simulation models per element, traditional routing.

	Model	Cycle time [s]	Utilization excluding (including) queue time [-]
Container	M/D/1	530.51	-
	Simulation	541.81	-
ASC	M/D/1	245.00	0.33 (0.39)
	Simulation	244.15	0.33 (0.39)
QC	M/D/1	231.40	0.77 (1.29)
	Simulation	237.28	0.77 (1.32)
AGV	M/D/1	384.51	2.52 (8.54)
	Simulation	396.63	2.66 (8.31)

The calculations of cycle times and discussion of results is provided in Appendix E. The main remarks are now briefly discussed.

The cycle times show good agreement between analytical result and simulation result. The main differences are the drive time to QC and waiting time at the QCs. The former can be explained with the fact that in the analytical model only the lowest unload spots are assumed, the simulation model uses all four unloading spots. This results in a maximum additional drive time of 9 seconds.
The waiting times at the QCs is ±6 seconds higher than the analytical result. The cause is found in the interarrival time of containers. Each ASC should receive a container each 840 seconds, but it appears that this happens each 800 seconds. This reduces the interarrival time at each QC with 11 seconds and thus an increased waiting time as a consequence.

Hand event trace

20 containers and assigned AGVs are traced by hand to evaluate their cycle times. 20 containers that are generated after 35000 seconds are traced. The results from the container process and AGV process are discussed in Appendix E. A brief summary is now given.

The container and AGV process both show small differences with the analytical results. The main cause is the average waiting time at the QC. The analytical model calculated a waiting time of 93.4 seconds, the average waiting time of the 20 containers is only 35.9 seconds. 14 out of 20 containers are handled immediately or within 5 seconds. It is unlikely that this also happens during longer runs, as does also the deterministic long-term run shows. It might be an result from the large interarrival times, the average interarrival time is 82.05 seconds, while the input interarrival time is 60 seconds.
Other work times show good agreement.

Conclusion

Overall it may be concluded that the model performs well. Some minor deviations are found in the analytical model verification, but these are within reasonable limits.

6.1.2 Validation

Now that the model is proved to execute the intended idea, the model must be tested to see if the right model is developed.

Validation can be done for the input parameters, the model assumptions and results of the model.

The terminal model is unique because it is not based on a real-life example, it is developed from scratch. This makes it hard to validate the model as there are no KPIs to compare with. The assumed input data is based on data found in literature and should thus be in correct of magnitude.
The interaction of processes is determined by knowledge gained during this master and in specific the container terminal example as worked out by Veeke and Ottjes during a lecture on discrete event simulation [59]. Some assumptions are however made during development based on logical reasoning. Tracing the containers, AGVs, ASCs and QCs show the correct sequences they should follow, thus the model has face validity.

Two studies in literature are used to do an attempt of validation: one study presents real-life terminal data, the other has a somewhat comparable routing topology. These two studies can show whether or not the terminal model and traditional routing topology produce correct results.

Validation of terminal model

Liu et al. [60] compared the influence of yard layout on terminal performance. Their model is validated with data from the Norfolk International Terminal (NIT), a manually operated terminal. This data is used to validate the terminal model in this report.

Detailed validation is presented in Appendix E. The main findings are discussed here.

The terminal performance was 1.4 times higher than the performance at NIT. This difference is most likely caused by a wrong ASC cycle time assumption, also the layout can be of influence. Given the little details that were given, no conclusion could be drawn from this result.

Liu et al. do experiments with their validated model. The results from the loading process is used to validate the terminal model.
The results as interpreted and obtained with the terminal model are shown in Table 12.

Table 12 - Comparison of results by Liu et al. [60] *and terminal model.*

Number of vehicles	Results from Liu et al.	Results terminal model	Difference with Liu et al. [%]
6	35	33.2	-0.05
9	45	46.5	0.03
12	60	58.4	-0.03
15	65	64.8	-0.00
18	73	67.1	-0.08
21	75	66.9	-0.11

These results show more agreement. The results start to diverge when the number of AGVs increases. The terminal model has a maximum performance of 67 moves/hr, the model of Liu et al. come closer to the QC capacity of 75 moves/hr.

The deviation in results can be due to a number of differences, for example there are differences in the implemented routing topology, routing control and ASC cycle time. But despite those facts, the overall increase of terminal performance as more AGVs are in operation shows good agreement.

Validation of routing topology

To validate the routing topology and performance, the study of Bae et al. [9] is used. Their topology is shown in Figure 10.

Table 13 shows the results of Bae et al. and the terminal model.

Table 13 - Comparison of results by Bae et al. and terminal model.

Number of AGVs	Results from Bae et al. [9]	Results terminal model	Difference with Bae et al. [%]
8	19.5	21.5	0.10
12	23.5	25.1	0.07
20	28.0	25.2	-0.10
28	29.5	25.0	-0.15
44	30.5	25.9	-0.15

The results show that the terminal model reaches a performance of ±25 moves/hr/QC with 12 AGVs, the results from Bae et al show more increase in performance as more AGVs are added. The performance with 44 AGVs differs by 5.4 moves/hr/QC. This can be explained by different reasons:

- The topology of Bae et al is completely bidirectional, the terminal model has one-directional paths in x-direction. Distances between nodes differ.
- The topology of Bae et al. allow 4 AGVs to be waiting at an ASC, the terminal model has one unload position per ASC.
- The terminal dimensions differ: the terminal of Bae et al. is less wide (- 123 m) but has more height (+ 30m).
- The ASC cycle times is a rough interpretation of their implementation, it is impossible to implement the correct cycle times as the figure is too unclear.

Despite the performance differences, the models show reasonable agreement.

Conclusion

The terminal model is a unique model and thus there exist no historical data for this model. This makes it hard to validate the terminal model. Data from the Norfolk International Terminal is used to test the terminal model. Although there exist differences in topology, the terminal shows reasonable agreement. The routing topology is compared to a comparable topology, the results differ slightly but the overall increase in terminal performance as more AGVs are employed shows good agreement.

6.2 Free-range routing

This section verifies and validates free-range routing.

6.2.1 Verification

The following table shows the cycle times per element and its utilization. The container flows through the system and thus has no utilization.

Table 14 - Results of analytical and simulation models per element, free-range routing.

	Model	Cycle time [s]	Utilization excluding (including) queue time [-]
Container	M/D/1	519.92	-
	Simulation	537.66	-
ASC	M/D/1	245.00	0.33 (0.39)
	Simulation	249.69	0.33 (0.40)
QC	M/D/1	231.40	0.77 (1.29)
	Simulation	245.70	0.77 (1.37)
AGV	M/D/1	374.92	2.22 (8.33)
	Simulation	389.90	2.19 (8.66)

The calculations of cycle times and a discussion of results is provided in Appendix E. The main remarks are now briefly discussed.

The work times of all elements show good agreement between the analytical model and the simulation. The main difference is found in the waiting time at the QCs (part of the container process, QCs and AGV process). A steady state of AGV waiting time at the QC is reached after 30000 seconds. It is seen that some QCs have a sudden increase in waiting time, many containers are generated in a short time frame which have the same QC destination. This is due to the seed its non-uniformity in small timeframes.

Hand event trace

20 containers and assigned AGVs are traced by hand to evaluate their cycle times, these containers appear after 35000 seconds. The system has 27 AGVs. The results from the container process and AGV process are discussed in Appendix E. A brief summary is now given.

The container and AGV process both show a similar deviation from their analytical results: the traced cycle times are 108 seconds faster. The work and drive times show good agreement, the waiting times at ASC and QC not. A container waits 27 seconds less at the ASCs, probably because the interarrival time is larger (82 seconds instead of 60 seconds). An AGV at a QC has a waiting time of only 22.0 seconds, while the analytical model calculated 93.4 seconds. The cause may be found in the larger interarrival time of containers.
The drive times can differ slightly, this can be due to the fact that the analytical model does not assume decelerations or due to the “arrival zone”. Each destination has an area of +-3 meters in which the AGV is assumed to be arrived.

Despite the small differences of simulation results compared to analytical results, it may be concluded that free-range routing performs as intended and thus can be assumed verified.

6.2.2 Validation

The terminal model is validated in Chapter 6.1.2. Free-range routing must be validated, but once again the problem arises that it is impossible to validate with historical data as there is no real-life case that uses free-ranging AGVs. The concept of free-ranging AGVs on a container terminal is not often studied in literature.

Choi and Zhu [14] did similar research: they implemented a potential field controller in AGVs operating at a container terminal. Unfortunately, they did not study the effect on routing or terminal performance.

Duinkerken and Lodewijks [61] implemented a DEFT controller, see Chapter 3.2. The routing performance of a mesh topology is compared to the performance of the DEFT controller. Their terminal model has different dimensions and they assume 6 QCs and 13 ASCs, the results show that the DEFT controller handles 1.5 times more containers/hour than when a mesh topology is used.
A similar experiment is performed with the terminal model; the difference between traditional routing and free-range routing is measured. 25 AGVs operate in the terminal model. Unfortunately, it appeared that major deadlocks occur using free-ranging and thus the results cannot be used for validation. This does show the benefit of a planner over a reactive method such as DEFT and the potential field method, respectively.

In terms of logical sequences of routing: the execution of processes to determine direction and speed of the AGV is done in the correct order and amounts. The potential field method has face validity.
The parameters implemented for the potential field method are determined empirically, it is impossible to validate them.

6.3 Mixed routing

Mixed routing uses both traditional routing and free-range routing. Traditional routing is applied to guide AGV in the vicinity of the QC.

6.3.1 Verification

This section verifies the element processes of the model by comparing simulation results with its analytical counterpart. This is done for each element cycle. Also a hand trace of 20 containers in the steady state is done. 23 AGVs are assumed to be in operation.
The following table shows the cycle times per element and its utilization. The container flows through the system and thus has no utilization.

Table 15 - Results of analytical and simulation model per element, mixed routing.

	Model	Cycle time [s]	Utilization excluding (including) queue time [-]
Container	M/D/1	532.63	-
	Simulation	555.88	-
ASC	M/D/1	245.00	0.33 (0.39)
	Simulation	250.69	0.33 (0.40)
QC	M/D/1	231.40	0.77 (1.29)
	Simulation	247.28	0.77 (1.37)
AGV	M/D/1	391.13	2.64 (8.69)
	Simulation	408.65	2.61 (9.08)

All processes show good agreement between the analytical result and its corresponding simulation result. Two main differences are noticed: the first is the container waiting time in stack, this is 5 seconds longer in the simulation model than in the analytical model. It is caused by a faster arrival rate (1/800 seconds versus 1/840 seconds, simulation model versus analytical model, respectively). The second difference is the waiting time of AGVs at QCs, the simulation model has a waiting time of 107.3 seconds while the analytical result has a waiting time of 93.4 seconds. The average waiting time per QC does not reach a steady state and that increases the waiting time, a steady state is assumed in the analytical model.

Hand event trace

20 containers and corresponding AGVs are traced for one container or AGV cycle. The results of the hand trace are presented in Appendix E, a brief summery is given next.

The average container cycle time differs only 1 second from the analytical cycle time, but some differences are noted in the cycle components. Waiting time of container in stack is half the analytical result, waiting time at QC is 13 seconds above the analytical result. The exact causes are unknown, probably the combination of interarrival times and stack + QC choices are of influence. The container retrieval time differs 0.3 seconds from its input mean, the put away time differs 12 seconds.
The average AGV cycle time is 399.38 seconds, compared to 391.13 from the analytical result. The drive times show good agreement. Waiting time of the AGV at the stack is 11 seconds longer than the analytical model calculated, a clear cause cannot be found. Most likely the interarrival time of previous containers at the stacks are of influence.
Both cycle times and its components show quite good agreement with the analytical results.

Overall, the model performed well in both verification methods and thus can be assumed correct.

6.3.2 Validation

The mixed model is not studied in literature, let alone implemented in real-life. Validation based on historical data of this routing type is therefore impossible.

Mixed routing consists of traditional routing and free-range routing, both were validated in Chapter 6.1.2 and 6.2.2 respectively. Therefore the mixed model can also be assumed to generate correct results.

The use of routing transfer points is new, AGVs are traced to make sure the correct procedures are executed. This appears to be the case and thus this routing type is assumed to have face validity.

6.4 Conclusions

Verification and validation of the terminal model and the three routing types are performed in this chapter. Verification is needed to make sure that the model is right, validation makes sure that the right model is developed.

The terminal model and traditional routing are verified and validated together. A checklist check, run-time visualization, test run verification, verification with analytical results and hand trace are done. Analytical results were obtained from a deterministic queueing theory model: M/D/1. The QC and ASC processes were each analyzed with the M/D/1 model. The results are used for the AGV process. Analytical results are compared to deterministic simulation results. All verification methods showed good agreement.
Validation of the terminal model and traditional routing is tough, since the model is not based on a real-life situation. The logistical processes are executed in the correct sequences. Terminal validation based on data from the Norfolk International Terminal is done as well as routing validation using a study of Bae et al. The result differ slightly, assumptions of the inputs had to be made, but overall showed reasonable agreement.

Free-range routing is verified with a comparison of analytical results and a hand trace of 20 containers. Despite some small deviations, the results showed good agreement. Validation of this routing type is impossible as no similar studies are performed. Tracing AGVs showed that the correct sequences are executed.

Mixed routing is also verified with a comparison of analytical results and a hand trace. Both methods showed good agreement. Validation is impossible as this type of routing is new to literature. The tracing of AGVs showed that the correct sequence of procedures is executed.

Chapter 7 – Experimental plan

The goal of the experiments is to compare three routing types on costs, terminal performance, routing performance and computational times. The three routing types are traditional, free-range and mixed routing. These routing types are implemented in a terminal model (Chapter 4). The three different routing types are explained in Chapter 5.

Before the experiments are explained in more detail, the run length and number of replications must be determined to obtain correct statistical results.

Run length

The run length, or simulation time, must be sufficient to eliminate the influence of the warm up period. The transport capacity and amount of driving AGVs reach a steady state after +-5000 seconds, see Figure 37. The figure shows the result for 15 AGVs, but similar output is generated with less or more AGVs.

It is decided to assume a run length of 50000 seconds. Results from the warm up period are included, but are negligible over the total run length.

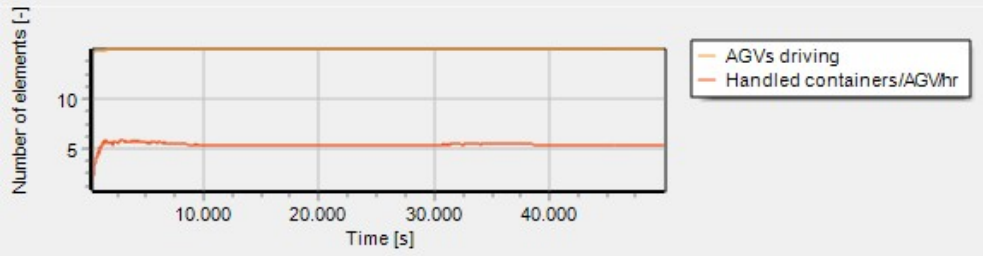


Figure 37 - Warm up and steady state of simulation run.

Number of replications

The number of replications must be determined to ensure that the results are within the chosen confidence interval. After the number of replications is determined, each experiment is performed that many times.

The number of replications is determined with the confidence interval method [62]. A desired precision is chosen. The minimum amount of runs is 3. When the calculated precision is lower than the desired precision and the next few replications are also within this bound, the amount of replications is found. Else, replications continue until all conditions are fulfilled.
The desired precision is chosen to be at least 5%. The following table shows the minimal amount of replications per method and routing type. The determination is done with 4, 15 and 23 AGVs and other standard input values (Table 7).

Table 16 - Required number of replication per routing type.

		Confidence interval method
Traditional routing	4	3
	15	3
	23	3
Free-range routing	4	3
	15	3
	23	3
Mixed routing	4	3
	15	3
	23	3

All results are and remain, within the 5% confidence bound with the minimal amount of three runs. Each experiment will be performed 3 times.

Fixed input parameters

There are three types of input parameters needed: topology, operational and costs related. Topology concerns the dimensions of the terminal. The operational parameters specify the characteristics of the equipment in the terminal model. The costs related parameters are used to convert outputs to a costs parameter. Table 17 shows the input parameters that are fixed.

Table 17 – Fixed input parameters of experiments.

	Parameter	Value	Unit
Topology	Size of terminal (w x h)	416 x 108	m
	d_{ASC}	32	m
	d_{QC}	96	m
	d_{buffer}	Depends on routing type	m
	$d_{unload\ spot}$	12	m
Operational	Speed of AGV	4	m/s
	Acceleration of AGV*	1	m/s^2
	Deceleration of AGV*	1	m/s^2
	Load time of AGV at ASC	10	s
	Container retrieval time of ASC	Empirical distribution[4]	s
	Simulation time	50000	s
Costs	Purchase of 1 AGV	282000	€
	Operational costs of 1 AGV	4.50	€

*Only applies to traditional routing. The PFM determines acceleration or deceleration per iteration.

Besides these fixed parameters, there are also parameters that are varied in the experiments. These are shown in Table 19 after the experiments are explained.

[4] Distribution as reported by Vis and Harika [18].

7.1 Experiment setups

The three routing types are tested in an identical terminal model to determine which type performs best. This is based on the following experiment:

- Lifetime costs of AGV system
  The costs of the AGV system are determined at different QC capacities while a terminal performance of 95% must be reached. The minimum amount of AGVs that reach this threshold is converted to AGV system costs. Costs of the AGV system consist of purchase costs and operational costs. Costs on the physical implementation of topology is not assumed. It is assumed that the lifespan of an AGV is 72.000 hours (10 years * 360 operational days * 20 operational hours/day).
  This experiment is performed for each routing type so that a comparison between the types can be made.

To compose the costs graph, multiple tests are performed in which QC capacity and amount of AGVs are varied. Data from these tests are used to compose graphs of:

1. Terminal performance
   The terminal performance, i.e. handling efficiency, is determined as a function of amounts of AGVs. One graph is generated per QC capacity.
2. Routing performance
   The routing performance, i.e. travel time delay factor, is determined as a function of amount of AGVs. One graph is generated per QC capacity.
3. Computational time
   The computational time of each test is measured. The results corresponding to the costs graph are represented in a graph. But also computational times as a result of amount of AGVs are visualized, per QC capacity.

Number of AGVs

First, the amount of AGVs in the terminal is varied. A minimum four AGVs are chosen, equal to the amount of QCs in the terminal model. The maximum amount of AGVs is determined by the mixed routing strategy that has a maximum of 23 buffer places, that were needed for the initialization.

The number of AGVs is increased from 4 to 23 with intermediate steps of 1.

QC capacity

Second, the QC capacity is varied. As base case, a capacity of 30 moves/hr/QC is chosen. This is close to the realized capacity at the ECT Delta Terminal [16]. The QC capacity is increased until 70 moves/hr/QC, which is close to the claimed capacity of the manufacturer.

QC capacities 30, 40, 50, 60 and 70 moves/hr/QC will be tested. In the case that the 95% threshold cannot be reached with the maximum amount of AGVs, intermediate capacities will be tested.

A cycle of a QC consists of unloading a container from an AGV at a QC unload point, traveling with the container to the vessel, placing the container in the vessel and travelling back to the unload point. The unloading time of a container from an AGV is a deterministic value, and is taken as ±13% of the total cycle time. The remaining part is denoted with the term “put away time of QC” and is modeled as a normal distribution. A mean and standard deviation are necessary as input to the model. The standard

deviation is taken as 1/4th of the mean put away time. These relations are based on data chosen by Veeke and Ottjes [59]. To clarify:

$$Cycle\ time\ [s] = \frac{3600}{QC\ capacity} \tag{6.1}$$

$$Cycle\ time\ [s] = \ mean\ put\ away\ time + unload\ AGV\ time \tag{6.2}$$

The following QC inputs are used in the experiments:

Table 18 - QC capacity inputs of the experiments.

Capacity [moves/hr]	Put away time, mean [s]	Put away time, deviation [s]	Unload time [s]
30	105	26.0	15
40	78	19.5	12
50	63	15.8	9
60	52	13.0	8
70	45	11.0	7

The varying input parameters are:

Table 19 - Varying input parameters of experiments.

	Parameter	Value	Unit
Operational	Amount of AGVs	4 - 23	-
	Put away time of QC	N(μ,σ), see Table 18	s
	Unload time of AGV at QC	±13% of mean “Put away time of QC”	s

7.2 Sensitivity analysis

A sensitivity analysis is performed to test the relation between uncertainty in output of the model and uncertainty in input. This is done by running the model with some alternative input variables. Insight into the robustness of the model is gained, but also more understanding in the relation between input and output variables.

Both sensitivity of the terminal model and routing types is tested. The sensitivity of the terminal model is tested by changing the container retrieval times of the ASCs and changing AGV speed. The sensitivity of the routing types is tested by changing obstacle-avoidance parameters. Since traditional routing does not have parameters that could be varied, this is only applied to free-ranging.

The reference case has input parameters as shown in Table 17. There are 16 AGVs assumed to be in operation.

Container retrieval time of ASC

The increase and decrease of retrieval time of the ASCs is studied on the terminal and routing performance. The former is represented with handled container efficiency, the latter with travel delay. The grey column represents the reference case.

Table 20 - Influence of container retrieval time on terminal and routing performance.

		Container retrieval time [s]				
		50	100	200	300	400
Traditional routing	Handling efficiency [-]	0.987	0.992	1.000	0.988	0.987
	Travel delay [-]	1.310	1.300	1.290	1.310	1.290
Free-range routing	Handling efficiency [-]	0.996	0.993	0.994	0.989	0.987
	Travel delay [-]	2.840	2.850	2.840	2.830	2.800
Mixed routing	Handling efficiency [-]	0.996	0.993	0.993	0.988	0.991
	Travel delay [-]	1.110	1.100	1.120	1.110	1.110

The container retrieval time does not have much influence on the results of all three routing types. The results differ a bit, but no relation between container retrieval time and performance can be found. This effect can be explained by the fact that the QC cycle times are too long and an AGV queue built up at each QC. Therefore, the differences in container retrieval times that are tested are too small to obtain a difference in performance. It is expected that faster container retrieval times result in longer queues at the QC and more traffic towards the ASCs. A slower container retrieval time reduces QC queues and traffic.

Overall, changing container retrieval times at the ASCs (in the range as is tested) shows little sensitivity to the performance.

AGV speed

The AGV speed is varied to study its effect on terminal and routing performance. The grey column represents the reference case.

Table 21 - Influence of AGV speed on terminal and routing performance.

		AGV speed [m/s]				
		1	2	4	6	8
Traditional routing	Handling efficiency [-]	0.821	0.987	0.995	0.993	0.993
	Travel delay [-]	1.170	1.160	1.290	1.500	1.750
Free-range routing	Handling efficiency [-]	0.893	0.991	0.994	0.989	0.993
	Travel delay [-]	1.430	1.730	2.040	3.870	2.480
Mixed routing	Handling efficiency [-]	0.382	0.985	0.993	0.993	0.985
	Travel delay [-]	2.420	1.240	1.112	1.070	1.060

The three highest AGV speeds show similar handling efficiency in all three routing types. This is because the QCs are running constantly and earlier arrived containers cannot be handled sooner. A low AGV speed decreases the handling efficiency, especially in mixed routing with a speed of 1 m/s.
Travel delay increases when speed increases for both traditional and free-range routing, mixed routing shows a reverse effect.
In mixed routing, delay increases as speed decreases. The traffic density increases in the free-ranging part of the terminal. Avoiding obstacles take longer as more iterations are needed to leave an obstacle. This effect is not seen in the second routing type, free-range routing, because the traffic is less dense in this type.

Free-range, obstacle avoidance parameters

The potential field method is implemented to represent the free-range routing type. This method senses its immediate environment and a travel direction is determined based on all obstacles in that environment and its destination. Environment is a circle with a constant radius and is the sum of radius and spread. Each AGV has a radius and no obstacle may be closer than that distance to the AGV. When an AGV is near a QC, the radius and environment are decreased to be able to reach an unloading spot when other AGVs are also waiting beneath that QC. This reduction is also applied when the AGV approaches a static obstacle (non-moving AGV). The reduced spread and AGV radius are changed for this sensitivity analysis.

The grey column represents the implemented case as in the terminal model. Each configuration ran 20 times.

Table 22 - Influence of routing obstacle-avoidance parameters on terminal and routing performance, 16 AGVs.

		AGV radius + spread [m]				
		8 + 12	10 + 8	10 + 10	12 + 12	16 + 20
Free-range routing	Handling efficiency [-]	0.994	0.994	0.995	0.940	0.038
	Travel delay [-]	2.340	2.300	2.840	4.120	3.910
	Collisions (σ)	0.400 (0.737)	0.333 (0.617)	0.200 (0.561)	0.000 (0.000)	0.000 (0.000)

The influence is also tested with the maximum amount of AGVs (23):

Table 23 - Influence of routing obstacle-avoidance parameters on terminal and routing performance, 23 AGVs.

		AGV radius + spread [m]				
		8 + 12	10 + 8	10 + 10	12 + 12	16 + 20
Free-range routing	Handling efficiency [-]	0.991	0.992	0.993	0.922	0.023
	Travel delay [-]	3.200	3.290	3.800	5.980	3.068
	Collisions (σ)	5.400 (8.659)	5.470 (6.198)	2.130 (2.588)	0.400 (0.632)	0.000 (0.000)

Both tables show that when the radius is increased, it is harder to reach the QC unloading spot and thus the amount of handled containers decreases. In the case of r=16 m and s = 20 m, the handling efficiency is extremely low because the radius is larger than the distance between unloading spots (12m) and thus deadlocks occur when multiple AGVs must unload at one QC.

A similar environment (spread + radius) yield comparable results, cases 8+12, 10+8 and 10+10. The delay is higher for the one with larger radius and spread, i.e. 10+10, because the AGV brake intensity is higher in this case (obstacle is closer to non-entering radius of AGV).

Furthermore, this analysis show the sensitiveness to collisions. Especially with 23 AGVs, collisions occur easily when the radius and spread are reduced. Only a large radius and spread are collision-free, case 16+20. Therefore, an extra collision-avoidance method should be applied near the QCs where the sensing environment must be reduced.

7.3 Conclusions

The experimental plan is presented in this chapter. The goal of the experiments is to determine the costs of the AGV system per routing type. Costs are a summation of fixed and variable costs. A terminal performance of 95% must be reached, different QC capacities are tested. Besides the costs, terminal and routing performance and computational times will be analyzed.

The number of AGVs is varied between 4 and 23 with intermediate steps of 1. QC capacities are varied from 30 to 70 moves/hr/QC with intermediate steps of 10.
The run length is determined and chosen to be 50000 seconds. Results from the warmup period, 4000 seconds, are included but negligible. Number of replications is determined with the confidence interval method. Three replications per experiment are sufficient to remain within 5% confidence.

A sensitivity analysis is performed to test the relation between uncertainty in output of the model and uncertainty in input. Alternative inputs are tested and the results are studied. Sensitivity is tested on the container retrieval time of the ASCs, AGV speed and free-range parameters.
Container retrieval time showed little sensitivity to the outputs of all routing types.
AGV speed had an influence on the outputs, especially in terms of delay. Delay increases as speed is higher, except mixed routing showed an opposite effect. Handling efficiency of the routing types showed few consistency. Overall an increase in performance is seen as speed is higher, but a turning point is seen when speed becomes ≥ 6 m/s.
With free-range parameters is meant the no-go area that represents the actual AGV (“radius”) and environment (“spread” + “radius”) of the AGV. As radius and spread are increased, terminal performance decreases. It is harder for the AGV to reach its destination as the radius prevents other AGVs to come close. Collisions decrease though.

Chapter 8 – Results and discussion

This chapter presents and discusses the results of the performed experiments. The three routing types are compared to find the best performing type under several conditions. Both the terminal and routing performance of the model is studied. Thereby are QC capacity and number of AGVs varied. Terminal performance is for the terminal operator the most important KPI, but routing performance shows more insight in the AGV behavior.
This chapter first presents the terminal performance as a result of costs. Then follows a more detailed analysis of the terminal performance in different situations. Hereafter an analysis of the routing performance is presented. Last, an insight in computational times is presented.

8.1 Costs of the AGV system

The minimum number of AGVs needed to cross the 95% handling efficiency threshold is shown for different QC capacities (Figure 38).

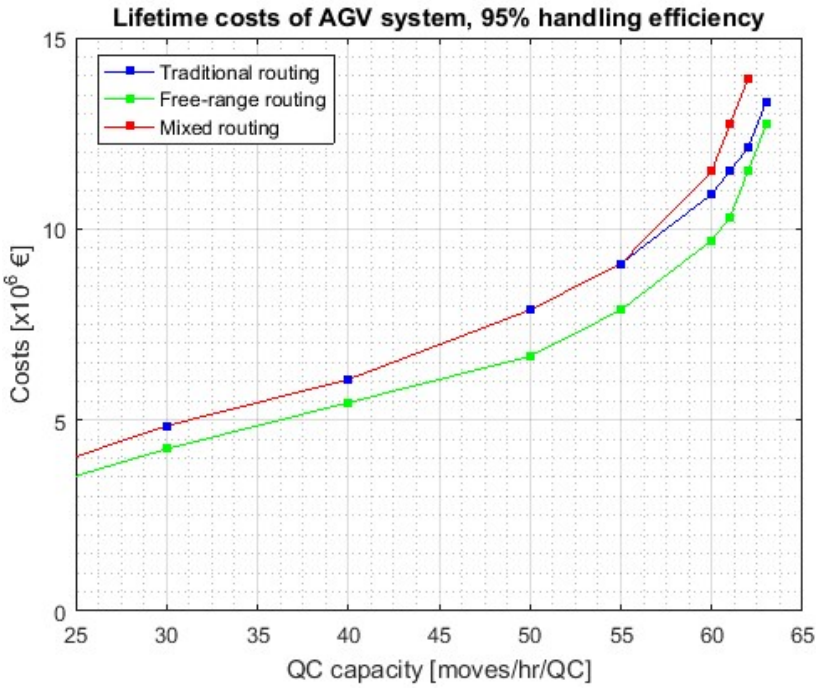


Figure 38 - Lifetime costs of the AGV system.

Note that the results for QC capacity 70 moves/hr/QC are not shown, this is because the 95% threshold could not be reached with the maximum amount of AGVs (23). Intermediate capacities are tested until this threshold could not be reached. Table 24 shows the required amount of AGVs per routing type and QC capacity.

For QC capacities up to and including 50 moves/hr/QC, costs increase proportionally with QC capacity. This is seen for all routing types. Free-range routing has lowest costs. Traditional and mixed routing require similar amounts of AGVs and thus costs, the costs are 1.15-1.18 times those of free-range.
At low QC capacities, few AGVs are required. Few to no congestion occurs and the length of the travel distances are leading for the results. Free-range routing has shortest paths and therefore its costs are significantly less.

Costs increase rapidly when QC capacity exceeds 50 moves/hr/QC, the differences between routing types are revealed. Free-range routing remains the cheapest type for all QC capacities, but more and more AGVs are needed per QC capacity increase to retain the 95% performance. As more AGVs are in operation, congestion occurs. The figure shows that free-ranging is sensitive to congestion and that this affects terminal performance and thus costs.

Traditional routing and mixed routing split after QC capacity exceeds 55 moves/hr/QC. Mixed routing is the most expensive choice, its costs increase more rapidly. Mixed routing has a steeper slope than traditional routing in each QC capacity interval. This is due to the congestion sensitivity of the free-range component in mixed routing.
Traditional routing has costs between mixed and free-range routing when QC capacity exceeds 55 moves/hr/QC. Slope increases only slightly between 60 and 62 moves/hr/QC compared to free-range and mixed routing, resulting in an approach towards free-range routing. Traditional routing appears to be flexible enough to keep congestions and detours low. As a result, its longer paths lengths become competitive with the shorter, but congested, paths of free-range routing.

It is surprising that the costs of mixed routing are highest of all routing types, especially in the higher QC capacity regions where mixed routing was expected to perform well. It clearly does not make use of the benefits of traditional and free-range routing. More in-depth analysis of the terminal and routing performances are presented next to explain this trend.

Computational times belonging to the results in Figure 38 are presented in Chapter 8.4.

Table 24 - Number of AGVs required to reach 95% terminal performance.

QC capacity [moves/hr/QC]	Traditional routing	Free-range routing	Mixed routing
30	8	7	8
40	10	9	10
50	13	11	13
60	18	16	19
61	19	17	21
62	20	19	23
63	22	21	-

8.2 Terminal performance

Terminal performance is quantified as container handling efficiency at the QCs. QCs are expensive equipment and the terminal operator is focused to reduce QC idle time as much as possible. AGVs must deliver a continuous infeed of containers at the QCs.
Figure 39 shows the terminal performance as a function of number of AGVs for different QC capacities. The figure shows even number of AGVs, Table 54 (Appendix F) shows the corresponding data including uneven numbers of AGVs.

The results show that at the lower QC capacities, few AGVs are needed to reach high performances. As QC capacity increases, more AGVs are needed to maintain high performance. This relation is also seen in Figure 38.
Higher QC capacities require an increase of AGV infeed rate. More AGVs are needed resulting in an increased risk of congestion. As QC capacity is higher, the stochastic infeed rate due to congestion will immediately affect terminal performance. This explains the flattening trend in performance between 18 and 22 AGVs at 70 moves/hr/QC (Figure 39). At 60 moves/hr/QC a continuous increase in performance is seen, QC handling speed and AGV congestions are in good balance such that performance is not affected when more AGVs are in operation. At lower QC capacities a plateau is reached as more AGVs are in operation, this is due to the QC handling speed that becomes the bottleneck of terminal performance.

Routing types

It may be assumed that no delay occurs when 4 AGVs are in operation, this provides an insight into the capabilities of each routing type. Free-range outperforms traditional and mixed routing, mixed routing performs second best. This is expected: free-ranging has shortest paths from any location to destination. Mixed routing has a path topology near the QCs, this increases travel distances and thus times.

Free-range routing reaches highest performances for all QC capacities. Especially when few AGVs are in operation, the differences with other routing types is significant. The fact that the differences with other routing types reduce when many AGVs are in operation shows that free-range routing is sensitive to congestion. The performances of free-ranging AGVs flatten sooner compared to traditional and mixed routing. This can be seen at QC capacities 50 and 70 moves/hr/QC for example, but bear in mind that QC handling speed also plays a role.

Traditional and mixed routing perform quite similar. When there is no congestion (4 AGVs in operation), mixed routing only performs slightly better than traditional routing. This implies that the path topology in mixed routing has too much influence on travel distances. As QC capacities are 50 moves/hr/QC or higher mixed routing is even outperformed by traditional routing when ≥10 AGVs are in operation. This is a surprising result and the routing performance must be analyzed to find the cause (Chapter 8.3).

Decrease in terminal performance due to congestion is seen in all routing types. Free-range routing is most sensitive to congestion. The PFM (free-range routing) has no intelligence on strategic obstacle avoidance and deadlocks are lurking (e.g. congestion). The fact that mixed routing is outperformed by traditional routing is due to congestion incurred by the free-range component. Traditional routing seems to be little affected by congestion, this routing type has multiple route options and the path finding algorithm immediately searches for alternative routes once the shortest route is occupied. The grid offers enough flexibility with little to no increased route distances. This corresponds with Figure 38.

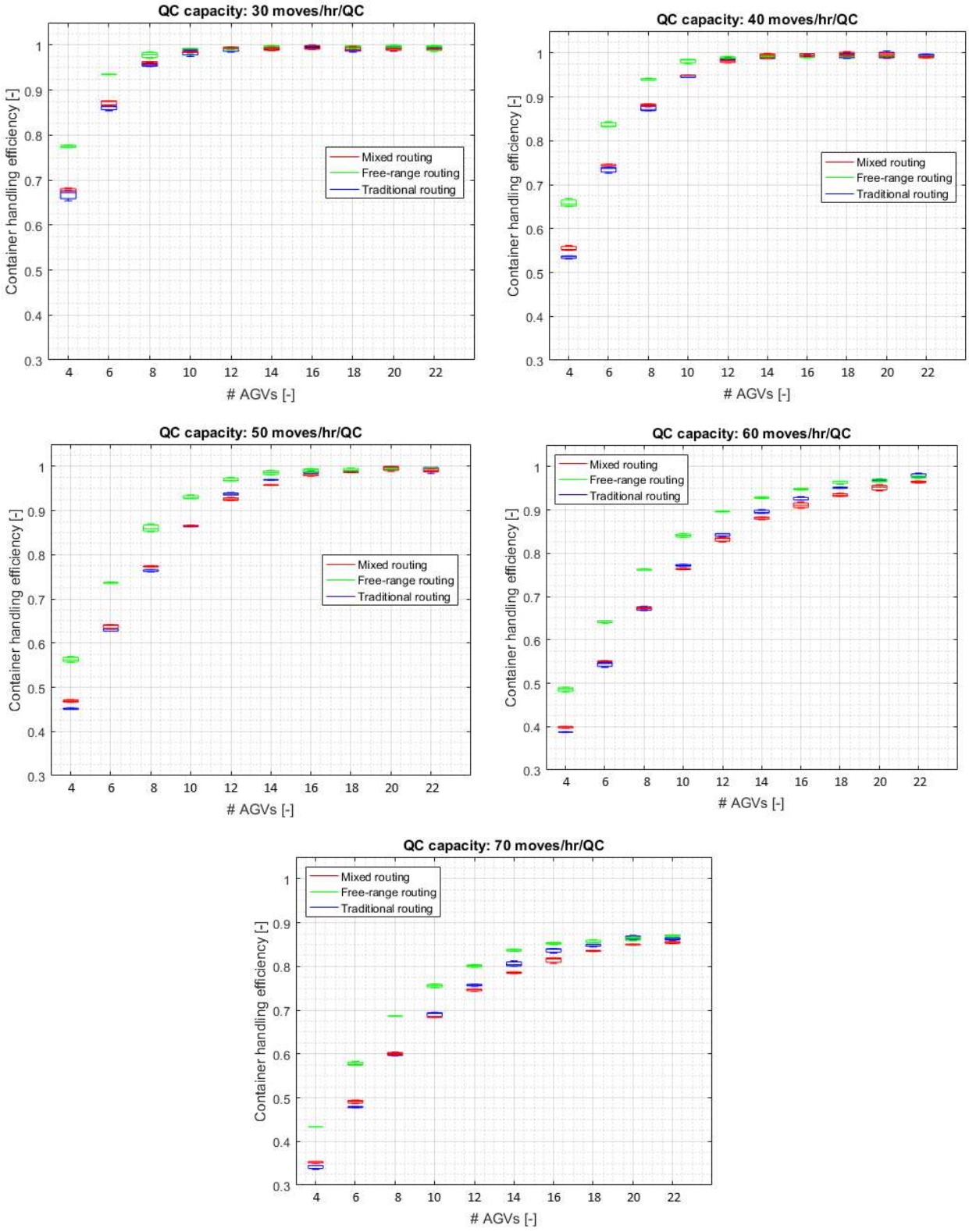


Figure 39 - Terminal performance for different QC capacities.

8.3 Routing performance

The results of routing performance will show more insight in the behavior of the three routing types. Routing performance is quantified as the realized travel time divided by the shortest path travel time. Results are thus a factor so that the three routing types can be compared.

The waiting time at the hold positions (when all QC unload spots are occupied) are not included in this results, see Chapter 5 for more information on the hold positions. The results are purely the realized travel times. Figure 44 shows the routing performance for different QC capacities at even AGV amounts, Table 55 (Appendix F) shows the corresponding data including uneven AGV amounts.

Comparing the results shows that delay increases when more AGV are operation simultaneously. More AGVs mean higher traffic density and thus more detours and/or congestion because the shortest path is not always available. The amount of delay strongly depends on the routing type implemented. Increasing the QC capacity leads to less delay, except for mixed routing. This effect of mixed routing is unexpected. Each routing type is analyzed on their performance.

Routing types

Free-ranging

What immediately stands out is the fact that free-ranging has a higher delay as more AGVs are in operation, and when QC capacity is lowered. More AGVs increase traffic density and thus more obstacles to avoid. The PFM has no intelligence on strategic obstacle avoidance and thus delay can increase fast.

Delay increases as QC capacity decreases, this is because the QC unload spots are hard to reach when other spots are occupied. As QC capacity is lower, sports are occupied for a longer time.

Suppose the AGV has assigned an unload spot between two others that are occupied with an AGV:

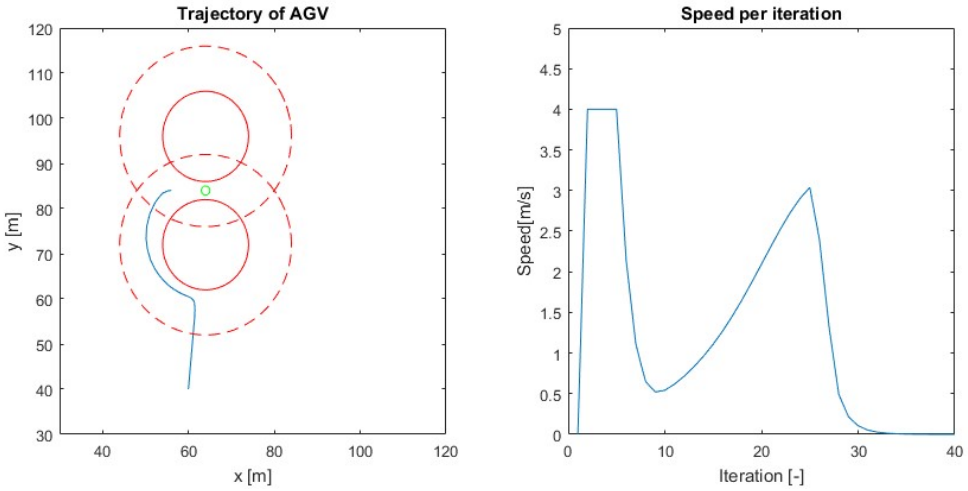


Figure 40 – AGV reaching a QC unload spot between two occupied spots. Left: Trajectory of AGV. Right: Speed per iteration.

Figure 40 shows the trajectory of such a situation and the corresponding speeds per iteration. The figure shows that the spreads of the obstacles (20 m) do slow the AGV down, especially when close to the obstacle its radius (10 m). This is a static case, the two obstacles are idle at their unloading spots.

Now suppose three AGVs simultaneously trying to reach their unload spot. It appears that a deadlock easily occurs in this situation and neither AGV reaches its destination. This deadlock is resolved when a fourth AGV comes closes and activates repulsive forces that resolve the deadlock. Such deadlocks are the cause of the travel delays. Figure 41 shows two deadlock situations.

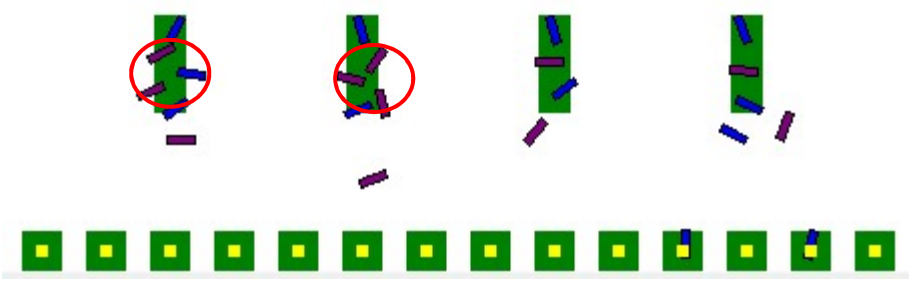


Figure 41 - Deadlocks at QCs (red circles).

Although the delays are high, there is plenty of time to reach an unload spot when the QC capacity is 30 moves/hr/QC. On average, each unload spot is handled once every 480 seconds. When an AGV leaves the QC, this gives new flexibility for an arriving AGV to reach its destination.

When QC capacity increases, the AGVs are more spaced across the terminal because more containers can be handled and thus more transports must be made. The chances decrease that many unload spots are occupied when an AGV arrives. This improves routing flexibility and simplifies travelling towards its destination. Significant delay decreases are seen in Figure 44 when QC capacity increases.

Traditional routing

Traditional routing shows the same delay relations as free-range routing but in lesser amounts. Routing via nodes goes smoother than free-ranging. When an obstacle blocks the preferred path, other routes can be found. The grid offers enough flexibility to reroute.

It is noticed that the hold function (Chapter 5) has some flaws; when an unload spot becomes free, all AGVs assigned to this unload spot start driving (this can be more than one AGV). The first to arrive claims the spot, leaving the others lost. Some go in a new hold, but others keep swarming. This causes the delay in traditional routing.

Also here counts that there is plenty of time to reach the unload spot in the situation of QC capacity 30 moves/hr/QC, thus terminal performance remains high.

Less delay occurs when QC capacity increases. More containers must be transported, spreading the traffic more over the terminal and increasing route options.

Mixed routing

Mixed routing shows opposite results: at low QC capacities almost no delay occurs, while delay increases with increasing QC capacities.

At low QC capacities (30 and 40 moves/hr), many AGVs are waiting at unload spots to be handled. This leaves an almost empty free-ranging zone and thus the ASCs and routing transfer points can be reached

without problems. No delay occurs in the traditional routing zone, because traffic can only head one way. Figure 42 shows the situation as it mostly occurs at QC capacity 30 moves/hr/QC.

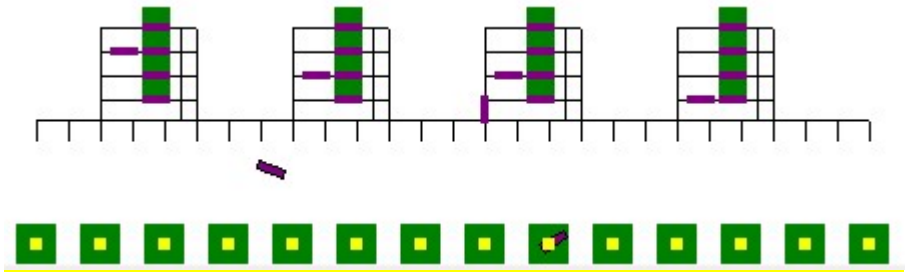


Figure 42 - Snapshot of mixed routing at a QC capacity of 30 moves/hr/QC.

But, delay increases as QC capacity increases. Something one would not expect at first thought. Because the QC capacity is higher, more containers can be handled and thus more traffic drives over the terminal. The free-range zone is quite narrow and that appears to be the bottleneck.

The traffic density is too high for free-ranging to perform well, many obstacles must be avoided. AGVs heading towards the routing transfer point are delayed by AGVs coming from the QC and heading towards an ASC. Quite often situations as in Figure 43 occur, making it hard for an AGV to reach the routing transfer point. Such situations are resolved quickly though.

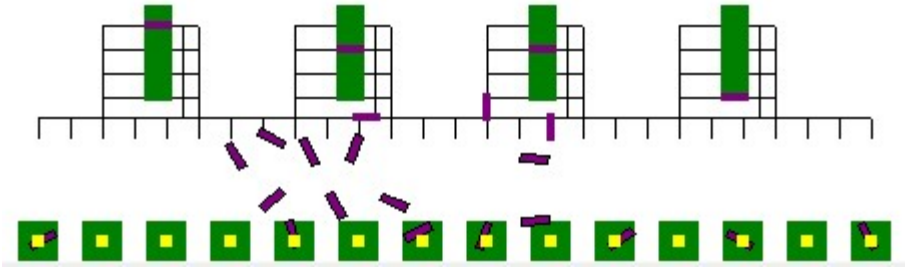


Figure 43 - Snapshot of mixed routing at a QC capacity of 70 moves/hr/QC.

As more AGVs are in operation, delay increases. This makes sense because there are more obstacles to avoid.

Routing conclusions

The results of routing performance show the need for improvement near the QCs when free-ranging is applied and QC capacities are low. Mixed routing has low delay in such situations.
Unfortunately, the topology as implemented is not optimal as delay increases when traffic density also increases. This is not the aim of this routing type and clearly no use is made of both the advantages in traditional and free-range routing. A different topology, with less and/or shorter paths, will improve both routing as terminal performance.

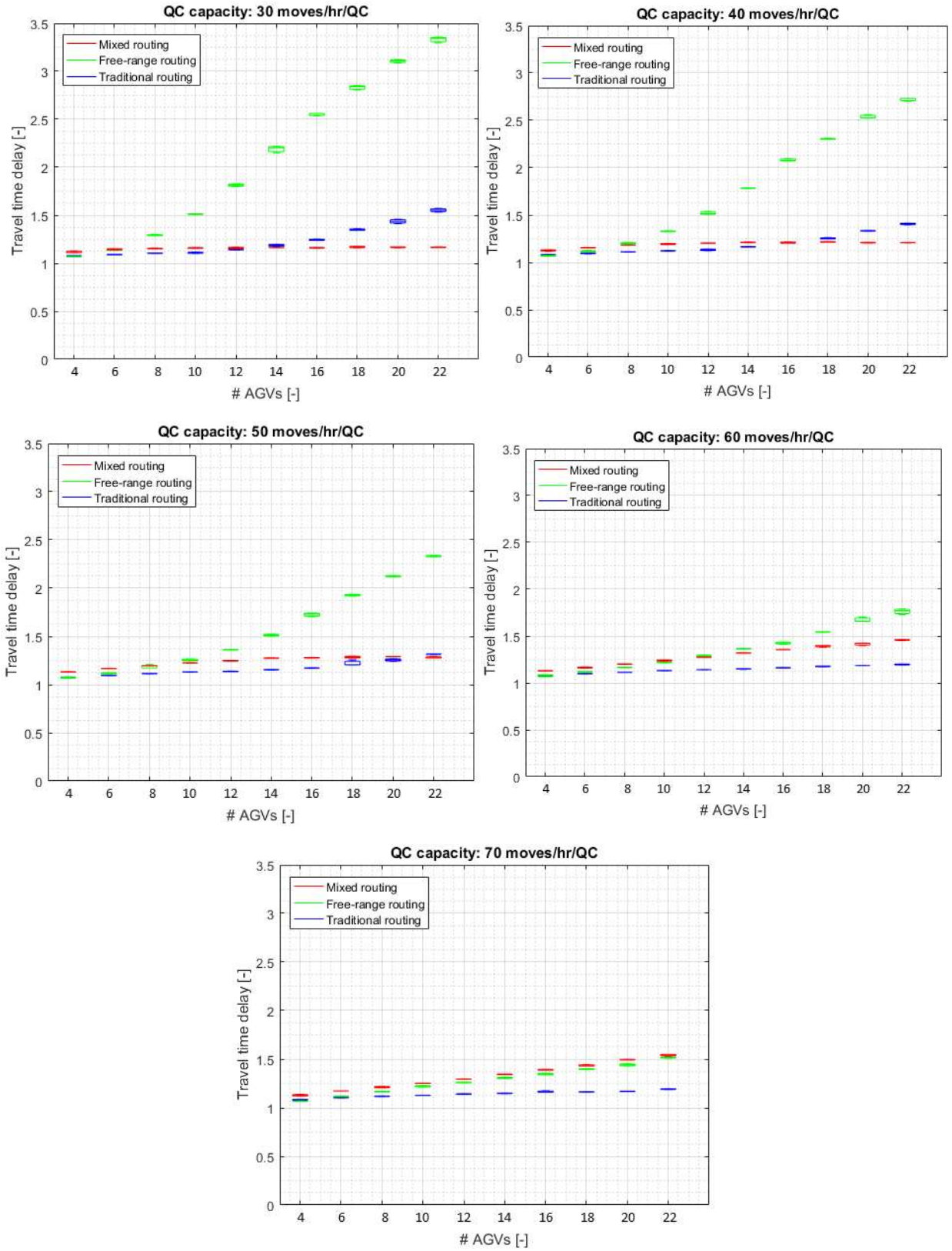


Figure 44 - Routing performance for different QC capacities.

8.4 Computational times

The computational times of the costs experiment in Chapter 8.1 are measured. Experiments are performed on a Windows10 laptop with a 2.0 Ghz Intel Core i7 processor and 12 GB of RAM.

The results are shown in Figure 45, corresponding data is presented in Table 53 (Appendix F).

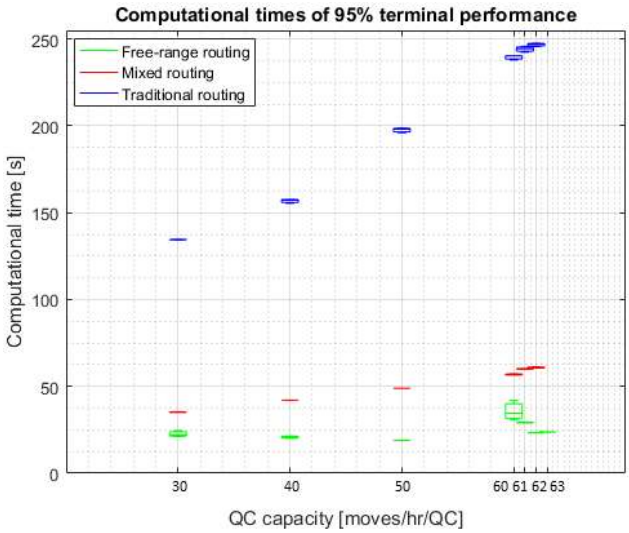


Figure 45 - Computational times of 95% terminal performance.

Traditional routing requires significantly more computational time than mixed or free-range routing. The latter requires least computational power, mixed routing somewhat more. Required AGVs per QC capacity to reach the 95% performance threshold are shown in Table 24.

Traditional routing requires an update of all nodes in each iteration (of each AGV), this is computationally intensive. Mixed routing has less nodes thus intensity is less. Free-range routing calls three functions each iteration, which requires few computational power.

Traditional and mixed routing require similar amounts of AGVs for QC capacities up to and including 50 moves/hr/QC. A clear difference and increase in computational times is seen between both routing types. Computational times of traditional routing increase as more AGVs are in operation (i.e. QC capacity is higher). This increase is smaller for free-range and mixed routing.
Free-range routing shows unexpected results for QC capacity 60 and 61. The cause of this “bump” is unknown, results of computational time as a function of number of AGVs per QC capacity (Figure 57, Appendix F) do not show this effect. Computational time depends on amount of iterations needed, this can differ greatly among similar tests.

An increase in QC capacity and thus traffic has few influence on computational time. It then can also be concluded that traditional routing scales badly: more AGVs increase computational times. The scalability of free-range and mixed routing AGVs is good, no significant increases in computation time are seen (Figure 57, Appendix F).

It must be noted that no efforts are made to make the algorithms computationally efficient. The Dijkstra algorithm used in traditional and mixed routing can be made much more efficient for example. The bad scalability of traditional routing will remain though.

8.5 Conclusions

This chapter presented and discussed the results of the performed experiment. The goal of the experiment is to determine the costs of the AGV system for different QC capacities while a 95% terminal performance must be reached. Both terminal and routing performance were then analyzed as well as the computational times.

It appeared that for capacities up to and including 50 moves/hr/QC the costs increase proportionally with QC capacity for all routing types. Free-ranging is the cheapest option, traditional and mixed routing are more expensive. After this point, more differences between the routing types are revealed. The costs of both mixed routing and free-range routing increase quickly while the costs of traditional routing increase less quick. This implies that free-range routing is sensitive to congestion as more AGVs are in operation.
For all capacities, mixed routing is (among) the most expensive option(s), free-range routing the cheapest option.

Looking into terminal performance it could be concluded that for QC capacities up to and including 50 moves/hr/QC the QC is the bottleneck of the system that prohibits that more AGVs result in higher terminal performance. At higher QC capacities, congestion of the AGVs becomes the bottleneck. The infeed rate of containers at the QCs must go up, therefore more AGVs are needed in the terminal. But this increases the risk of congestion. As a result there is a stochastic infeed of AGVs that affect the terminal performance. The higher the QC capacity, the more influence it has on terminal performance.

Routing performance results showed that free-range routing is indeed sensitive to congestions, especially when QC capacities are low and QC unload spots are hard to reach. At higher QC capacities there is more flexibility in traffic and congestion decreases. Surprisingly, mixed routing has increased delay as QC capacities are higher. The free-range zone appears to be too narrow resulting in too dense traffic. Traditional routing has little delay, the grid offers enough flexibility to reroute.

In terms of computational times, traditional routing requires significantly more computational power. The grid of nodes that must be updated and consulted each iteration is computationally intensive. The potential field method only requires three functions to determine its next action. More AGVs increase the computational times of traditional routing. Free-range and mixed routing remain quite stable. Free-range and mixed routing thus scale well, traditional routing scales bad.

Chapter 9 - Conclusions and recommendations

This chapter presents the conclusions of this research, Chapter 9.1, and the recommendations for further research, Chapter 9.2.

9.1 Conclusions

The research question central to this report was:
What is the effect of combined fixed-path and free-range AGV routing on the routing and terminal performance of an automated container terminal?

The combination of routing types (so-called mixed routing) is suggested to make use of each of the type its benefits and thus to increase performance. Free-range routing has shortest paths but has trouble navigating in crowded environments, there is a high risk of deadlocks. Traditional routing suffers from longer paths but has less issues with high traffic densities. In the newly suggested routing type a path topology is implemented near the QCs, the rest of the terminal is free-ranging.
The implemented topology does not live up to the expectations: at low QC capacities few AGVs are needed and the risk of congestion is low. The travel distances of mixed routing cannot compete with free-range routing. In terms of delay, i.e. routing performance, mixed routing has less delay than free-range routing. At high QC capacities the topology of path in mixed routing appeared not optimal resulting in a free-range zone that is too dense. Congestions affected terminal and routing performance.

The sub questions are now briefly answered.

1. What terminal operations take place in the considered part of an automated terminal and what are the key performance indicators of terminal and routing performance?

AGVs drive on the waterside area of a container terminal and this area is the assumed system in this report. QCs (4x) and ASCs (14x) are on the boundaries of the system and are modeled with stochastic behavior. The loading process of the terminal is assumed: containers are stored in the stack, an ASC picks up a container from the stack and brings it to the transfer point. There, the container is loaded onto an AGV and the AGV drives towards the QC. The QC unloads the AGV and places the container in the vessel. Routing of the AGVs is the studied operation in this report, AGVs must drive in a safe and fast manner towards its destination.

Terminal and routing performance of the system are studied to compare the three routing types.
Terminal performance is quantified as handling efficiency: the handled amount of containers at the QCs divided by the theoretical amount of containers that could be handled.
Routing performance is quantified as a factor of delay: realized travel times are divided by the travel times of the shortest path.
Experiments are performed to determine the lifetime costs of the AGV system of each routing type at different QC capacities while a threshold of 95% terminal performance must be reached. Costs are composed of fixed and variable costs.

2. How is fixed-path routing (“traditional routing”) arranged in container terminals and what is its performance?

The first routing type is fixed-path routing as found on operating container terminals, denoted as traditional routing. A topology of paths is constructed such that traffic control is most reliable. The routes can be static or dynamic. In the former case there is chosen a fixed route from A to B, in the latter case there are multiple options to drive from A to B and the route can be changed on-the-go. Dynamic routing is chosen to be implemented and planning of the route goes incrementally. Both bidirectional (y-direction) and one-directional (x-direction) paths are implemented. The AGVs drive from node by node, each node can be claimed by one AGV simultaneously.

As more AGVs are in operation, terminal performance increases except when QCs become the bottleneck of the system. As QC capacity increases, terminal performance decreases. Routing performance shows that traditional routing has little delay. Delay increases as more AGVs are in operation and decreases as QC capacity is increased. These results are expected.
The costs show a proportional increase for QC capacities up to 50 moves/hr/QC. After this point, more AGVs (and thus costs) are needed to maintain the performance threshold.

3. What are the implementation possibilities for free-range routing into a container terminal and what is its performance?

Free-range methods are roughly divided into classical methods and intelligent methods. Classical methods are cell decomposition, roadmap, potential field and virtual force. Intelligent methods are often inspired by biology and is for example ant colony optimization. MPC is also an emerging technique. Methods can plan entire routes, part of routes or no routes. Methods are rated on completeness, optimality, time and algorithm complexity, flexibility, scalability, safety and robustness. Among a variety of free-range method, the potential field method is chosen. According to literature, the PFM obtains good results while the computational burden remains low. Other methods suffer from a high computational burden. The PFM does not plan a route but drives towards its destination while avoiding obstacles. The risk of deadlocks is the drawback of this method. Each AGV is represented as a circle that may not be entered by another AGV. The repulsive force becomes higher as an obstacle is closer.

In free-range routing the AGVs can drive in a straight line towards their destination, its terminal performance is therefore higher than with traditional routing. As more AGVs are in operation, the risk of congestions increase which affects terminal performance. The increase in terminal performance flattens. Free-ranging outperforms traditional routing for all tested QC capacities and amounts of AGVs. Routing performance shows high delays as more AGVs are in operation, this reduces as QC capacity increases. Similar effect is seen with traditional routing. At the low QC capacities, there is dense traffic near the QCs and deadlocks occur easily. This does however not affect terminal performance as the infeed rate of AGVs at the QCs is still higher than handling rate of the QCs. At high QC capacities, the AGVs are more spaced across the terminal and therefore delay decreases.

In terms of overall AGV system costs, free-range routing is the cheapest type for all QC capacities. A steep increase in costs is however seen at the higher QC capacities.

4. How can traditional and free-range routing be combined ("mixed routing") and what is its performance?

Free-range routing is assumed as basis and a topology of paths (traditional routing) is implemented near the QCs, because crowding is worst in that zone. Paths are one-directional to the right. Each QC has a routing transfer point in which the AGV switches its routing type.

As there is no congestion, mixed routing performs just above traditional routing. The one-directional paths near the QCs add travel distance while the other two routing types could enter from both sides. When amount of AGVs and/or QC capacity is increased, the performance of mixed routing is close to the performance of traditional routing. Traditional routing even outperforms mixed routing slightly as ≥10 AGVs are in operation and QC capacity ≥ 50 moves/hr/QC, this is unexpected. Routing performance shows that delay increases as QC capacity also increases, an opposite effect as seen at traditional and free-range routing. It appears that the free-range zone is too narrow resulting in high traffic densities and thus congestions at the higher QC capacities. The routing transfer point that must be reached when travelling to the QC is hard to reach as departing AGVs from the QC have high repulsive forces. The problem of crowding is shifted from the QCs towards the routing transfer points, this was not the intention of mixed routing. At low QC capacities there is little delay.
The costs of mixed routing are similar to traditional routing at the lower QC capacities, but increase rapidly as QC capacity increases. This is a consequence of the congestion sensitivity at high QC capacities.

Mixed routing, with a topology as implemented in this report, does not make use of the benefits of traditional and free-range routing. Traditional routing is outperformed by free-range routing for all tested QC capacities. The scalability of traditional routing also appeared to be bad, while free-range routing scales well. This proves that free-range routing has high potential to fulfill future terminal needs. However, free-range routing is indeed sensitive to congestions and deadlocks as traffic density increases. The idea of mixed routing is thus still advised, but a proper topology should be developed that increases performance.

9.2 Recommendations

Recommendations can be given for the implemented model of this report, but also more general to the problem statement.

- Mixed routing
  - One could start with the implementation of a procedure in which AGVs that approach the QC get priority over departing AGVs, this will reduce delay and improve terminal performance.
  - It is recommended to let the AGVs enter the QC unload spots from two sides instead of one. Either by making paths bidirectional or reversing the direction of two of the four paths in x-direction.
  - Add more routing transfer points, when the AGV travels to the QC, to reduce traffic density.
  - New topologies must be developed so that the free-range zone is wider and thus traffic density decreases.
- Free-range routing
  - The routing parameters of the PFM should be optimized by analytical models.
  - An overlaying safety system to prevent collisions and deadlocks should be implemented.
  - Represent AGVs as an ellipse instead of circle to improve space utilization.
  - Guide AGVs to cranes so that they park in correct (un)load positions.
  - Improve performance by using checkpoints or heuristics such as particle swarm optimization. This does not require a mixed topology.
- Kinematics of the AGV should be taken into account.
- The terminal model must be extended with the unloading process to simulate more realistic terminal operations.
- The routing types should be implemented in a scaled laboratory to validate the results.
- Strategies to reduce traffic in the terminal, while not affecting performance, should be tested. One simple strategy could be to hold AGVs at the ASCs when all QC unload spots are occupied. Another idea is to use more QCs but with less capacity. Although this report assumes fixed locations of the QCs, in real-life the QCs can be moved to new positions. An interesting research would be to compare the influence on routing and terminal performance when 2 QCs are in operation with a capacity of 30 moves/hr/QC and on the other hand when 4 QCs are in operation with a capacity of 15 moves/hr/QC. This reduces traffic density near the QCs.
- Currently, the routing types do not plan. The influence of a planner should be tested and compared to non-planning results. It is recommended to use time windows for traditional routing and claim coordinates of free-range trajectories. Note that a different free-range method must be chosen as the PFM is a reactive method. Once the routes are planned beforehand, a scheduler can also be implemented to improve performances. This also opens possibilities to optimize entire terminal operations.

References

[1] Port of Rotterdam, “Voortgangs-rapportage 2016 Havenvisie 2030.” Rotterdam, pp. 1–36, 2016.

[2] J. J. M. Evers and S. A. J. Koppers, “Automated Guided Vehicle Traffic Control at a Container Terminal,” *Transp. Res. Part A Policy Pract.* , vol. 30, no. 1, pp. 21–34, 1996.

[3] M. B. Duinkerken, M. Van der Zee, and G. Lodewijks, “Dynamic Free Range Routing for Automated Guided Vehicles,” in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 312–317.

[4] R. Stahlbock and S. Voß, “Vehicle Routing Problems and Container Terminal Operations – An Update of Research,” in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. Golden et al., Ed. Springer Science + Business Media, 2008, pp. 551–589.

[5] F. A. Systems, “AGV KENNIS INSTITUUT.” [Online]. Available: http://www.frog.nl/AGV_Systemen/AGV_Kennis_Instituut. [Accessed: 12-Jan-2017].

[6] A. A. Jose, B. Adarsh S., and C. A. S. Pillai, “A Novel Approach for Scheduling and Routing of the Self Guided Vehicles in Mesh Topology Using Velocity Control and Alternate Path Techniques,” in *Process Automation, Control and Computing (PACC), 2011 International Conference on*, 2011, pp. 1–5.

[7] M. B. Duinkerken, J. A. Ottjes, and G. Lodewijks, “Comparison of routing strategies for AGV systems using simulation,” in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 1523–1530.

[8] H. Young Bae, R. Choe, T. Park, and K. Ryel Ryu, “Travel Time Estimation and Deadlock-free Routing of an AGV System,” *Dyn. Logist.*, pp. 77–84, 2008.

[9] H. Y. Bae, R. Choe, T. Park, and K. R. Ryu, “Comparison of operations of AGVs and ALVs in an automated container terminal,” *J. Intell. Manuf.*, vol. 22, no. 3, pp. 413–426, 2011.

[10] T. Le-Anh and M. B. . De Koster, “A review of design and control of automated guided vehicle systems,” *Eur. J. Oper. Res.*, vol. 171, no. 1, pp. 1–23, 2006.

[11] H. P. M. Veeke, J. A. Ottjes, and G. Lodewijks, *The Delft Systems Approach*. London: Springer, 2008.

[12] L. Qiu, W.-J. Hsu, S.-Y. Huang, and H. Wang, “Scheduling and routing algorithms for AGVs: A survey,” *Int. J. Prod. Res.*, vol. 40, no. 3, pp. 745–760, 2002.

[13] T. J. W. Bentvelsen, “[2015.TEL.7932] Vehicle routing for container transport between quay cranes and stack,” Delft, 2015.

[14] S. H. Choi and W. K. Zhu, “A Bio-Inspired Intelligent Approach to Motion Planning for Mobile Robots,” *Comput. Aided. Des. Appl.*, vol. 8, no. 5, pp. 773–783, 2011.

[15] M. B. Duinkerken, “ME1405: Automation of Transport Systems [Powerpoint slides].” Delft, pp. 1–61.

[16] H. Schim van der Loeff, “Quay Crane productivity at the ECT Delta terminal,” Delft, 2008.

[17] Europe Container Terminals, “ECT Delta Terminal.” [Online]. Available: http://www.ect.nl/nl/content/ect-delta-terminal. [Accessed: 03-Nov-2016].

[18] I. F. A. Vis and I. Harika, “Comparison of vehicle types at an automated container terminal,” *OR Spectr.*, vol. 26, no. 1, pp. 117–143, 2004.

[19] VDL Containersystemen, “Automatic Guided Vehicle,” VDL Containersystemen, 2015.

[20] C. W. Kim, J. M. A. Tanchoco, and P. H. Koo, “Deadlock Prevention in Manufacturing Systems With AGV Systems: Banker’s Algorithm Approach,” *J. Manuf. Sci. Eng. Trans. ASME*, vol. 119, no. 4B, pp. 849–854, 1997.

[21] F. Taghaboni-Dutta and J. M. A. Tanchoco, “Comparison of dynamic routeing techniques for automated guided vehicle system,” *Int. J. Prod. Res.*, vol. 33, no. 10, pp. 2653–2669, 1995.

[22] I. F. A. Vis, “Survey of research in the design and control of automated guided vehicle systems,”

Eur. J. Oper. Res., vol. 170, no. 3, pp. 677–709, 2006.
[23] H. Fazlollahtabar and M. Saidi-Mehrabad, "Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study," *J. Intell. Robot. Syst.*, vol. 77, no. 3–4, pp. 525–545, 2015.
[24] R. Stahlbock and S. Voß, "Operations research at container terminals: A literature update," *OR Spectr.*, vol. 30, no. 1, pp. 1–52, 2008.
[25] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, "Transport operations in container terminals: Literature overview, trends, research directions and classification scheme," *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 1–13, 2014.
[26] Q. Li, A. C. Adriaansen, J. T. Udding, and A. Y. Pogromsky, "Design and Control of Automated Guided Vehicle Systems: A Case Study," in *IFAC Proceedings Volumes*, 2011, vol. 44, no. 1, pp. 13852–13857.
[27] M. Tlig, O. Buffet, and O. Simonin, "Cooperative behaviors for the self-regulation of autonomous vehicles in space sharing conflicts," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2012, vol. 1, pp. 1126–1132.
[28] G. E. Jan, K. Y. Chang, and I. Parberry, "Optimal Path Planning for Mobile Robot Navigation," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 4, pp. 451–460, 2008.
[29] Q. Li, J. T. Udding, and A. Y. Pogromsky, "Modeling and control of the AGV system in an automated container terminal," in *Proceedings of the IASTED International Conference on Modelling, Identification and Control*, 2010.
[30] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic, "Time Windows Based Dynamic Routing in Multi-AGV Systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 151–155, 2010.
[31] S. M. Jeon, K. H. Kim, and H. Kopfer, "Routing automated guided vehicles in container terminals through the Q-learning technique," *Logist. Res.*, vol. 3, no. 1, pp. 19–27, 2010.
[32] S. Shao, Z. Xia, G. Chen, J. Zhang, Y. Hu, and J. Zhang, "A new scheme of multiple automated guided vehicle system for collision and deadlock free," in *ICIST 2014 - Proceedings of 2014 4th IEEE International Conference on Information Science and Technology*, 2014, pp. 606–610.
[33] ECT, "ECT Delta Terminal." [Online]. Available: http://myservices.ect.nl/Terminals/rotterdamterminals/deltaterminal/Pages/default.aspx. [Accessed: 28-Nov-2016].
[34] S. G. Anavatti, S. L. X. Francis, and M. Garratt, "Path-Planning Modules for Autonomous Vehicles: Current Status and Challenges," in *ICAMIMIA 2015 - International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation, Proceeding - In conjunction with Industrial Mechatronics and Automation Exhibition, IMAE*, 2015, pp. 205–214.
[35] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: A survey," *Rob. Auton. Syst.*, vol. 86, pp. 13–28, 2016.
[36] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.
[37] F. Kamil, S. H. Tang, W. Khaksar, N. Zulkifli, and S. A. Ahmad, "A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments," *Adv. Robot. Autom.*, vol. 4, no. 2, pp. 1–8, 2015.
[38] S. Ware and R. Su, "Incremental Scheduling of Discrete Event Systems," in *Proceedings of the 13th International Workshop on Discrete Event Systems*, 2016, pp. 147–152.
[39] T. Weerakoon, K. Ishii, and A. A. F. Nassiraei, "AN ARTIFICIAL POTENTIAL FIELD BASED MOBILE ROBOT NAVIGATION METHOD TO PREVENT FROM DEADLOCK," *J. Artif. Intell. Soft Comput. Res.*, vol. 5, no. 3, pp. 189–203, 2015.
[40] J. K. Goyal and K. S. Nagla, "A new approach of path planning for mobile robots," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*,

2014, pp. 863–867.
[41] S. A. M. Coenen, “Motion Planning for Mobile Robots - A Guide,” Eindhoven University of Technology, 2012.
[42] F. Fahimi, *Autonomous Robots*. New York, NY: Springer, 2009.
[43] S. H. Choi and W. K. Zhu, “Performance Optimisation of Mobile Robots in Dynamic Environments,” in *Proceedings of IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems,VECIMS*, 2012, pp. 54–59.
[44] J. Borenstein and Y. Koren, “Real-Time Obstacle Avoidance for Fast Mobile Robots,” *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1989.
[45] J. Borenstein and Y. Koren, “The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots,” *IEEE J. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, 1991.
[46] M. A. Montes de Oca, “Behavior of real ants,” *ACO Methaheuristics*, 2009. [Online]. Available: http://www.aco-metaheuristic.org/RealAnts.html. [Accessed: 29-Nov-2016].
[47] M. Hamzheei, R. Z. Farahani, and H. Rashidi-Bajgan, “An ant colony-based algorithm for finding the shortest bidirectional path for automated guided vehicles in a block layout,” *Int. J. Adv. Manuf. Technol.*, vol. 64, no. 1–4, pp. 399–409, 2013.
[48] R. R. Negenborn, “ME1406 - Control for Intelligent Transport Infrastructures [PowerPoint presentation],” 2015, pp. 1–61.
[49] J. Xin, R. R. Negenborn, and G. Lodewijks, “Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control,” *Transp. Res. Part C Emerg. Technol.*, vol. 44, pp. 214–230, 2014.
[50] J. Xin, R. R. Negenborn, and G. Lodewijks, “Trajectory planning for AGVs in automated container terminals using avoidance constraints: a case study,” in *IFAC Proceedings Volumes*, 2014, pp. 9828–9833.
[51] F. Corman *et al.*, “Optimal Scheduling and Routing of Free-range AGVs at Large Scale Automated Container Terminals,” *Period. Polytech. Transp. Eng.*, vol. 44, no. 3, pp. 145–154, 2016.
[52] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks, “Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance,” *Transp. Res. Part C Emerg. Technol.*, vol. 60, pp. 377–396, 2015.
[53] H. Ismkhan, “Effective heuristics for ant colony optimization to handle large-scale problems,” *Swarm Evol. Comput.*, vol. 32, pp. 140–149, 2017.
[54] E. Ferrera, J. Capitán, A. R. Castaño, and P. J. Marrón, “Decentralized safe conflict resolution for multiple robots in dense scenarios,” *Rob. Auton. Syst.*, vol. 91, pp. 179–193, 2017.
[55] L. Pallottino, V. G. Scordio, A. Bicchi, and E. Frazzoli, “Decentralized Cooperative Policy for Conflict Resolution in Multivehicle Systems,” *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1170–1183, 2007.
[56] M. Bähr, R. V. Donner, and T. Seidel, “A Pedestrian Dynamics Based Approach to Autonomous Movement Control of Automatic Guided Vehicles,” in *Robust Manufacturing Control*, 2012, pp. 175–187.
[57] M. Zhou, H. Dong, F.-Y. Wang, Q. Wang, and X. Yang, “Modeling and simulation of pedestrian dynamical behavior based on a fuzzy logic approach,” *Inf. Sci. (Ny).*, vol. 360, pp. 112–130, 2016.
[58] D. Helbing, L. Buzna, A. Johansson, and T. Werner, “Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions,” *Transp. Sci.*, vol. 39, no. 1, pp. 1–24, 2005.
[59] J. A. Ottjes and H. P. M. Veeke, “WB3417 - Lecture September 21, 2015,” 2015, pp. 1–23.
[60] C. I. Liu, H. Jula, K. Vukadinovic, and P. Ioannou, “Automated guided vehicle system for two container yard layouts,” *Transp. Res. Part C Emerg. Technol.*, vol. 12, no. 5, pp. 349–368, 2004.
[61] M. B. Duinkerken and G. Lodewijks, “Routing of AGVs on automated container terminals,” in *Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative*

Work in Design, CSCWD 2015, 2015, pp. 401–406.
[62] K. Hoad, S. Robinson, and R. Davies, "Automating DES Output Analysis: How Many Replications to Run," in *Proceedings of the 2007 Winter Simulation Conference*, 2007, pp. 505–512.
[63] L. Henesey, P. Davidsson, and J. A. Persson, "Comparison and Evaluation of Two Automated Guided Vehicle Systems in the Transshipment of Containers at a Container Terminal," in *MODSIM World Conference 2007*, 2007.

Appendix A - Scientific research paper

Combining fixed-path and free-range AGV routing on a container terminal

D.L. Overbeek, M.B. Duinkerken, R.R. Negenborn

***Abstract*— Terminals are encouraged to redesign their operations to facilitate future growth of container freight. Routing of Automated Guided Vehicles (AGVs) can contribute significantly to this trend. Traditional routing, as implemented in real-life terminals, uses a topology of paths. To improve flexibility and scalability of the AGV system, a promising control method is already proposed in literature that does not require a topology of paths: free-ranging. For local control methods, problems are expected to rise near the cranes where the risk of deadlocks is high due to crowding. A new routing type is proposed in this paper that is aims to combine the benefits of free-range and traditional routing: mixed routing. A topology of paths is implemented near the quay cranes (QCs) to guide the AGV flow, the rest of the terminal is a free-range zone. This study compares traditional, free-range and mixed routing on both routing- and terminal performance for several QC capacities. Results show that the implemented topology does not live up to the expectations. At high QC capacities, delay of mixed routing increases which affects terminal performance. The topology appears not suited for high traffic densities. Free-range and traditional routing have less delay and outperform mixed routing. At low QC capacities, few AGVs are needed and thus travel distances are leading for the results. Free-range routing performs best, followed by mixed and traditional routing.**

I. Introduction

Container freight grew since the introduction of the container and this trend is expected to continue in the future [1]. Container terminals are encouraged to increase their terminal performance to facilitate future growth and remain competitive. To contribute to terminal performance, automated equipment such as Quay Cranes (QCs), Automated Stacking Cranes (ASCs) and Automated Guided Vehicles (AGVs) are introduced. AGVs are in operation since 1993 and have been studied much since then, especially with focus on their routing. Stahlbock & Voβ provide a thorough review [2]. AGVs operating in container terminals drive on pre-determined paths, this topology is designed such that traffic control is most reliable. Traffic control ensures that the AGV reaches its destination in a collision-, conflict- and deadlock-free manner. The first topology consisted of a loop along the stacks and quay, but nowadays more crossings are added to decrease travel distances. Traditional routing is arranged with a centralized controller. Drawbacks are the lack of flexibility and scalability.

A new type of routing is introduced in literature: free-range routing. This is similar to a mobile robot in the broader studied field of robotics. Free-range routing does not require a path topology and thus the amount of route options is infinite. This type is not implemented in real-life terminals. Free-range routing can be controlled via central or decentral control. Central control methods are for example cell decomposition, roadmap and Model Predictive Control. A well-known decentral control method is the Potential Field Method (PFM). The DEFT controller [3], MILP controller [4] and potential field method (PFM) [5] are studied for an AGV case at a container terminal. Only PFM is a decentral control method. These studies merely focused on routing performance, not on terminal performance.

The performance of free-ranging AGVs in crowded, chaotic zones is questionable, crowds shall occur near the cranes. The risk of deadlocks is high, which affects terminal performance. This paper introduces a new routing type that is expected to make use of the benefits of both free-range and traditional routing: mixed routing. Mixed routing has a topology of paths near the cranes to guide the AGV flow. Free-range routing is applied in the remaining part of the terminal.

This paper studies the effect of mixed routing on terminal and routing performance for different QC capacities. The results are compared to traditional routing and free-range routing. Mixed routing is expected to outperform free-range and traditional routing when QC capacity increases. Especially the performance of free-range routing is expected to decrease as more AGVs are in operation due to the resulting deadlocks. Traditional routing is expected to have the lowest terminal and routing performance.

This paper is organized as follows: Section II explains the terminal model and the three routing types. Section III describes the experiments and results. Conclusions and recommendations are presented in Section IV.

II. Terminal model and routing types

The assumed system is the waterside area of a container terminal where AGVs transport containers. QCs and ASCs are on the boundaries of this system. This research only assumes the loading process of containers, thus transport from stack (ASC) to quay (QC). There are four QCs and 14 ASCs present in the terminal. The terminal topology is shown in Figure 1. AGVs are represented in orange, the lines in y-direction represent buffer places. A simulation model of the terminal is made in Delphi Borland and TOMAS to cope with the stochastic complexity of the system.

Each container follows the handling sequence of ASC, AGV and QC.

ASC

An ASC cycles to the container location in stack, picks up the container and cycles to the transfer point. There, the container is loaded onto an AGV. Two parameters represent the ASC behavior: container retrieval time and load time. The exact container location in stack is not assumed; the container retrieval time is stochastic. There is one loading spot for AGVs at each ASC.

AGV

An AGV transports the container from stack to quay. An idle or empty-riding AGV is assigned to a container, the AGV drives towards the correct ASC and waits until the container is loaded onto the AGV. Then it drives to the QC and waits at an unloading spot until the container is unloaded. When unloading is completed, the AGV drives towards the buffer, but in the meanwhile it can be assigned to a new container (i.e. empty-riding AGV).

QC

A QC unloads the container from the AGV that waits below the QC, the container is placed in the vessel and the QC cycles back to the transfer point. Two parameters represent QC behavior: unload time and put away time. Exact container locations in the vessel are not assumed; put away time of the QC is stochastic. There are four unloading spots for AGVs below each QC.

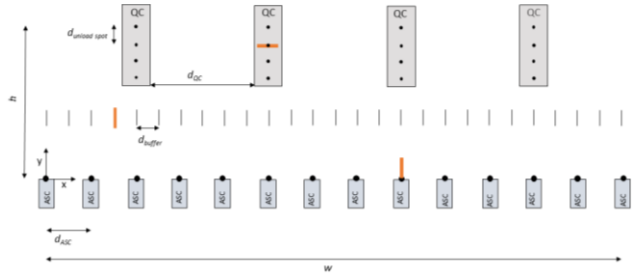


Figure 1 - Topology of terminal model.

The ASCs and QCs are an open system in which containers enter and leave the system, respectively. The AGV system is a closed system. Input parameters and fixed values of the model are shown in Table 1.

KPIs

Results of the terminal model will be compared based on the costs of the AGV system, the terminal performance and routing performance. Costs of the AGV system are calculated by:

$$AGV\ costs\ [€] = \#_{AGVs} \cdot c_{fixed} + \#_{AGVs} \cdot c_{variable} * t_{operating\ hours} \quad (1)$$

The fixed costs are the purchase costs of an AGV. The variable costs are the operational costs of an AGV per operational hour, this includes maintenance, depreciation and fuel costs [6].

The KPI to represent terminal performance is quantified as handling efficiency and is calculated as:

$$Handling\ efficiency\ [-] = \frac{Handled\ containers\ in\ simtime}{\frac{Simtime}{Cycle\ time\ of\ QC} \cdot 4\ QCs} \quad (2)$$

Routing performance is represented with a factor of delay:

$$Travel\ time\ delay\ [-] = \frac{Realized\ travel\ time\ to\ ASC + realized\ travel\ time\ to\ QC}{Minimal\ travel\ time\ to\ ASC + minimal\ travel\ time\ to\ QC} \quad (3)$$

The minimal travel time is the shortest path from current location to destination, thereby assuming no traffic and thus no congestion or detours.

Table 1 - Input parameters of model.

	Parameter	Value	Unit	Type of value
Topology	Terminal size (wxh)	416 x 108	m	Constant
	d_{ASC}	32	m	Constant
	d_{QC}	96	m	Constant
	d_{buffer}*	Depends on routing	m	Constant
	$d_{unload\ spot}$	12	m	Constant
Operational	Number of AGVs	Varies	-	-
	Max. speed of AGV	4	m/s	Constant
	Size of AGV(lxw)	14.7 x 3.7	m	Constant
	Acceleration of AGV*	1	m/s^2	Constant
	Deceleration of AGV**	1	m/s^2	Constant
	Unload time of AGV at QC	Varies	s	Constant
	Put away time of QC	N(μ,σ)	s	Normal distribution
	Load time of AGV at ASC	10	s	Constant
	Container retrieval time of ASC	Empirical distribution	s	Empirical distribution[1]
	Simulation time	50000	s	Constant
Costs related	Purchase of 1 AGV	282000	€	Constant
	Operational costs of 1 AGV per hr	4.50	€	Constant

* Traditional routing: d_{bufer} = 8 m. Free-range and mixed routing: : d_{bufer} = 16 m.
**When traditional routing is applied. Free-range routing has no fixed acceleration and deceleration.

In general, routing of the AGV goes as follows:

```
Receive destination
While current AGV location is not destination
    Determine next move
    Hold(drive time)
    Update location
    If near QC and all unload spots occupied
        Standby
```

Figure 2 - Routing pseudo algorithm.

The routing part of the terminal model can be of the type traditional, free-range or mixed routing. The three routing types will be explained into more detail.

A. Traditional routing

The first routing type is the type that is currently used and is found on terminals in which AGVs are in operation. A topology of paths is constructed on which the AGVs drive, see Figure 3. The topology looks like a grid to improve route flexibility. The lanes in x-direction are one directional: the direction on the upper two lanes of each set is restricted to the right, the direction of the bottom two lanes of each set is restricted to the left. The lanes in y-direction are bidirectional, except below the QCs (no paths) and the red arrows in Figure

[1] Distribution as reported by Vis and Harika [8].

3. Each intersection is a node, as are the ASC loading spots. AGVs travel from node to node. Traffic control is arranged with semaphores, each node has a capacity of one.

Route planning goes in an incremental manner. A Dijkstra algorithm calculates the shortest path from current location to destination, the first three nodes on the route must be free. When a route is found, the AGV drives to the first node on this route. Once arrived at the node, the algorithm runs again to determine the next node to travel to. When a node is occupied, the AGV holds at its current location.
Because the AGV its length exceeds the distance between two nodes, two nodes are claimed simultaneously by the AGV. Both claims are updated each iteration.
In the case that an AGV drives towards a QC but all its unload spots are occupied, the AGV is held at a buffer place until a spot becomes free.
AGV speed is assumed 4 m/s, acceleration and deceleration both 1 m/s^2.

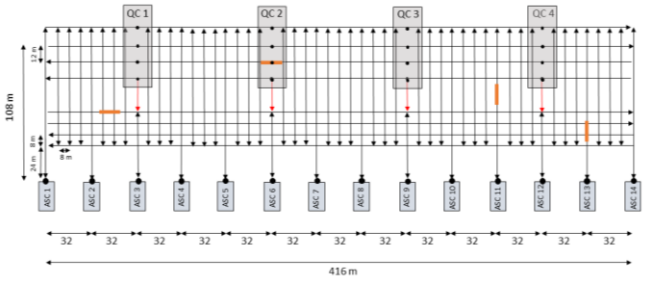


Figure 3 - Topology of traditional routing.

B. *Free-range routing*

The second routing type is free-range routing. No topology of paths is needed thus the terminal is similar to the terminal model (Figure 1). There are numerous methods to arrange free-ranging [7], but this research chose to implement the potential field method (PFM). PFM is a decentral control method and has the advantage of low computational times. It is a reactive method that does not plan.

Each iteration, the potential field is constructed for the AGV. The destination thereby has an attractive force, while all obstacles have a repulsive force. All repulsive forces are summed into one resultant repulsive force. The resultant of the attractive force and the resultant repulsive force determines the direction and speed of the AGV. The algorithm constructs the potential field each second.
Obstacles in a specified environment of the AGV are assumed. The closer an obstacle is to the AGV, the more repulsive force it exceeds. Attractive force acts opposite: the closer the AGV is to the goal, the lower the attractive force is. Each goal and obstacle have a radius and spread. The sum of radius and spread equals the environment. The radius of the obstacle determines the minimum distance to another obstacle. When an AGV is within this distance, an escape force is activated. The radius of a driving AGV is determined by the situation that two AGVs are heading towards each other. The breaking distance from full speed to stop equals the radius:

$$r_{AGV} = \frac{{v_{max}}^2}{a_{max}} = \frac{4^2}{1} = 16\ m \qquad (4)$$

If the obstacle does not move, the radius can be reduced. The PFM parameters are implemented as shown in Table 2. Near the QCs, the obstacle radius and spread are reduced to make it possible to arrive at an unloading spot.
Speed of the AGV is limited to 4 m/s. Acceleration and deceleration are determined by the PFM and differ per iteration.

Table 2 - PFM implemented input parameters.

Parameter	Value	Unit	Type of value
radiusGoal	1.00	m	Constant
radiusObstacle	16.00	m	Constant
spreadGoal	9.00	m	Constant
spreadObstacle	20.00	m	Constant
reduced radiusObstacle	10.00	m	Constant
reduced spreadObstacle	10.00	m	Constant
Alpha (scaling factor for attractive force)	0.44	-	Constant
Beta (scaling factor for repulsive force)	0.20	-	Constant

C. *Mixed routing*

A drawback of the PFM is its sensitivity to deadlocks. Traditional routing is less sensitive to this. Applying a path topology near the QCs, where crowds and thus deadlocks will occur, is expected to use the benefits of both traditional and free-range routing.
This topology is connected to the buffer places and shown in Figure 4. Free-range routing is not allowed above the buffer places, a repulsive force downwards is implemented. Path routing is controlled by semaphores, as explained in section *A*. Each QC has one routing transfer point where the AGV switches from free-range routing to traditional routing. This point is located to the left of each QC, the red dots in Figure 4. Routing transfer points where the AGV switches from traditional routing to free-range routing are located at the remaining buffer places.

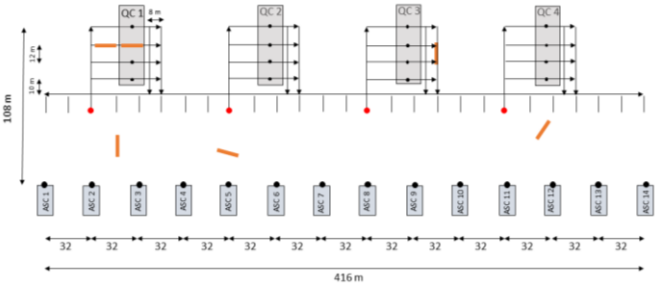


Figure 4 - Topology of mixed routing.

Routing of the AGV, considered in buffer, goes as follows. It is assigned a container and it free-ranges to the ASC. After the container is loaded onto the AGV, the AGV free-ranges towards the correct routing transfer point (at the left of the QC). When it is arrived at this point, the AGV is routed via the paths towards an unloading spot. After unloading, the AGV is routed via the paths to a buffer place. If no new container is assigned to the AGV, the AGV goes idle when arrived at the buffer. Else, the buffer place contributing to the shortest route to the container (that is still in stack) is chosen as the routing transfer point from traditional routing to free-range routing.
If all unload spots of a QC are occupied, AGVs wait on the path section to the left of an unload spot.

The terminal model and three routings types are verified and validated.

III. Experimental plan

Experiments are performed to obtain the AGV costs per routing type for different QC capacities. For each routing type and QC capacity, the minimum amount of AGVs needed to reach a 95% terminal performance (2) threshold is determined. Amounts are converted to costs with (1), costs are calculated for a lifetime of the AGV system and are a summation of fixed and variable costs. A lifetime of the AGV is assumed as 72000 operating hours, ±10 years. Results are explained with terminal and routing performance.

Number of AGVs varies between 4 and 23 with steps of 1. QC capacity ranges between 30 and 70 moves/hr/QC with intermediate steps of 10. Terminal model inputs are as shown in Table 1.

Each simulation ran for 50000 seconds of which the first 4000 seconds are the warmup period. The results of the warmup period are included in the results, but are of negligible influence. Each simulation is replicated three times, based on the confidence interval method.

IV. Results

The lifetime costs of the AGV system as a result of the QC capacity are determined (Figure 5). Corresponding number of AGVs are presented in Table 3.Table 3 – AGVs required to reach 95% terminal performance.

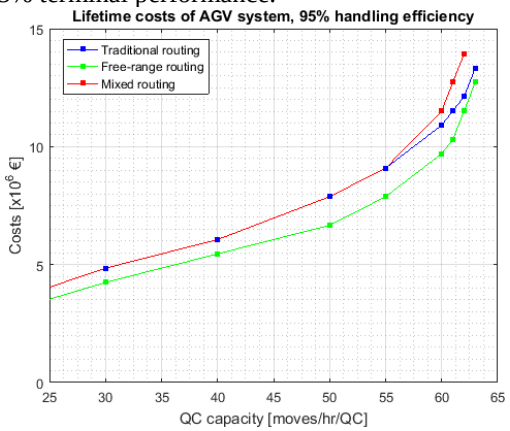


Figure 5 - Lifetime costs of AGV system, assuming 95% terminal performance.

For QC capacities up to and including 50 moves/hr/QC, costs increase proportionally with QC capacity. Mixed and traditional routing have similar costs, free-range routing has fewest costs.

At higher QC capacities, the routing types show non-proportional costs. The costs of mixed and free-range routing increase rapidly. The costs of traditional routing has a flatter slope, resulting in an approach towards free-range routing costs. Traditional routing appears to be flexible enough to keep congestions and detours low. As a result, its longer paths lengths become competitive with the shorter, but congested, paths of free-range routing.

Note that the results of QC capacity 70 moves/hr/QC are not shown, because neither routing type could reach the 95% threshold. Starting from 60 moves/hr/QC, QC capacity is increased in steps of 1 until the threshold could not be reached.

Table 3 – AGVs required to reach 95% terminal performance.

Capacity [moves/hr/QC]	Traditional routing	Free-range routing	Mixed routing
30	8	7	8
40	10	9	10
50	13	11	13
60	18	16	19
61	19	17	21
62	20	19	23
63	22	21	-

A. Terminal performance

The terminal performances are shown in Figure 6, Figure 7 and Figure 8. Results of QC capacities 40 and 60 are not shown.

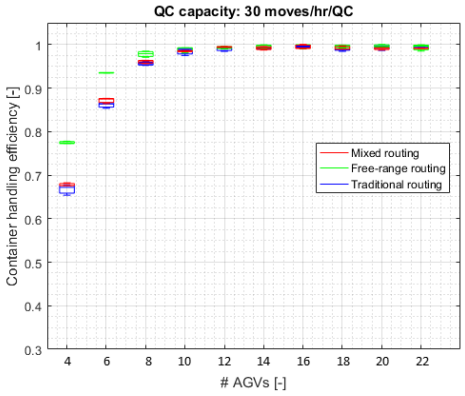


Figure 6 – Terminal performance, QC capacity 30 moves/hr/QC.

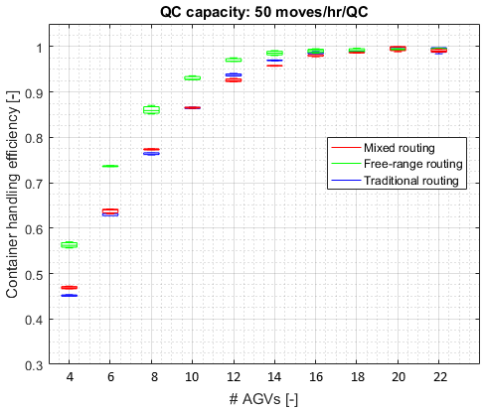


Figure 7 – Terminal performance, QC capacity 50 moves/hr/QC.

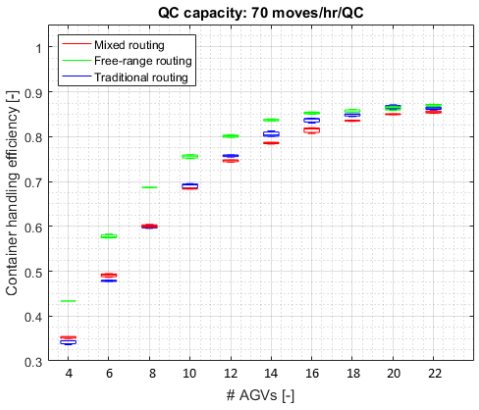


Figure 8 – Terminal performance, QC capacity 70 moves/hr/QC.

A low QC capacity requires few AGVs to obtain high performance. The QCs cannot cope with a faster infeed rate and thus a performance of nearly 1 is quickly reached for all routing types (Figure 6). The shortest path distances are leading for the results, because little to no congestion will occur. Free-ranging reaches significantly more performance than mixed or traditional routing.

As QC capacity increases, more AGVs are needed to reach high performances. Free-range routing remains the routing type that reaches highest performances. Traditional and mixed routing perform similar. Furthermore, traditional routing slightly outperforms mixed routing as QC capacity is 50 or 70 moves/hr/QC and number of AGVs is ≥ 12 or 10, respectively. This is unexpected and further analyzed with routing performance.
As QC capacity is 70 moves/hr/QC, a significant performance reduction is seen for all routing types. The routing types reach similar performance as 22 AGVs are in operation. Neither routing type can cope with the high QC capacity.
The higher QC capacities show the sensitivity of free-range routing to congestion. Free-range routing is the first type of which its trend start to flatten, followed by mixed routing and last traditional routing. The free-range component of mixed routing is the cause of being outperformed by traditional routing.

B. Routing performance

Routing performances corresponding to terminal performances are shown in Figure 11, Figure 12 and Figure 13.
For QC capacities 30 and 50 moves/hr/QC, free-range routing has a vast increase in delay as more AGVs are in operation. Often deadlocks occur between three AGVs and are resolved when a fourth AGV arrives (Figure 9). Traditional routing also experiences delay but in lesser amounts. Delay in traditional routing occurs when all QC unload spots are occupied and the AGVs swarm around the QC. Mixed routing has stable delay even when more AGVs are in operation. The path topology proves its use.

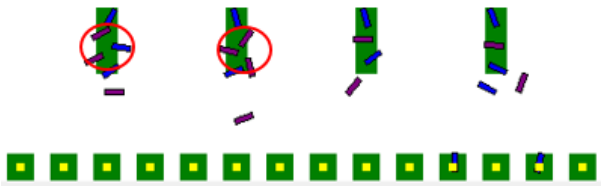


Figure 9 - Free-range routing, deadlocks at QCs.

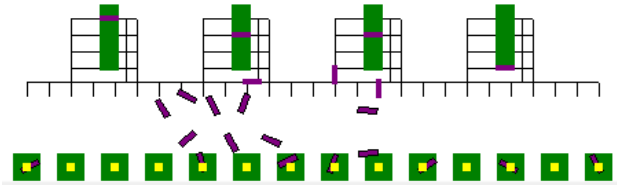


Figure 10 - Mixed routing, QC capacity 70 moves/hr/QC.

As QC capacity is 70 moves/hr/QC, delay is decreased for free-range and traditional routing. The QCs handle containers fast and thus traffic is more spaced across the terminal. This increases the route flexibility in both routing types.
Mixed routing has increased delay at this QC capacity. The free-range zone appears to be too narrow resulting in more congestion. AGVs struggle to reach the routing transfer point when on its way to the QC (Figure 10).

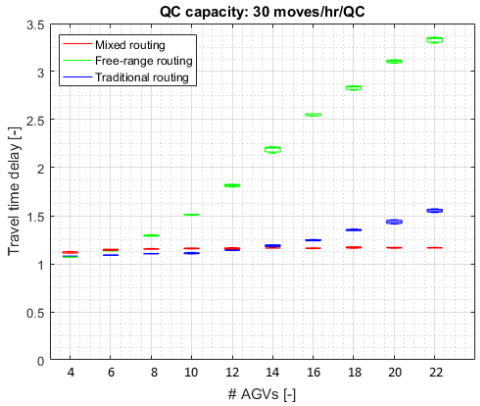


Figure 11 - Routing performance, QC capacity 30 moves/hr/QC.

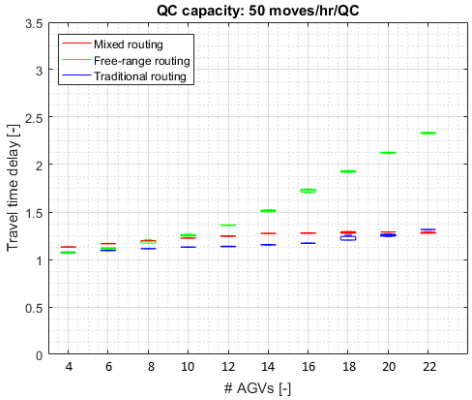


Figure 12 - Routing performance, QC capacity 50 moves/hr/QC.

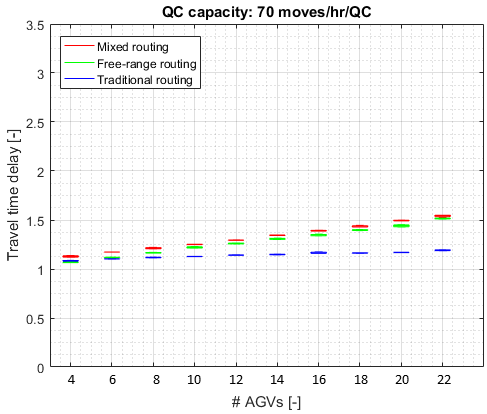


Figure 13 - Routing performance, QC capacity 70 moves/hr/QC.

Despite the low delay of mixed routing at low QC capacity, mixed routing never stands out as favorable routing type.
At low QC capacities, QCs are the bottleneck of the terminal. There are four unload spots and AGVs have plenty of time to reach the QC. The delay of free-range routing does not affect terminal performance.

Routing becomes more important as QC capacities are high and congestion immediately affects terminal performance. This is seen at a QC capacity of 70 moves/hr/QC. At these higher QC capacities, mixed routing had the unfortunate effect of increased delay.

V. Conclusion and recommendations

This paper proposed mixed routing: a combination of fixed-path and free-range routing. Mixed routing guides AGVs in dense traffic areas via paths, situations in which free-range routing is sensitive to deadlocks. It is expected that terminal performance increases because the benefits of both fixed-path and free-range routing are combined.
Experiments in a generic terminal model are performed with three routing types: traditional (fixed-paths), free-range and mixed routing. The results of mixed routing did not meet its expectations, because of the following two explanations:

- At low QC capacities few AGVs are needed and the risk of congestion is low. The travel distances of mixed routing cannot compete with those of free-range routing. In terms of delay, i.e. routing performance, mixed routing has less delay than free-range and traditional routing.
- At high QC capacities, the path topology in mixed routing appeared not optimal resulting in a free-range zone that is too dense. The problem of dense, chaotic areas is shifted towards the routing transfer point. Congestions affected terminal and routing performance.

Lifetime costs of the AGV system, both fixed and variable, revealed that for lower QC capacities, costs scale proportional with QC capacity. Free-range routing is cheapest. Traditional and mixed routing have similar costs and are more expensive. At capacity > 50 moves/hr/QC, costs of mixed and free-range routing increase quickly while traditional routing has a less steep costs increase. Traditional routing slowly approaches free-range routing in terms of costs.

Mixed routing, as implemented in this study, did not live up to its expectations. Looking at the potential of free-range routing and its serious drawbacks, mixed routing is still advised as a routing solution. It is recommended to change the topology into one that does make use of the benefits of both traditional and free-range routing. For example, use bidirectional paths and implement more routing transfer points when the AGV travels towards the QC. This will reduce the traffic density and provide more flexibility. Another idea is to give arriving AGVs priority over departing AGVs near the routing transfer points when AGVs are headed towards a QC.
Traffic reduction strategies are also recommended. The effect on performance should be studied when AGVs are held at the ASCs while the QC unload spots are occupied. Another suggestion is to spread total QC capacity over multiple QCs. The terminal model should be extended with the unloading process to represent all terminal processes. Once routing is optimized, a new challenge lies in the optimization of equipment interaction to facilitate future terminal throughputs.

References

[1] Port of Rotterdam, “Voortgangs-rapportage 2016 Havenvisie 2030.” Rotterdam, pp. 1–36, 2016.

[2] R. Stahlbock and S. Voß, “Vehicle Routing Problems and Container Terminal Operations – An Update of Research,” in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. Golden et al., Ed. Springer Science + Business Media, 2008, pp. 551–589.

[3] M. B. Duinkerken, M. Van der Zee, and G. Lodewijks, “Dynamic Free Range Routing for Automated Guided Vehicles,” in *Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 312–317.

[4] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks, “Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance,” *Transp. Res. Part C Emerg. Technol.*, vol. 60, pp. 377–396, 2015.

[5] S. H. Choi and W. K. Zhu, “A Bio-Inspired Intelligent Approach to Motion Planning for Mobile Robots,” *Comput. Aided. Des. Appl.*, vol. 8, no. 5, pp. 773–783, 2011.

[6] L. Henesey, P. Davidsson, and J. A. Persson, “Comparison and Evaluation of Two Automated Guided Vehicle Systems in the Transshipment of Containers at a Container Terminal,” in *MODSIM World Conference 2007*, 2007.

[7] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Rob. Auton. Syst.*, vol. 86, pp. 13–28, 2016.

[8] I. F. A. Vis and I. Harika, “Comparison of vehicle types at an automated container terminal,” *OR Spectr.*, vol. 26, no. 1, pp. 117–143, 2004.

Appendix B - Related literature (Chapter 3)

This appendix briefly reviews the free-range methods and the organization methods on their criteria points, see Chapter 3.2 and Chapter 3.3 respectively. An overview of studied literature is presented at the end of this Appendix.

Free-range methods

Cell decomposition

Out of the three decomposition methods, approximate cell decomposition is most suited for AGVs at a container terminal. Exact decomposition is based on static objects, these are present but do not generate computationally efficient cells. Adaptive decomposition would require an update of the grid each time step, which it computationally very inefficient. Approximate cell decomposition is thus most suited. Due to the rectangular size of an AGV, it is recommended to use cell sizes smaller than an AGV to improve the space utilization.

Routes in approximate cell decomposition can be planned entirely or incremental. Cell decomposition with complete route planning is a complete method provided that the cell resolution is small enough, else the case as in the middle figure of Figure 13 occurs. It is an optimal method.
Cell decomposition with incremental planning is incomplete, but it is likely that a route to the destination is eventually found. Incremental planning is not optimal. A proper time or distance horizon should be determined such that it the correct balance between computational burden and optimality (as far as possible) is found.
A smaller cell size increases the computational burden, thus a feasible balance must be found when using this method. The algorithm itself is not that complex. Both methods score also well on the other requirements; it is flexible because many route can be (re)planned and very safe because a cell can only be claimed by one AGV. This method is robust when an AGV breaks down, but if the central controller fails the entire system fails. It is possible to scale this system up with this method, but the risk of deadlocks increases.

Roadmap

Between a visibility graph and a Voronoi diagram, the latter would be more suited for a container terminal as the flexibility is better with this method. A Voronoi diagram in the case of an automated container terminal would look like a grid such as with approximate cell decomposition because the field is based on regularly spaced nodes instead of the few available static obstacles. Therefore, this evaluation is the same as already discussed at Approximate cell decomposition. This method can use incremental or complete route planning.

Potential field

Potential field is an incomplete method, because the AGV can be trapped in a local minima. There are tricks to avoid the occurrence of local minima, such as escape force [41]. This method is not optimal, as there is no planner that takes the objective (shortest path) into account. This method has a local approach, thus computational times are low. The amount of environment to include in this method should be based on breaking distance of the AGVs. The algorithm is easy to implement. It is a flexible method, because each time step every thinkable direction can be chosen. Scalability is very good, because it is a local method it does not necessarily increase computational times. Safety is not guaranteed and should be carefully taken care of, a high repulsive force is needed to avoid obstacles. It is a robust method, operations continue always.

Virtual forces
Originally virtual forces differed from the potential field method because it used a certainty grid. Nowadays, this grid is not used anymore and thus the two methods are equal. Therefore, the virtual force method has the same evaluation as the potential field method.

DEFT
This method is complete and optimal [3]. The controller is computational intensive, a complex algorithm is implemented. The time horizon of the planner should be large enough to plan a route to the destination. Flexibility is well, but the amount of route options depend on the state of the system at time of planning. Safety is taken into account, but cannot be guaranteed in dense areas. Robustness of the controller is well, except when the central controller fails. Scaling is possible, but increases computational times.

ACO
This method is an incomplete one, it might occur that newly explored paths do not reach the destination. Also the optimality is not that well, it takes quite some time to find new, better paths [35]. ACO is a good choice that balances between time complexity and solution quality [35]. However, on large scale systems, the time complexity is high and heuristics should be implemented [53]. Flexibility is reasonable, the ants can explore the entire terminal area, but once paths are found the flexibility decreases as those paths are taken. Safety is a criteria to take into account. ACO relies a lot on a central controller, thus robustness scores a minus. The advantage is that new paths will be constructed around a broken AGV in the field.

MPC
A MPC controller requires a path planning algorithm that is assumed to be complete. A path is planned for the considered time horizon. The optimality of the trajectories depend on the path planner and time horizon. MPC is a complex controller, constraints are converted to a (non-)linear programming problem to solve an objective function. This increases the computational burden. This method is however flexible, because each time step a new action is chosen that minimizes the objective function. Safety is expected to be well, the disturbances of the time step in the planning horizon are updated every time, such that the controller can respond to the most recent situation. Robustness is well because the system can keep operating under all types of conditions. Because it is a distributed controller, more AGVs can easily be added to the system that does not necessarily result in an increase of computational burden.

Table 25 - Overview of studied literature.

		Routing type		Equipment			Traffic control (Chapters 3.1 & 3.2.1)					Solution		Routing bjective
		Fixed path, dynamic routing	Free-ranging (Chapter 3.2)	AGV general (Chapter 3.2)	AGV on ACT (Chapter 3.1 & 3.2.1)	Number of AGVs	Collision avoidance	Deadlock avoidance	Zones	Time windows	Semaphores	Solution method (Chapter 3.1 & 3.2.3)	Free-ranging model (Chapter 3.2.1)	
(Evers & Koppers, 1996)	[2]	x			x	Not mentioned	x	x			x	Simulation	-	Not specified
(Young Bae et al., 2008)	[8]	x			x	24	x	x	x	x		Simulation	-	Optimize individual trajectory
(Bae et al., 2011)	[9]	x			x	16	x	x	x	x		Simulation	-	Optimize individual trajectory
(Kim et al., 1997)	[20]	x		x		3		x				Simulation, Banker's algorithm		*Prevent deadlock*
(Li et al., 2011)	[26]	x			x	120	x	x	x			Simulation	-	Optimize individual trajectory
(Tlig et al., 2012)[27]	[27]	x		x		100					x	Simulation, lane sharing algorithm		Optimize individual trajectory
(Li et al., 2010)	[29]	x			x	200	x	x	x			Algorithm	-	Optimize overall performance
(Smolic-Rocak et al., 2010)	[30]	x		x		Not mentioned		x		x		Simulation	-	Optimize individual trajectory
(Jeon et al., 2010)	[31]	x			x	15		x	x		x	Simulation	-	Optimize individual trajectory
(Shao et al., 2014)	[32]	x		x		18	x	x			x	Simulation	-	Not specified
(Taghaboni-Dutta & Tanchoco, 1995)	[21]	x		x		10	x				x	Simulation, incremental planning algorithm	-	Shortest path using incremental planning
(Ware & Su, 2016)	[38]	x		x		30	x		x		x	Simulation, prioritized path algorithm	-	Shortest path using incremental planning
(Duinkerken, Ottjes, et al., 2006)	[7]		x		x	50	x			x		Simulation	Social force	Optimize individual trajectory
(Choi & Zhu, 2011)	[14]		x		x	10	x					Simulation	Virtual force	Optimize individual trajectory
(Jan et al., 2008)	[28]		x	x		1	x		x			Simulation	-	Optimize individual trajectory
(Choi & Zhu, 2012)	[43]		x		x	10	x					Simulation	Social force	Optimize overall performance
(Borenstein & Koren, 1991)	[45]		x	x		1	x					Experiment	Vector field histogram	Not specified
(Duinkerken, Van der Zee, et al., 2006)	[3]		x		x	14	x	x				Simulation	Social force	Optimize individual trajectory
(Ferrera et al., 2017)	[54]		x	x		40	x	x				Simulation	PFM	Not specified, focused on navigation in dense area
(Hamzheei et al., 2013)	[47]		x	x		Not mentioned						ACO algorithm	ACO	Optimize individual trajectory
(Xin et al., 2014a)	[49]		x		x	2				x		Simulation	MPC	Optimize overall performance
(Xin et al., 2014b)	[50]		x		x	2	x			x		Simulation	MPC	Optimize overall performance
(Corman et al., 2016)	[51]		x		x	10	x			x		Simulation	MPC	Optimize overall performance
(Xin et al., 2015)	[52]		x		x	10	x			x		Simulation	MPC	Optimize overall performance
(Bähr et al., 2012)	[56]		x	x		Not mentioned	x	x				Simulation	Virtual force	Not specified

Appendix C - Terminal model (Chapter 4)

This Appendix presents information on the procedures and GUI and corresponds to Chapter 4.

Procedures in PDL

Container generator procedure
Repeat
  Create new container
  Draw a stack number from a uniform distribution
  Draw a QC number from a uniform distribution

All containers are generated before the ASC, AGV and QC procedures start. Each ASC has thus at all times a variety of containers in its stack to handle.

Request AGV procedure
Repeat
  Wait for a request from ASC
  Choose closest AGV from idle AGVs (in buffer)
  Choose closest AGV from empty riding AGVs (heading to buffer)
  If no AGV is found
    Standby, try again when new AGVs are idle or empty-riding

This procedure dispatches the AGVs. Both the AGVs in buffer and all AGVs heading to the buffer are examined such that the AGV closest and thus fastest to the ASC is chosen.
Depending on the routing type, a Dijkstra algorithm determines the shortest route of each AGV (traditional routing) or the Euclidean distance is calculated (free-range routing):

$$d(ASC, AGV) = \sqrt{(x_{AGV} - x_{ASC})^2 + (y_{AGV} - y_{ASC})^2} \quad \text{(C.1)}$$

In case of mixed routing, a summation of both distances is done when needed (empty-riding AGVs on paths heading towards an ASC).

Request Task procedure
Repeat
  Wait for a request from QC
  Wait for an AGV to wait at the QC
  Pick first AGV that has the correct QC destination, assign to QC
  If no AGV is found
    Standby

This procedure assigns an AGV to handle to the QC. AGVs are handled FCFS.

GUI

The user starts the simulation via a graphical user interface (GUI). There are some input boxes that the user must specify. These inputs concern the equipment behavior and overall simulation input.
The user can choose whether the AGVs drive with or without traffic control. In the latter case, the vehicles can drive over each other and thus always drive their shortest path.
Another option is to choose a deterministic container retrieval time of the ASC, the user must click the radio button and insert the deterministic time.

Numerical outputs, graphs and animation appear when the simulation starts running. The simulation stops when the simulation time is crossed.
Once the simulation is completed, the GUI remains open. TOMAS queue data can be accessed via a screen that opens automatically in the background of the GUI. The GUI looks as follows:

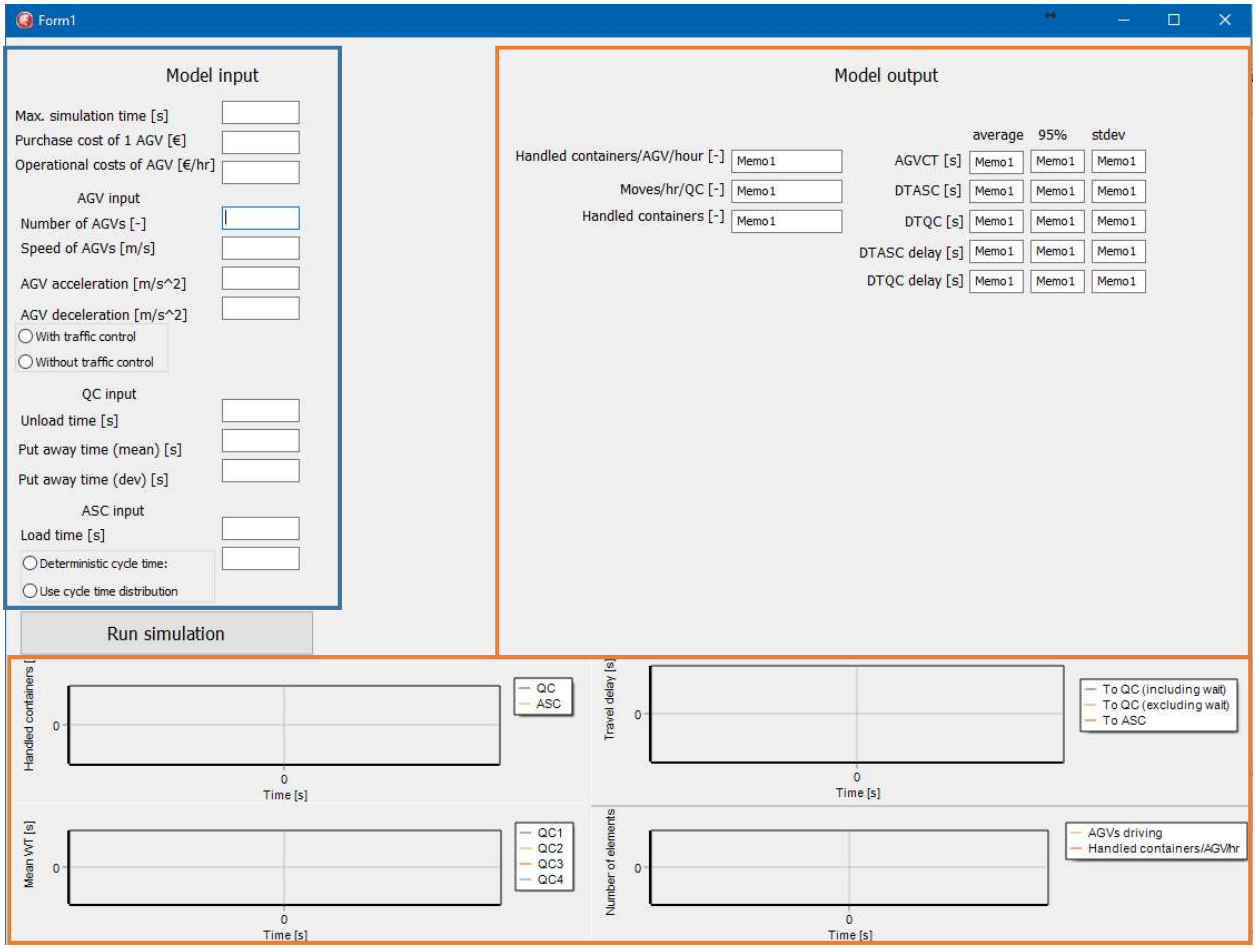


Figure 46 - GUI.

The input is shown by the blue rectangle, output by the orange rectangles.

Appendix D - Routing types (Chapter 5)

This appendix is relevant to Chapter 5 and shows the PDLS of routing components belonging to each routing type.

D.1 Traditional routing PDLs and Usage of buffer places

D.1.1 Traditional routing components

The following procedures and functions are explained:

- DijkstraInit Initializes TNetwork before a path is to be calculated
- DijkstraRoute Calculates the shortest, feasible path from Start to Destination
- getDriveTime Calculate the drive time from node to node, including acceleration, deceleration
- getDistance Calculates the distance between two nodes
- getCraneSpot Assigns a unload position to the AGV
- getBufferNode Assigns a buffer place to the AGV

```
Procedure DijkstraInit
For all nodes
    Make previous node nil
    Make distance infinite
```

```
Function DijkstraRoute
Make distance to start node zero
While something changed and less than 30 attempts are done
    For all nodes
        If there exists an arc up
            If node up is not occupied
                Calculate distance from start node to current node (getDistance)
                If distance < current distance to up-node
                    Update distance and route to up-node
        If there exists an arc down
            If node down is not occupied
                Calculate distance from start node to current node (getDistance)
                If distance < current distance to down-node
                    Update distance and route to down-node
        If there exists an arc left
            If node left is not occupied
                Calculate distance from start node to current node (getDistance)
                If distance < current distance to left-node
                    Update distance and route to left-node
        If there exists an arc right
            If node right is not occupied
                Calculate distance from start node to current node (getDistance)
                If distance < current distance to right-node
                    Update distance and route to right-node
Trace back from destination node to find first step of route from start-node
```

Function getDriveTime
If AGV does not move and AGV is not arrived at a crane
    DriveTime = 0
Else if AGV drives at full speed
    Calculate distance (using getDistance) and divide by AGV speed
Else if AGV decelerates or AGV is arrived at a crane
    DriveTime = deceleration distance/AGV speed

Function getDistance
Distance between two nodes = $abs(x_{destination} - x_{location}) + abs(y_{destination} - y_{location})$

Function getCraneSpot
Make starttime infinite
For all four unload spots
    If available time of spot < starttime and node is not claimed
        Update starttime with time of node
        Update unload node index
If no spot is found
    Make starttime infinite
    For all four unload nodes
        If available time of spot < starttime
            Update starttime of node
            Update unload node index
Update time of node

Function getBufferNode
Make DistanceToBuffer infinite
For all buffer places
    If buffer place is empty
        Calculate distance from start location to buffer place
        If distance < DistanceToBuffer
            Update DistanceToBuffer with distance
            Update buffer place index
Claim buffer place

D.1.2 Usage of buffer places

Usage of buffer places – AGV drives from ASC to QC

When the AGV travels from ASC to QC, it must at some point cross a buffer place to reach the QC. When being at a buffer place, the AGV checks if its destination node is occupied. If so, the AGV waits at the buffer place until the node is free. Each QC has a set of buffer places that the AGV can cross, these are highlighted in blue in Figure 47.

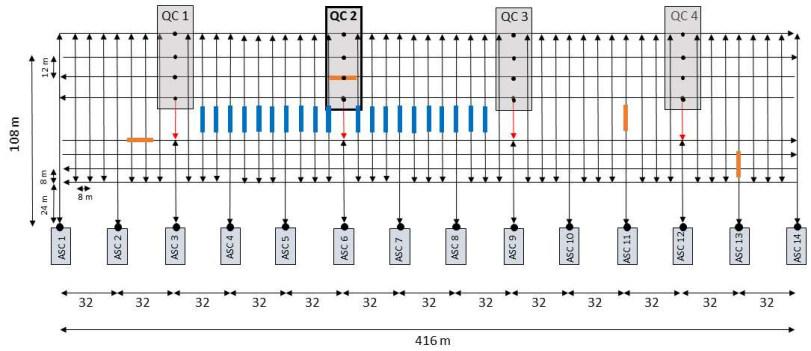


Figure 47 - Crossings and buffer places used when AGV drives from ASC to QC.

Figure 47 shows the example for QC2. It is prohibited for the AGV to drive across other QCs than its destination QC. This is done to make sure that the AGVs are not blocked by (waiting to be) unloaded AGVs at other QCs.

Usage of buffer places – AGV drives from QC to buffer

When the AGV can continue from the QC, a buffer place is chosen as new destination for the AGV. The chosen place is the closest, available, non-restricted place to the current location of the AGV.

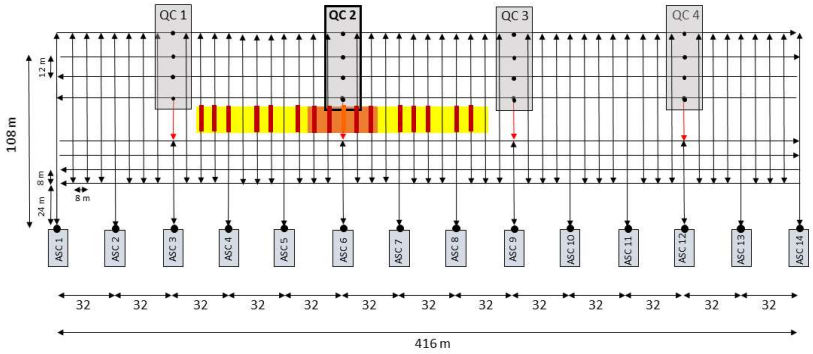


Figure 48 - Buffer places choices and crossing when driving towards ASC.

In practice, since there are plenty containers to handle, the buffer places closest to the QC will be chosen. These are highlighted by the yellow zone in Figure 48, the red ones are restricted to ensure crossings for the AGVs. The orange zone are the five places nearest to the QC in which the AGVs are discouraged to park.

Usage of buffer places – AGV drives from QC to ASC

When an AGV is reassigned while driving to the buffer, a buffer place must be crossed to reach its destination. The location of the crossing depends on the traffic and shortest path. There is no restriction that the AGV cannot drive along another QC. Waiting AGVs at those other QCs will however force the AGV to cross a buffer place.

In practice it is most likely that one of the places in the yellow zone is crossed. The red crossings are allowed to be used because the AGV will not go idle there.

D.2 Free-range routing PDLs and Usage of buffer places

D.2.1 Free-range routing components

- getCraneSpot calculates which unloading spot is first available for handling.
- getBufferNode determines which parking place should be chosen.

```
Function getCraneSpot
Make starttime infinite
For all four unload spots
    If available time of spot < starttime and node is not claimed
        Update starttime with time of node
        Update unload node index
If spot is found
    Update time of node and claim node
```

This function is basically similar to the function in traditional routing, except that it is now required that the unload spot is available. Also, the free-ranging function claims the node when one is chosen. This is not possible while traditional routing, because the corresponding node is then claimed and the AGV cannot reach its destination.

```
Function getBufferNode
Make DistanceToBuffer infinite
For all buffer places
    If buffer place is empty
        Calculate distance from start location to buffer place
        If distance < DistanceToBuffer
            Update DistanceToBuffer with distance
            Update buffer place index
Claim buffer place
```

D.2.2 Usage of buffer places

Usage of buffer places – AGV drives from ASC to QC

Using traditional routing, AGVs must cross a buffer place when they travel to the QC. This is not the case with free-ranging, by coincidence the AGV can cross over a buffer place. The buffer places in the second routing type are not used when the AGV travels from ASC to QC.
Instead, the distance from AGV to QC (QC x-location, 72) is calculated and when this distance ≤ 30 m, the AGV checks if its unload spot is free. If not, the AGV holds at its current location while constantly checking for a new available spot.

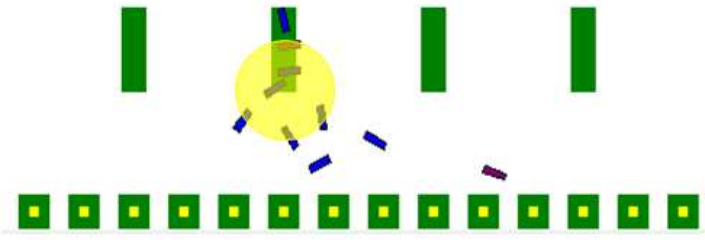


Figure 49 - Waiting AGVs to be routed to an unloading spot.

Figure 49 shows that 3 AGVs are held at a distance ≤ 30 m, the ones in the yellow area.

Usage of buffer places – AGV drives from QC to buffer

When the AGV can continue from the QC, a buffer place is chosen as new destination for the AGV. The chosen place is the closest, available, non-restricted place to the current location of the AGV. In theory, this thus can be any place that has not received weight factor infinite at initialization. Figure 50 shows the buffers chosen for AGVs leaving QC2.

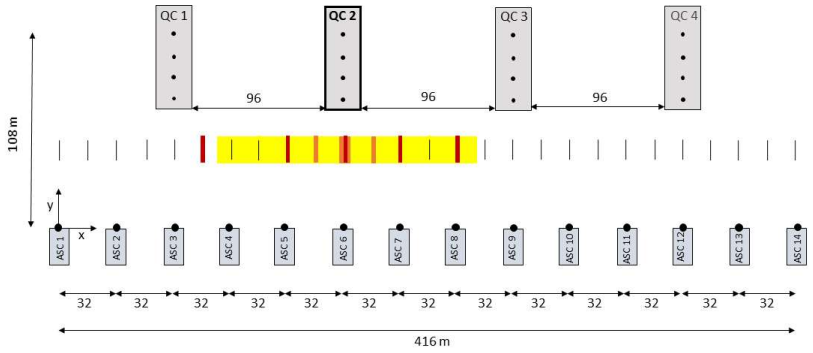


Figure 50 - Buffer place choices and crossings when driving towards ASC.

In practice the buffer places closest to the QC will be chosen, because there are plenty of containers to handle and thus the buffer places are empty most of the time. The places in the yellow zone are most likely to be chosen. The red places are restricted to park on to ensure crossings, these are determined at initialization and are depending on the amount of AGVs (in this example 16 AGVs). The three orange places are closest near the QC and are discouraged to park on to ensure crossings near the QCs.

Usage of buffer places – AGV drives from QC to ASC

When an AGV is reassigned to a ASC while driving towards the buffer, the most direct and feasible direction is chosen. While driving, the AGV may or may not cross a buffer place. The AGV most likely crosses the yellow zone of Figure 50.

D.3 Mixed routing PDLs and Usage of buffer places

D.3.1 Mixed routing components

- getCraneSpot calculates which unloading spot is first available for handling. This is called when the AGV is in the traditional routing mode.
- getBufferNode determines which parking place should be chosen. This function is called when the AGV is in the traditional routing mode, either when travelling to the buffer or when reassigned to an ASC. In the latter case, the parking place contributing to the shortest path must be chosen as transfer point.

```
Function getCraneSpot
Make starttime infinite
For all four unload spots
    If available time of spot < starttime and node is not claimed
        Update starttime with time of node
        Update unload node index
If no spot is found
    Make starttime infinite
    For all four waiting nodes
        If available time of unload spot < starttime and waiting node is available
            Update starttime of node
            Update unload node index
Update time of node
```

This function is an extended version of the one used in the second model. The difference is that when a unload spot is found, the corresponding node is not claimed yet. Else the AGV cannot reach its destination. The second difference is that when no unload spot is free, an available waiting spot (left to the unload spots) is chosen. Preferably, the waiting spot of the expected, earliest available QC spot is chosen.

```
Function getBufferNode
Make DistanceToBuffer infinite
Make DistanceToASC infinite
For all AGVs
    If not self and AGV is idle or waiting
        Calculate distance from AGV to buffer place
        If distance < 12
            AGV is near buffer place
For all buffer places
    If buffer place is empty
        Calculate distance from start location to buffer place
        If buffer place is a routing transfer point from free-range to traditional routing
            Make distance infinite
        If AGV is reassigned to a new job and must travel to ASC
            Add free-range distance (buffer place -> ASC) to distance
            If distance < DistanceToASC and no other AGV near buffer place
                Update DistanceToASC and buffer place index
        Else if AGV must go to buffer
            If distance < DistanceToBuffer and no other AGV near buffer place
                Update DistanceToBuffer with distance
                Update buffer place index
Claim buffer place
```

The difference of this function with the functions in the first model is that a check is performed if other AGVs are near a possible buffer place. Also, when reassigned to an ASC, the shortest path is determined based on shortest path while routing and free-ranging, the corresponding buffer place (functioning as transfer point) is chosen. This may differ from the choice of buffer place when the buffer is the AGV its destination.

D.3.2 Usage of buffer places

Usage of buffer places – AGV drives from ASC to QC

The buffer place that functions as routing transfer point is always crossed when an AGV travels from ASC to QC. Each QC has it own routing transfer point.

Usage of buffer places – AGV drives from QC to buffer

After unloading, the AGV choses the nearest available buffer place, thereby making sure no free-ranging AGVs are near that buffer place. The four routing transfer points can never be chosen. All other buffer places can be chosen, but since most places are empty (there are plenty of containers to handle), the nearest ones will be chosen. Figure 51 shows an example for QC2.

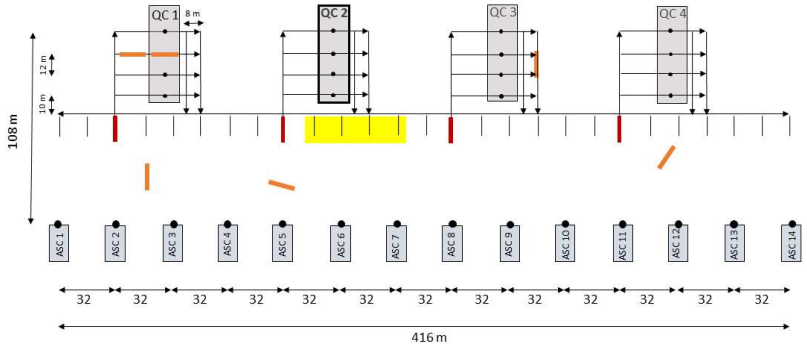


Figure 51 - Buffer place choices and transfer points when driving towards ASC.

Usage of buffer places – AGV drives from QC to ASC

When an AGV is reassigned to an ASC and still driving on the paths, the shortest route of combined traditional routing and free-ranging towards the ASC is determined. The buffer place, or now functioning as transfer point from traditional routing to free-ranging, that contributes to the shortest path is chosen as destination for traditional routing.

In practice, one of the buffer places in the yellow zone of Figure 51 will be chosen in case of leaving QC2.

Appendix E – Verification and validation (Chapter 6)

This appendix presents all results belonging to Chapter 6.

E.1 Traditional routing and terminal model

E.1.1 Verification of traditional routing terminal

Terminal model checks

The following general checks should be passed:

- The (average and current) sum of DrivingQ and AGV2Q should be equal to the amount of AGVs in the system.
- The maximum length of the AGV2Q, AGVQ, DrivingQ, DriveToBufferQ, DriveToASCQ and DriveToQCQ may not exceed the amount of AGVs in the system.
- AGVs can only enter and exit the terminal via the buffer. A new container cannot be picked up when the AGV has not been pass the buffer yet.
- The sum of the average length of the QCQ and QCbusyQ should be equal to the amount of QCs in the system (4).
- The sum of the average length of the ASCbusyQ and ASCQ should be equal to the amount of ASCs in the system (14).
- The maximum waiting time of DriveToASCQ cannot exceed 100 seconds, DriveToQCQ cannot exceed 110 seconds and DriveToBufferQ cannot exceed 14 seconds.
- Containers should always follow the order: ContainerQ and ContainerASC[#]Q, DriveToQCQ, AGVQ,QCbusyQ.
- An AGV cannot pick up more than 1 container. An ASC and QC can only handle one container at the time.

Run-time visualization

Four graphs are chosen as output of the simulation model. These are results from a simulation run that lasts 50000 and in which there are 40 AGVs in the system.

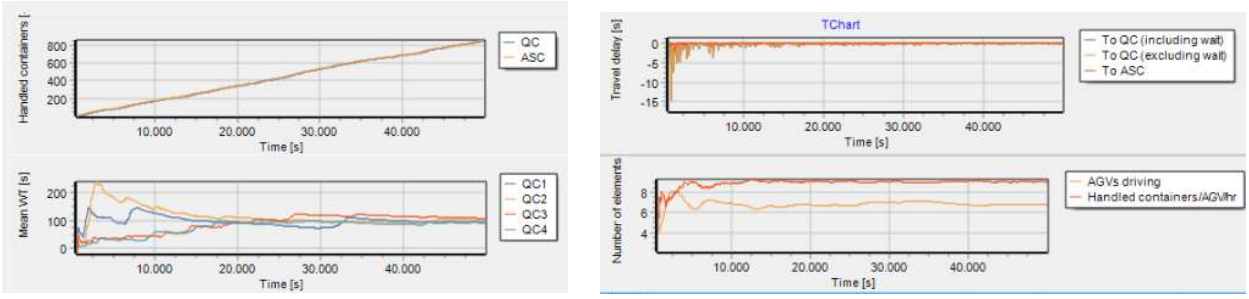


Figure 52 - Model output.

The first graph shows the amount of handled containers in time for the ASC and QC system. Both lines show a linear behavior, the ASC line just before the QC as is correct.
The second graph shows the average waiting time per QC in time. It is expected that a steady state sets in, this state is reached after 35000 seconds.
The third graph shows the travel time delay to ASC and QC. During the travel to QC the AGV can be held at a buffer place when its end node is occupied, this is meant with “wait”. This will not happen in the verification runs as traffic control is off. The delays are zero, which is correct.

The fourth graph shows the amount of AGVs driving in the terminal (yellow line) and the average amount of containers handled per AGV per hour (orange line). Both reach a steady state after 15000 seconds.

Test runs verification

Seed independence

Five tests are done in which the seed are varied for the five distributions in the model. The (random) seeds inputs are:

Table 26 - Seed independence tests.

	Inter arrival time	Container Stack number	Container QC number	QC put away time	ASC cycle time	Containers handled [-]
Test 1	8	1679	297	[QC#]+278	[ASC#]+488	851
Test 2	351	1884	1605	[QC#]+559	[ASC#]+403	796
Test 3	240	1354	1640	[QC#]+494	[ASC#]+889	817
Test 4	1962	369	99	[QC#]+143	[ASC#]+556	809
Test 5	125	1549	1684	[QC#]+21	[ASC#]+1862	894

No irregularities were found using different seeds in the simulation. The amount of handled containers varies with a maximum of 98 containers. On average 833 containers enter the system during simulation time. Different seeds do not lead to widely varying results.

Consistency checks

Utilization is calculated with: $\rho = \frac{\lambda}{s\mu}$ in which λ is the mean arrival rate, s the amount of cranes and μ the mean service rate. Doubled (halved) capacity should result in halved (doubled) utilization.

Table 27 - Consistency tests.

	Number of ASCs	Number of QCs	Utilization (cycletime) of QC system	Utilization of ASC system	Remarks
Base	14	4	0.77 (138.75)	0.33 (206.83)	-
Test 1	7	2	1.53 (138.00)	1.39 (437.52)	No steady state will be reached
Test 2	14	4	0.38 (69.00)	0.17 (105.00)	Handle times are halved (loading: 5 sec, unloading: 9 sec, put away: 60 sec)
Test 3	7	2	0.77 (69.00)	0.33 (105.00)	Handle times are halved (loading: 5 sec, unloading: 9 sec, put away: 60 sec)

The tests show what is to be expected. A double capacity (half handle times) of the cranes results in halved utilization (Test 2). Halving the capacity (half amount of cranes) results in a doubled utilization. An exception is the utilization of the ASC system in Test 1. Compared to the base case is the utilization quadrupled. This is because the ASC system does not reach a steady state.

Verification with analytical results

The M/D/1, a queueing theory model, is applicable to the model. Deterministic times are assumed.

Drive time

The average driving distances from buffer to ASC, from ASC to QC and from QC to buffer is determined experimentally in MATLAB. The shortest distance from each starting location to destination is calculated, a randomizer draws an infinite amount of drives and the averages are calculated:

Table 28 - Average drive times and distances of AGV.

	Distance [m]	Travel time [s]
Buffer to ASC	90.00	22.50
ASC to QC	216.45	54.11
QC to buffer	36.00	9.00

Steady state

In the system, one container arrives every 60 seconds. For the ASCs this means that 1 container arrives on average every 840 (14*60) seconds. At the QCs, this number is 240 (4*60) seconds. Steady state condition of both the crane types is fulfilled:

$$\lambda_{ASC} < s \cdot \mu_{ASC} \qquad \frac{1}{60} < 14 \cdot \frac{1}{210} \tag{E.1}$$

$$\lambda_{QC} < s \cdot \mu_{QC} \qquad \frac{1}{60} < 4 \cdot \frac{1}{138} \tag{E.2}$$

Results and discussion of M/D/1 and deterministic simulation model

The mathematical models can calculate the queueing time and length, both including and excluding service time. The M/G/1 and M/D/1 model are calculated with:

$$L_q = \frac{(\frac{\lambda}{\mu})^2}{2(1-\frac{\lambda}{\mu})} \tag{E.3}$$

$$L = L_q + \frac{\lambda}{\mu} \tag{E.4}$$

$$W_q = \frac{L_q}{\lambda} \tag{E.5}$$

$$W = W_q + \frac{1}{\mu} \tag{E.6}$$

In which λ represents the inter arrival time and μ the service time.

Table 29 - M/D/1 and simulation model results for ASC and QC systems.

	ASC M/D/1	QC M/D/1	ASC Simulation	QC Simulation
Lq [-]	0.04	0.39	0.04	0.43
L [-]	0.29	0.96	0.31	1.01
Wq [s]	35.00	93.40	34.15	99.28
W [s]	245.00	231.35	244.15	237.28

Lq represents the length of the queue, *L* represents the length of elements including servicing. *Wq* represents the waiting time of the queue excluding service time, *W* the waiting time including service.

ASC

The M/D/1 model results are very close to those found in the simulation. The queue time differ less than a second. The analytical model and simulation model differ somewhat in strategy; the analytical model assumes FIFO. The simulation model choses the container that must be transported to the QC with at this moment the smallest waiting queue. Despite this strategy difference, the total time spend in the system shows good agreement.

The simulation model shows that the ASC receives 1 container every 800.73 seconds. This is +-40 seconds below the input of one container every 840 seconds.

QC

The elements in queue and system show good agreement between the M/D/1 model and the simulation model. The waiting time of the queue is +-6 seconds higher in the simulation model than the M/D/1 model.

The arrival rate of containers is 234.33 seconds per container in the simulation models, this is close to the analytical input of 1 container every 240 seconds.

Cycle times and discussion per element

The model results and input data is used to calculate the cycle times per element. The elements in the system are containers, ASCs, QCs and AGVs.

Container process:

Table 30 - Results of the M/D/1 model and deterministic simulation model for the container process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	34.15
Cycling in stack	200.00	200.00
Wait for AGV	0.00	0.00
Load container onto AGV	10.00	10.00
Driven to QC	54.11	60.38
Wait for QC	93.40	99.28
Unload container from AGV	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**530.51**	**541.81**

The cycle times of the M/G/1 model and the simulation model differ 11 seconds. The drive time to QC and waiting time at QC are both +- 6 seconds above the analytical results. Drive time can be explained with the fact that in the analytical model only the lowest unload spots are assumed. The simulation model also sends AGVs to the other unloading spots.

Process of ASC:

Table 31 - Results of the M/D/1 model and deterministic simulation model for the ASC process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	34.15
Cycling	200.00	200.00
Wait for AGV	0.00	0.00
Load container	10.00	10.00
Total cycle time	**245.00**	**244.15**

The ASC process shows good agreement between both models. The only difference lays in the waiting time for the ASC to start handling the container. This might be due to the strategy difference, although a seed can also be the cause.

Process of QC:

Table 32 - Results of the M/D/1 model and deterministic simulation model for the QC process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for AGV	0.00	0.00
Wait for QC	93.40	99.28
Unload container	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**231.40**	**237.28**

The deterministic simulation has a cycle time that is 6 seconds longer than the analytical cycle time. This difference is most likely caused by the chosen seed.

Process of AGV:

Table 33 - Results of the M/D/1 model and deterministic simulation model for the AGV process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Drive to ASC	22.50	22.19
Wait for ASC	177.50	177.81
Load container	10.00	10.00
Drive to QC	54.11	60.38
Wait for QC	93.40	99.28
Unload container	18.00	18.00
Drive to buffer	9.00	8.97
Total cycle time	**384.51**	**396.63**

The AGV cycle times differ between the simulation model and M/D/1 model. The drive time to QC and waiting at the QC are the causes. Drive time to QC is larger because in the analytical result only the lowest unloading spot is assumed. The cause of the waiting time might be due to a seed that generated multiple containers in a row that have the same QC destination.

Hand trace events

Container process:

The interarrival time is 20 seconds higher than the average interarrival time, the interarrival times differ greatly. The stack distribution is not that uniform, stack 4 is often chosen while others are not chosen. The QC distribution is also not uniform; QC 3 has little containers to handle while QC 4 is busy. The non-uniform distribution might be due to an unfortunate seed choice.

The average waiting time of a container to begin handling is 52.6 seconds, this is 17 seconds longer than as in the analytical model (35.0 seconds). The ASC cycle time is 5 seconds shorter than the average of 197 seconds. Waiting time of the container and AGV for the QC is 35.9 seconds, this is about one minute shorter than as calculated in the analytical model (93.4). More than half of the AGVs that arrive at the QCs can be handled immediately, reducing waiting times significantly. Put away time is 8 seconds above the mean put away time.

Overall, the average completion time of the 20 containers is 502.25 seconds while the minimum completion time would be 408.46 seconds on average. About 1.5 minute is thus lost while waiting for a crane to begin handling. The analytical model calculated an average cycle time of 530.51 seconds. Both

cycle times lay relatively close to each other, given the fact that this hand trace only assumed 20 containers.

The AGV process:
The average distance to drive from buffer to ASC is 85.6 meters or 21.4 seconds, one second below the analytical results (22.5). It is assumed that the AGV leaves from the parking place below the QC unload spot (thus x=64, 160, 256 or 352), in the simulation these values are slightly more spaced (x=56, 60, 68, 152, 160, 168, 248, 256, 264, 344, 352 or 360). The waiting time at the ASC is on average 167.7 seconds, 10 seconds lower than calculated analytical result (177.5 seconds). The driven distance to QC is on average 240.8 meters, or 60.2 seconds, 6 seconds above the analytical result. The analytical model only assumed the lowest QC unload point, while the simulation uses all four. Waiting time at the QC is much lower than the analytical model: 35.9 seconds compared to 93.4. Many containers are handled immediately, while in the long-term simulation it appears that this is not always the case. The drive time to the buffer is assumed to be 12 or 44 meters, depending on the stack position compared to QC (right or left to QC, respectively).

Overall, the average cycle time of the 20 AGVs is 329.7 seconds per AGV, the analytical calculation had a cycle time of 384.5 seconds per AGV. The main difference is found in the shorter waiting time at the QCs. It is surprising that although QC4 is chosen often as destination, the corresponding containers can be handled immediately. In the long-term simulation, this will not always be the case.

Table 34 - Results of the hand event tracing of the container process. Shown left.

Table 35 - Results of the hand event tracing for the AGV process. Shown right.

Container process																Container	Container	Container
Container	Arrived	Inter arrival time	Stack	QC	ASC starts	Idle	Cycle time	Start loading	AGV arrived at QC	QC starts	AGV idle at QC	Container finished	Put away time	Min completion time	Completion time	Delay
1	35026,1	26,1	3	2	35211,82	185,72	165	35373,66	35432,82	35442	9,18	35592,66	132,66	371,66	566,56	194,9
2	35096,1	70	1	4	35096,1	0	135	35229,82	35356,75	35621,96	265,21	35797,53	157,57	430,57	701,43	270,86
3	35200	103,9	6	1	35200	0	195	35394,82	35460	35462	2	35632,66	152,66	417,66	432,66	15
4	35370	170	9	4	35371,89	1,89	135	35502,66	35565,89	35801,53	235,64	35941,39	121,86	326,86	571,39	244,53
5	35417,9	47,9	14	3	35417,9	0	405	35822,53	35901,31	35929,39	28,08	36127,31	179,92	670,92	709,41	38,49
6	35465,63	47,73	3	3	35468,52	2,89	195	35662,66	35742,66	35752	9,34	35921,31	151,31	444,31	455,68	11,37
7	35602,66	137,03	9	1	35606,53	3,87	165	35770	35857,31	35858,31	1	35953,39	77,08	336,08	350,73	14,65
8	35610	7,34	14	2	35832,53	222,53	225	36050	36159,54	36162	2,46	36329,2	149,2	484,2	719,2	235
9	35782,66	172,66	5	4	35786,39	3,73	135	35921,31	36008,56	36008,56	0	36144,31	117,75	358,75	361,65	2,9
10	35798,53	15,87	6	1	35798,53	0	135	35933,39	35988,56	35988,56	0	36130	123,44	328,44	331,47	3,03
11	35894,31	95,78	11	4	35894,31	0	255	36148,31	36191,54	36191,54	0	36290	80,46	393,46	395,69	2,23
12	36092,31	198	4	4	36094,2	1,89	225	36309,2	36420	36420	0	36558,06	120,06	459,06	465,75	6,69
13	36094,2	1,89	4	1	36137,31	43,11	165	36490	36529,29	36529,29	0	36669,06	121,77	340,77	574,86	234,09
14	36137,31	43,11	10	2	36329,2	191,89	255	36392,2	36455,94	36455,94	0	36603,06	129,12	462,12	465,75	3,63
15	36329,2	191,89	4	2	36504,2	175	105	36609,06	36662,2	36662,2	0	36782,63	102,43	273,43	453,43	180
16	36392,94	63,74	12	3	36394,29	1,35	135	36528,06	36594,29	36594,29	0	36753,63	141,34	346,34	360,69	14,35
17	36394,29	1,35	14	1	36394,29	0	105	36500	36629	36677,06	48,06	36789,6	94,54	333,54	395,31	61,77
18	36412,2	17,91	4	4	36619,06	206,86	195	36813,6	36910	36910	0	37088,8	160,8	469,8	676,6	206,8
19	36460	47,8	9	4	36468,63	8,63	255	36720	36782,63	36782,63	0	36907,67	107,04	432,04	447,67	15,63
20	36641	181	7	4	36642,67	1,67	255	36895,2	36971	37087,8	116,8	37250	144,2	489,2	609	119,8
Average [s]		**82,05**				**52,5515**	**192**				**35,8885**		**128,2605**	**408,4605**	**502,2465**	**93,786**

AGV process															AGV	AGV	AGV
Container	ASC starts	AGV arrives at ASC	Shortest path	Start loading	AGV idle at ASC	AGV arrived at QC	Shortest path	QC starts	AGV idle at QC	Unload finished	AGV at buffer	Shortest path	Min cycle time	Cycle time	Delay
1	35211,82	35229,66	15	35373,66	144	35432,82	46	35442	9,18	35460	35462,66	11	100	250,84	150,84
2	35098,66	35129,66	31	35229,82	100,16	35356,75	110	35621,96	265,21	35639,96	35652,66	11	180	554	374
3	35201,82	35218,82	15	35394,82	176	35460	42	35462	2	35480	35492,66	3	88	290,84	202,84
4	35371,89	35389,66	15	35502,66	113	35565,89	42	35801,53	235,64	35819,53	35827,53	11	96	455,64	359,64
5	35417,9	35446,9	31	35822,53	375,63	35901,31	58	35929,39	28,08	35947,39	35959,39	3	120	541,49	421,49
6	35470	35487	15	35662,66	175,66	35742,66	70	35752	9,34	35770	35772,66	11	124	302,66	178,66
7	35610	35627	15	35770	143	35857,31	66	35858,31	1	35876,31	35886,31	3	112	276,31	164,31
8	35834,53	35863,31	31	36050	186,69	36159,54	82	36162	2,46	36180	36191,54	3	144	357,01	213,01
9	35787,53	35808,53	23	35921,31	112,78	36008,56	78	36008,56	0	36020	36030	11	140	242,47	102,47
10	35799,53	35814,53	15	35933,39	118,86	35988,56	42	35988,56	0	36004,39	36011,56	3	88	212,03	124,03
11	35896,31	35917,31	23	36148,31	231	36191,54	30	36191,54	0	36200	36220	11	92	323,69	231,69
12	36094,2	36115,31	23	36309,2	193,89	36420	86	36420	0	36437,94	36451,94	11	148	357,74	209,74
13	36329,2	36350,32	23	36490	139,68	36529,29	26	36529,29	0	36547,29	36549,29	3	80	220,09	140,09
14	36139,31	36159,54	23	36392,2	232,66	36455,94	50	36455,94	0	36473	36480	3	104	340,69	236,69
15	36510	36531	23	36609,06	78,06	36662,2	38	36662,2	0	36679,06	36699,06	11	100	189,06	89,06
16	36394,29	36409,2	15	36528,06	118,86	36594,29	42	36594,29	0	36611,06	36625,2	3	88	230,91	142,91
17	36394,29	36424,2	31	36500	75,8	36629	106	36677,06	48,06	36695,06	36710	3	168	315,71	147,71
18	36619,06	36641	23	36813,6	172,6	36910	86	36910	0	36927,67	36929,67	11	148	310,61	162,61
19	36470	36485	15	36720	235	36782,63	42	36782,63	0	36800	36814,2	11	96	344,2	248,2
20	36642,67	36664,2	23	36895,2	231	36971	62	37087,8	116,8	37105,8	37119,96	11	124	477,29	353,29
Average [s]			**21,4**		**167,7165**		**60,2**		**35,8885**			**7,4**	**117**	**329,664**	**212,664**

E.1.2 Validation of traditional routing terminal

The Norfolk International Terminal has 3 QCs and 7 ASCs in use. 18 hostlers transport containers between stack and quay. Exact transportation area dimensions are missing but on average a hostler drives 457.2 meters per move, thus from quay to stack or vice versa. The terminal model as implemented has an average transport distance of 216.5 meters. To make the two comparable, the speed of the AGV is adapted in the terminal model. The hostler drives 7.6 m/s, one move thus takes ±60 s. To drive 60 seconds over 216.5 m, the AGV speed must be 3.6 m/s, Other equipment characteristics are shown in Table 36.

Table 36 - Equipment characteristics of Norfolk International Terminal.

Parameter	Value	Unit
Number of hostlers	18.0	-
Speed of hostler	7.6	m/s
Acceleration of hostler	*Unknown, assumed 1.0*	m/s^2
Deceleration of hostler	*Unknown, assumed 1.0*	m/s^2
Unload time of hostler at QC	20.0	s
Put away time of QC	60.0	s
Load time of hostler at ASC	50.0	s
Container retrieval time of ASC	150.0	s

A terminal simultaneously loads and unloads AGVs or hostlers, while the terminal model of this report only assumes the loading process. To compare the data from the Norfolk International Terminal with the terminal model, the following assumptions are made:

- An AGV is “unloaded” for the duration of two QC cycles at the QC: one cycle to place the unloaded container in the vessel, one cycle to retrieve a container from vessel and load onto AGV. The QC performance is doubled at the end of the simulation to obtain the moves/hr/QC. The AGV has thus an unload time of 160 seconds in which it cannot continue driving.
- An AGV is “loaded” for the duration of two ASC cycles at the ASC: one cycle to place the container in the stack, one cycle to retrieve a container in the stack and load onto the AGV. The AGV cannot leave until both cycles are completed.
- The cycle time of the ASC is not stated, only that the crane has a speed of 2.24 m/s. It is assumed that the stack has a length of 400 m. The half closest to quay is used for export containers, the other half is used for import containers. It is assumed that a container must be placed on average at $0.75xL_{stack}$ and is retrieved at $0.25xL_{stack}$. In total, $1.5xL_{stack}$ is then traveled for the import and export containers. A cycle also includes 50 seconds of (un)loading time and 15 seconds of crane aligning with stack. All included, it is assumed that the AGV must hold for 400 seconds at the ASC.
- NIT has a horizontal stack layout, while the terminal model has a vertical stack layout. A reduction of speed to correct for the average transport distance is made, assuming that the layout has no further influence. The terminal has one directional horizontal paths while the hostlers on the NIT can drive in any direction, this is assumed to have no influence.

Running these parameters, the terminal performance becomes 37.5 moves/hr/QC. At NIT, the terminal performance is 26 moves/hr/QC. This difference is most likely caused by a wrong ASC cycle time assumption, also the layout can be of influence (as is shown in their simulation results). It appears that 26 moves/hr/QC is reached when the ASC cycle time is 600 seconds, this relates to a stack of ±700 m. Since no more details are given, the inputs cannot be improved.

After validating their model, Liu et al. used other input parameters to test two yard layouts (horizontal and vertical). Their experiments were divided into loading process, unloading process and combined. Their input and results from the loading process and vertical stack layout can be used for the validation of the terminal model (Figure 53).

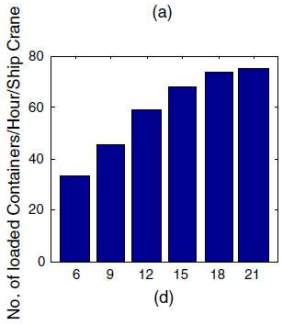


Figure 53 - Terminal performance as obtained by Liu et al. [60].

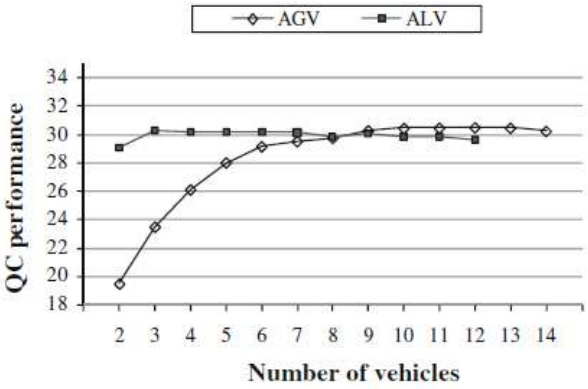


Figure 54 - Results from Bae et al. [9].

The influence of number of AGVs on the QC performance is tested, the QC capacity is assumed 75 moves/hr by Liu et al. Their model has 15 stacks, while the terminal model has 14 stacks.
The following input parameters are used:

Table 37 - Equipment characteristics of Norfolk International Terminal.

Parameter	Value	Unit
Number of AGVs	To be varied	-
Speed of AGV	3.36	m/s
Acceleration of AGV	*Unknown, assumed 1.00*	m/s^2
Deceleration of AGV	*Unknown, assumed 1.00*	m/s^2
Unload time of AGV at QC	12.00	s
Put away time of QC	36.00	s
Load time of AGV at ASC	50.00	s
Container retrieval time of ASC	195.00	s

AGV speed is the average of driving empty and driving loaded. The results are presented in Table 12.

Routing validation

A QC capacity of 31 moves/hr/QC is assumed. The following input parameters are used:

Table 38 - Input parameters as used by Bae et al. [9].

Parameter	Value	Unit
Number of AGVs	To be varied	-
Speed of AGV	3.00	m/s
Acceleration of AGV	0.64	m/s^2
Deceleration of AGV	1.55	m/s^2
Unload time of AGV at QC	40.00	s
Put away time of QC	76.00	s
Load time of AGV at ASC	40.00	s
Container retrieval time of ASC	Empirical distribution	s

AGV speed is assumed as an average of corner speed and forward speed. The cycle times of an ASC is a rough implementation of a figure presented by Bae et al.
Results of the terminal model and Bae et al. are presented in Table 13.

E.2 Free-range routing

E.2.1 Verification

The average driving distances from buffer to ASC, from ASC to QC and from QC to buffer is determined experimentally in MATLAB. The shortest distance from each starting location to destination is calculated, a randomizer draws an infinite amount of drives and the averages are calculated:

Table 39 - Average drive times and distances of AGV.

	Distance [m]	Travel time [s]
Buffer to ASC	73.23	18.31
ASC to QC	174.06	43.52
QC to buffer	40.00	10.00

Cycle times and discussion per element

The model results and input data is used to calculate the cycle times per element. The elements in the system are containers, ASCs, QCs and AGVs.

Container process:

Table 40 - Results of the M/D/1 model and simulation model for the container process, free-range routing.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	39.69
Cycling in stack	200.00	200.00
Wait for AGV	0.00	0.00
Load container onto AGV	10.00	10.00
Driven to QC	43.52	42.27
Wait for QC	93.40	107.70
Unload container from AGV	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**519.92**	**537.66**

The cycle times of the M/S/1 model and the simulation model differ +-20 seconds. Especially the waiting times at the QC cause the difference in cycle times. An explanation can be found in the average waiting time of an AGV per QC, see Figure 55. At the start of the simulation, many containers must be transported to QC3 which increases its waiting time up to a maximum of 350 seconds. Other QCs have at that moment an average waiting time of 50 or 80 seconds. The waiting time at QC3 then gradually decreases and QC1 and 4 reach a steady state. The waiting time at QC2 does not reach a steady state. At around 23000 seconds, the waiting time at QC1 suddenly increases 50 seconds and remains at that level. These increases point out the seed non-uniformity: within a small time frame many containers can be generated with the same QC destination which increases the average waiting time at the QC.

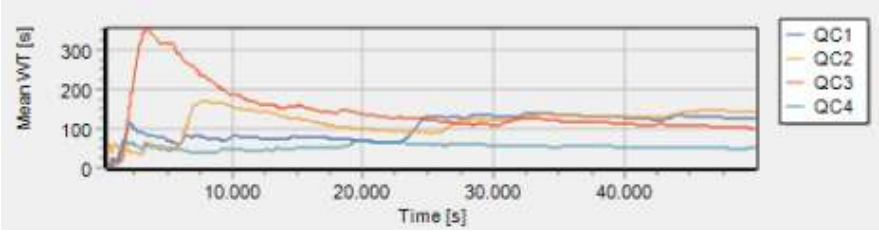


Figure 55 - Average waiting time per QC in verification run.

Process of ASC:

Table 41 - Results of the M/D/1 model and deterministic simulation model for the ASC process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	39.69
Cycling	200.00	200.00
Wait for AGV	0.00	0.00
Load container	10.00	10.00
Total cycle time	**245.00**	**249.69**

The ASC process shows good agreement between both models. The main difference lays in the waiting time for the ASC to start handling the container. This might be due to the strategy difference (see Chapter 6.1), although a seed can also be the cause.

Process of QC:

Table 42 - Results of the M/D/1 model and deterministic simulation model for the QC process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for AGV	0.00	0.00
Wait for QC	93.40	107.70
Unload container	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**231.40**	**245.70**

The difference in cycle times is the waiting time of the AGV and container before it starts handling. An explanation is given at the container process.

Process of AGV:

Table 43 - Results of the M/D/1 model and deterministic simulation model for the AGV process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Drive to ASC	18.31	16.22
Wait for ASC	181.69	183.78
Load container	10.00	10.00
Drive to QC	43.52	42.27
Wait for QC	93.40	107.70
Unload container	18.00	18.00
Drive to buffer	10.00	11.93
Total cycle time	**374.92**	**389.90**

The AGV cycle times of the simulation model is 15 seconds higher than that of the analytical model. The drive times differ slightly. This might be due to the destination “zone”, once the AGV is within this zone it is assumed to be arrived, this zone is often 3 meters in both x- and y-direction. The difference in waiting time at the QC is explained before.

Hand trace events

Container process:

The interarrival time of container is 22 seconds above the average. QC3 is chosen 9 out of 20 times, QC4 is only chosen 2 times. The average time that a container spends in the stack is 8.21 seconds, this is +-27 seconds less than the analytical result (35 seconds). The average ASC cycle time is 4 seconds shorter than the average input. Waiting time of the container until the QC starts handling is 22.02 seconds, this is much shorter than calculated (93.4 seconds). 14 of the 20 containers are handled within 10 seconds of arrival at the QC, this is unlikely to occur in the long-term simulation. Waiting time is building up at QC3, as is expected since it is chosen often in a short timeframe. Put away time 4 seconds less than the input mean value.

The container cycle took on average 408.3 seconds with an average waiting time of 33.8 seconds. This is quite low; the analytical cycle time is calculated as 519.92 seconds. The large interarrival time seems to be the cause of the low waiting times at ASC and QC.

The AGV process:

The drive time to ASC is less than calculated with the analytical model (15.6 versus 18.3). This is probably due to the “arrival zone”, the AGV has a tolerance of +-3 meters from its destination in which it is assumed to be arrived. The drive time to QC is 6 seconds above the analytical result, this can be explained with the fact that the analytical model does not assume decelerations and the fact that the lower unloading spots are chosen more often than the upper spots (the analytical model assumes an even spread over the four positions). The drive time to buffer shows good agreement.
Waiting time at the QC is very low, as is already explained in the container process.

The average cycle time of an AGV is 296.9 seconds, if there would be no waiting times this cycle would only be 105 seconds. The analytical result had a cycle time of 374.9 seconds. The difference is caused by the low waiting times, which are also not seen in the results of the long-term simulation.
Other components of the cycle show good agreement.

Table 44 - Results of the hand event tracing of the container process. Shown left.

Table 45 - Results of the hand event tracing for the AGV process. Shown right.

Container process																	Container	Container	Container
Container	Arrived	Inter arrival time	Stack	QC	ASC starts	Idle	Cycle time	Start loading	AGV arrived at QC	QC starts	AGV idle at QC	Container finished	Put away time	Min completion time	Completion time	Delay
1	35020,36	20,36	10	3	35029,33	8,97	135	35160,66	35195,33	35237,76	42,43	35375,9	120,14	295,47	355,54	60,07
2	35096,66	76,3	4	4	35096,66	0	165	35259,76	35337,75	35418,75	81	35556,75	120	387,1	460,09	72,99
3	35199,76	103,1	7	1	35200,9	1,14	165	35361,89	35412,9	35415,75	2,85	35468,52	34,77	263,76	268,76	5
4	35368,75	168,99	2	2	35371,89	3,14	225	35596,53	35643,89	35644,52	0,63	35782,89	120,37	399,01	414,14	15,13
5	35413,9	45,15	10	3	35418,31	4,41	225	35642,52	35677,52	35677,52	0	35815,89	120,37	388,58	401,99	13,41
6	35447,75	33,85	11	1	35468,52	20,77	165	35632,89	35695,52	35695,52	0	35851,26	137,74	502,37	503,77	1,4
7	35596,89	149,14	2	3	35606,53	9,64	165	35767,26	35833,26	35833,26	0	35894,54	43,28	282,01	297,65	15,64
8	35611,89	15	3	1	35611,89	0	165	35776,89	35810,89	35849,26	38,37	35998,39	131,13	336,13	386,5	50,37
9	35785,89	174	12	3	35785,89	0	195	35981,26	36015,54	36015,54	0	36137,94	104,4	341,77	352,05	10,28
10	35797,89	12	4	3	35797,89	0	405	36193,56	36241,56	36241,56	0	36397,2	137,64	595,31	599,31	4
11	35892,26	94,37	5	1	35894,54	2,28	105	35999,39	36032,54	36033,39	0,85	36171,39	120	276,85	279,13	2,28
12	36060,39	168,13	3	3	36094,2	33,81	225	36310,29	36382,2	36538,29	156,09	36672,94	116,65	414,74	612,55	197,81
13	36094,2	33,81	8	3	36095,29	1,09	225	36320,2	36349,2	36398,94	49,74	36537,29	120,35	393,26	443,09	49,83
14	36111,29	17,09	6	2	36137,94	26,65	225	36362,2	36396,2	36398,94	2,74	36537,29	120,35	384,61	426	41,39
15	36329,2	217,91	6	3	36372,2	43	195	36567,06	36612,82	36676,94	64,12	36814,94	120	378,86	485,74	106,88
16	36394,2	65	7	4	36394,2	0	105	36489,29	36555,29	36556,06	0,77	36694,94	120,88	295,97	300,74	4,77
17	36394,2	0	10	2	36394,2	0	135	36529,29	36576,06	36576,82	0,76	36711,29	116,47	301,56	317,09	15,53
18	36411,94	17,74	13	1	36411,94	0	165	36577,06	36659,29	36659,29	0	36810,78	133,49	396,61	398,84	2,23
19	36459,29	47,35	9	2	36468,63	9,34	255	36713,63	36750,63	36750,63	0	36905,67	137,04	442,04	446,38	4,34
20	36641,82	182,53	4	3	36641,82	0	225	36857,67	36903,67	36903,67	0	37059,17	137,5	415,35	417,35	2
Average [s]		**82,091**				**8,212**	**193,5**				**22,0175**		**115,6285**	**374,568**	**408,3355**	**33,7675**

AGV process														AGV	AGV	AGV
Container	AGV starts	AGV arrives at ASC	Shortest path	Start loading	AGV idle at ASC	AGV arrived	Shortest path	QC starts	AGV idle at QC	Unload finished	AGV at buffer	Shortest path	Min cycle time	Cycle time	Delay
1	35030,36	35044,36	16,5	35160,66	116,3	35195,33	26	35237,76	42,43	35460	35265,76	9,5	80	236,43	156,43
2	35096,66	35109,66	16,5	35259,76	150,1	35337,75	86	35418,75	81	35639,96	35447,75	9,5	140	351,09	211,09
3	35209,76	35224,76	16,5	35361,89	137,13	35412,9	50	35415,75	2,85	35480	35446,75	9,5	104	245,85	141,85
4	35375,9	35390,9	16,5	35596,53	205,63	35643,89	54	35644,52	0,63	35819,53	35673,52	9,5	108	301,63	193,63
5	35418,31	35432,75	16,5	35642,52	209,77	35677,52	26	35677,52	0	35947,39	35708,52	9,5	80	290,21	210,21
6	35556,75	35570,75	16,5	35632,89	62,14	35695,52	82	35695,52	0	35770	35747,52	9,5	136	279	143
7	35607,89	35622,89	16,5	35767,26	144,37	35833,26	78	35833,26	0	35876,31	35866,26	9,5	132	259,73	127,73
8	35612,89	35625,89	13,5	35776,89	151	35810,89	22	35849,26	38,37	36180	35891,26	9,5	73	279,37	206,37
9	35786,89	35799,89	13,5	35981,26	181,37	36015,54	42	36015,54	0	36020	36059,39	9,5	93	273,5	180,5
10	35798,89	35813,89	16,5	36193,56	379,67	36241,56	62	36241,56	0	36004,39	36287,56	9,5	116	489,67	373,67
11	35979,26	35993,26	16,5	35999,39	6,13	36032,54	34	36033,39	0,85	36200	36058,39	9,5	88	163,85	75,85
12	36095,29	36107,29	13,5	36310,29	203	36382,2	64	36538,29	156,09	36437,94	36570,29	9,5	115	476,09	361,09
13	36096,29	36110,29	16,5	36320,2	209,91	36349,2	30	36398,94	49,74	36547,29	36423,94	9,5	84	328,65	244,65
14	36164,39	36190,39	13,5	36362,2	171,81	36396,2	22	36398,94	2,74	36473	36429,29	9,5	73	291,35	218,35
15	36373,2	36399,2	13,5	36567,06	167,86	36612,82	46	36676,94	64,12	36679,06	36710,29	9,5	97	338,09	241,09
16	36394,2	36413,94	16,5	36489,29	75,35	36555,29	62	36556,06	0,77	36611,06	36590,06	9,5	116	195,86	79,86
17	36395,2	36409,2	16,5	36529,29	120,09	36576,06	50	36576,82	0,76	36695,06	36602,82	9,5	104	208,62	104,62
18	36412,94	36426,94	16,5	36577,06	150,12	36659,29	98	36659,29	0	36927,67	36700,94	9,5	152	289	137
19	36468,63	36481,29	13,5	36713,63	232,34	36750,63	42	36750,63	0	36800	36801,63	9,5	93	333	240
20	36642,82	36662,82	16,5	36857,67	194,85	36903,67	62	36903,67	0	37105,8	36949,67	9,5	116	307,85	191,85
Average [s]			**15,6**		**163,447**		**51,9**		**22,0175**			**9,5**	**105**	**296,942**	**191,942**

E.3 Mixed routing

E.3.1 Verification of mixed routing

The average driving distances from buffer to ASC, from ASC to QC and from QC to buffer is determined experimentally in MATLAB. The shortest distance from each starting location to destination is calculated, a randomizer draws an infinite amount of drives and the averages are calculated:

Table 46 - Average drive times and distances of AGV.

	Distance [m]	Travel time [s]
Buffer to ASC	83.77	20.94
ASC to QC	224.91	56.23
QC to buffer	54.00	13.50

Cycle times and discussion per element

The model results and input data is used to calculate the cycle times per element. The elements in the system are containers, ASCs, QCs and AGVs.

Container process:

Table 47 - Results of the M/D/1 model and deterministic simulation model for the container process, free-range routing.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	40.69
Cycling in stack	200.00	200.00
Wait for AGV	0.00	0.00
Load container onto AGV	10.00	10.00
Driven to QC	56.23	57.91
Wait for QC	93.40	109.28
Unload container from AGV	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**532.63**	**555.88**

The cycle times of the M/D/1 model and the simulation model differ +-23 seconds. The handling times show good agreement. Containers spend on average 5 more seconds in the stack than as calculated with the M/D/1 model, this can be due to a faster arrival rate (1/800 versus 1/840, simulation versus analytical model, respectively). The waiting times at the QC differ the most, it can be seen that not all QCs reach a steady state (Figure 56). The analytical model assumes a steady state.

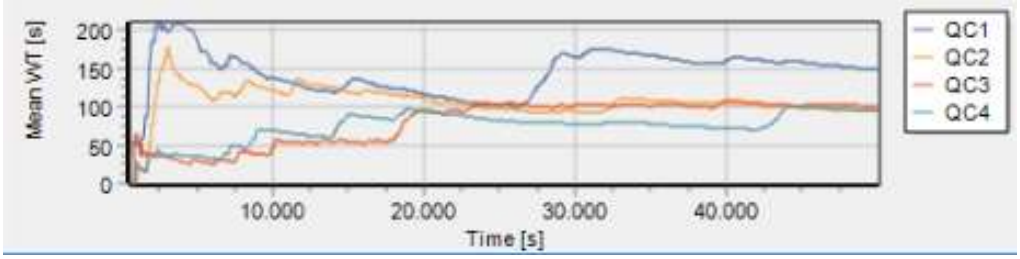


Figure 56 - Average waiting time of AGV per QC.

Process of ASC:

Table 48 - Results of the M/G/1 model, simulation model, M/D/1 model and deterministic simulation model for the ASC process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Wait for ASC	35.00	40.69
Cycling	200.00	200.00
Wait for AGV	0.00	0.00
Load container	10.00	10.00
Total cycle time	**245.00**	**250.69**

The ASC process shows good agreement between both models. The main difference lays in the waiting time for the ASC to start handling the container. The simulation waiting time is +- 5 seconds longer. This is probably caused by the faster arrival time of containers (1/800 seconds instead of 1/840 seconds).

Process of QC:

Table 49 - Results of the M/D/1 model and deterministic simulation model for the QC process.

	Analytical M/D/1	Deterministic simulation [s]
Wait for AGV	0.00	0.00
Wait for QC	93.40	109.28
Unload container	18.00	18.00
Put away at vessel	120.00	120.00
Total cycle time	**231.40**	**247.28**

The QC process shows a difference of 16 seconds between both models. The difference can be found in QC waiting time of the AGV. The waiting time does not reach a steady state and this increases the waiting time.

Process of AGV:

Table 50 - Results of the M/D/1 model and deterministic simulation model for the AGV process.

	Analytical M/D/1 [s]	Deterministic simulation [s]
Drive to ASC	20.94	17.88
Wait for ASC	179.06	182.12
Load container	10.00	10.00
Drive to QC	56.23	57.91
Wait for QC	93.40	109.28
Unload container	18.00	18.00
Drive to buffer	13.50	13.46
Total cycle time	**391.13**	**408.65**

The AGV cycle times differ by 17 seconds. The drive time to ASC is 3 seconds shorter in the simulation model, this can be due to the “arrival zone”. When the AGV is within 3 meters of its destination in both x- and y- direction, the AGV is assumed to be arrived. This reduces travel time. The difference in waiting time at QC is explained before.

Hand trace events

Container process:

The interarrival time of containers is 22 seconds above the mean input value. Stack uniformity is quite reasonable, although not all stacks are chosen during the hand trace. The QC uniformity is off: QC2 is chosen 8 out of 20 times, while QC1 is chosen two times.

In most cases, the stack handles the container within 5 seconds after it is generated. On average the waiting time in stack is 18.6 seconds, about half of the analytical result (35 seconds).

The waiting time of an AGV at the QC lays 13 seconds above the analytical queue time, which is quite reasonable. The put away time is almost 12 seconds above the input mean value.

The average container cycle time shows good agreement with the analytical result: 533.22 versus 532.63 seconds (hand trace and analytical, respectively).

Overall, the hand traced container process shows good agreement with the analytical result.

The AGV process:

The minimal drive times towards ASC, QC and buffer show good agreement with the analytical values.

Waiting of the AGV at the stack is 11 seconds above the analytical result, a clear cause cannot be given.

Waiting time at the QC is 13 seconds longer than calculated in the analytical model.

The average cycle time of AGV is 399.38 seconds, compared to 391.13 from the analytical result. This is a good result for a hand trace of only 20 containers.

Table 51 - Results of the hand event tracing of the container process. Shown left.

Table 52 - Results of the hand event tracing for the AGV process. Shown right.

Container process															Container		Container	
Container	Arrived	Inter arrival time	Stack	QC	ASC starts	Idle	Cycle time	Start loading	AGV arrived at QC	QC starts	AGV idle at QC	Container finished	Put away time	Min completion time	Completion time	Delay
1	35025,98	25,98	12	4	35029,33	3,35	315	35343,66	35385,83	35639,52	253,69	35742,52	85	460,775	716,54	255,765
2	35095,98	70	6	4	35098,66	2,68	195	35292,82	35359,33	35459,66	100,33	35632,89	155,23	455,78	536,91	81,13
3	35192,66	96,68	12	2	35353,66	161	495	35849,26	35936,33	36061,54	125,21	36155,52	75,98	673,955	962,86	288,905
4	35370,33	177,67	3	3	35371,89	1,56	225	35596,53	35670,89	35672,31	1,42	35822,52	132,21	444,76	452,19	7,43
5	35409,66	39,33	11	3	35418,31	8,65	255	35672,52	35726,31	35934,52	208,21	36067,56	115,04	442,84	657,9	215,06
6	35467,66	58	7	3	35467,66	0	165	35632,89	35672,52	35831,53	159,01	35937,39	87,86	313,635	469,73	156,095
7	35599,66	132	1	1	35606,53	6,87	225	35827,26	35876,33	35882,33	6	36064,56	164,23	450,005	464,9	14,895
8	35611,89	12,23	5	2	35611,89	0	225	35836,53	35876,33	35882,33	6	36054,56	154,23	437,73	442,67	4,94
9	35782,52	170,63	7	2	35786,39	3,87	105	35891,33	35945,39	36144,52	199,13	36279,2	116,68	287,805	496,68	208,875
10	35798,52	16	2	4	35798,52	0	195	35992,52	36095,54	36098,54	3	36283,99	167,45	481,125	485,47	4,345
11	35894,33	95,81	5	3	35894,33	0	165	36058,56	36116,54	36119,94	3,4	36216,2	78,26	316,06	321,87	5,81
12	36093,54	199,21	2	2	36093,54	0	105	36188,52	36249,2	36271,2	22	36447,91	158,71	336,51	354,37	17,86
13	36094,54	1	2	1	36137,94	43,4	195	36403,94	36449,91	36452,91	3	36543,91	73	326,5	449,37	122,87
14	36135,52	40,98	9	2	36209,2	73,68	195	36331,94	36392,94	36452,91	59,97	36623,2	152,29	427,34	487,68	60,34
15	36316,2	180,68	11	2	36329,91	13,71	105	36434,2	36513,91	36633,2	119,29	36750,17	98,97	299,195	433,97	134,775
16	36392,94	76,74	2	4	36413,94	21	195	36603,06	36705,2	36875,91	170,71	37078,41	184,5	498,175	685,47	187,295
17	36394,29	1,35	7	2	36394,29	0	165	36553,91	36613,06	36753,67	140,61	36959,17	187,5	418,625	564,88	146,255
18	36411,94	17,65	11	4	36444,2	32,26	165	36609,2	36655,2	36663,2	8	36873,91	192,71	416,21	461,97	45,76
19	36467,91	55,97	10	2	36467,91	0	165	36633,2	36698,63	36963,17	264,54	37106,91	125,74	378,29	639	260,71
20	36642,2	174,29	13	4	36642,2	0	105	36747,36	36793,91	37075,91	282	37222,17	128,26	299,385	579,97	280,585
Average [s]		**82,11**				**18,601**	**198**				**106,776**		**131,6925**	**408,235**	**533,22**	**124,985**

AGV process														AGV	AGV	AGV
Container	ASC starts	AGV arrives at ASC	Shortest path	Start loading	AGV idle at ASC	AGV arrived at QC	Shortest path	QC starts	AGV idle at QC	Unload finished	AGV at buffer	Shortest path	Min cycle time	Cycle time	Delay
1	35029,33	35044,79	13	35343,66	298,87	35385,83	32,775	35639,52	253,69	35657,52	35669,66	13,5	87,275	640,33	553,055
2	35098,66	35126,66	13	35292,82	166,16	35359,33	59,55	35459,66	100,33	35459,66	35476,66	13,5	114,05	378	263,95
3	35353,66	35368,33	13	35849,26	480,93	35936,33	74,975	36061,54	125,21	36079,54	36094,54	13,5	129,475	740,88	611,405
4	35371,89	35384,83	13	35596,53	211,7	35670,89	59,55	35672,31	1,42	35690,31	35708,31	13,5	114,05	336,42	222,37
5	35418,31	35433,66	15,275	35672,52	238,86	35726,31	44,8	35934,52	208,21	35952,52	35972,52	13,5	101,575	554,21	452,635
6	35467,66	35483,66	15,275	35632,89	149,23	35672,52	32,775	35831,53	159,01	35849,53	35859,26	13,5	89,55	391,6	302,05
7	35606,53	35625,89	20,625	35827,26	201,37	35876,33	32,775	35882,33	6	35900,33	35915,33	13,5	94,9	308,8	213,9
8	35611,89	35631,89	15,275	35836,53	204,64	35876,33	30,5	35882,33	6	35900,33	35912,33	13,5	87,275	300,44	213,165
9	35786,39	35812,52	15,275	35891,33	78,81	35945,39	38,125	36144,52	199,13	36162,52	36198,52	13,5	94,9	412,13	317,23
10	35798,52	35813,52	15,275	35992,52	179	36095,54	90,675	36098,54	3	36116,54	36135,52	13,5	147,45	337	189,55
11	35894,33	35915,33	15,275	36058,56	143,23	36116,54	44,8	36119,94	3,4	36137,94	36155,52	13,5	101,575	261,19	159,615
12	36093,54	36108,54	15,275	36188,52	79,98	36249,2	44,8	36271,2	22	36289,2	36316,2	13,5	101,575	222,66	121,085
13	36137,94	36152,52	15,275	36403,94	251,42	36449,91	30,5	36452,91	3	36470,91	36488,91	13,5	87,275	350,97	263,695
14	36209,2	36225,2	13	36331,94	106,74	36392,94	52,05	36452,91	59,97	36470,91	36482,91	13,5	106,55	273,71	167,16
15	36329,91	36347,94	15,275	36434,2	86,26	36513,91	67,225	36633,2	119,29	36651,2	36666,2	13,5	124	336,29	212,29
16	36413,94	36428,94	15,275	36603,06	174,12	36705,2	90,675	36875,91	170,71	36893,91	36910,91	13,5	147,45	496,97	349,52
17	36394,29	36416,94	15,275	36553,91	136,97	36613,06	38,125	36753,67	140,61	36771,67	36789,91	13,5	94,9	395,62	300,72
18	36444,2	36460,91	15,275	36609,2	148,29	36655,2	30,5	36663,2	8	36681,2	36699,91	13,5	87,275	255,71	168,435
19	36467,91	36483,91	15,275	36633,2	149,29	36698,63	59,55	36963,17	264,54	36981,17	36990,91	13,5	116,325	523	406,675
20	36642,2	36658,2	15,275	36747,36	89,16	36793,91	38,125	37075,91	282	37093,91	37113,91	13,5	94,9	471,71	376,81
Average [s]			**14,97375**		**178,7515**		**49,6425**		**106,776**			**13,5**	**106,11625**	**399,382**	**293,26575**

Appendix F – Data of results (Chapter 8)

This Appendix presents the data corresponding to the results of Chapter 8.

Costs and computational times

Table 53 – AGV lifetime costs for 95% terminal performance, corresponding AGVs and computational times.

QC capacity	Routing type	Costs [x10^6 €]	Number of AGVs [-]	Computational time [s]
30	*Traditional*	4.848	8	134.5
	Free-range	4.242	7	22.6
	Mixed	4.848	8	35.2
40	*Traditional*	6.060	10	156.8
	Free-range	5.454	9	20.9
	Mixed	6.060	10	42.1
50	*Traditional*	7.878	13	197.7
	Free-range	6.666	11	18.9
	Mixed	7.878	13	49.0
55	*Traditional*	9.090	15	243.9
	Free-range	7.878	13	26.7
	Mixed	9.090	15	62.6
60	*Traditional*	10.908	18	239.5
	Free-range	9.696	16	35.7
	Mixed	11.514	19	56.9
61	*Traditional*	11.514	19	244.3
	Free-range	10.302	17	29.2
	Mixed	12.726	21	60.1
62	*Traditional*	12.120	20	246.8
	Free-range	11.514	19	23.3
	Mixed	13.938	23	60.8
63	*Traditional*	13.332	22	271.7
	Free-range	12.726	21	23.8
	Mixed	-	-	-

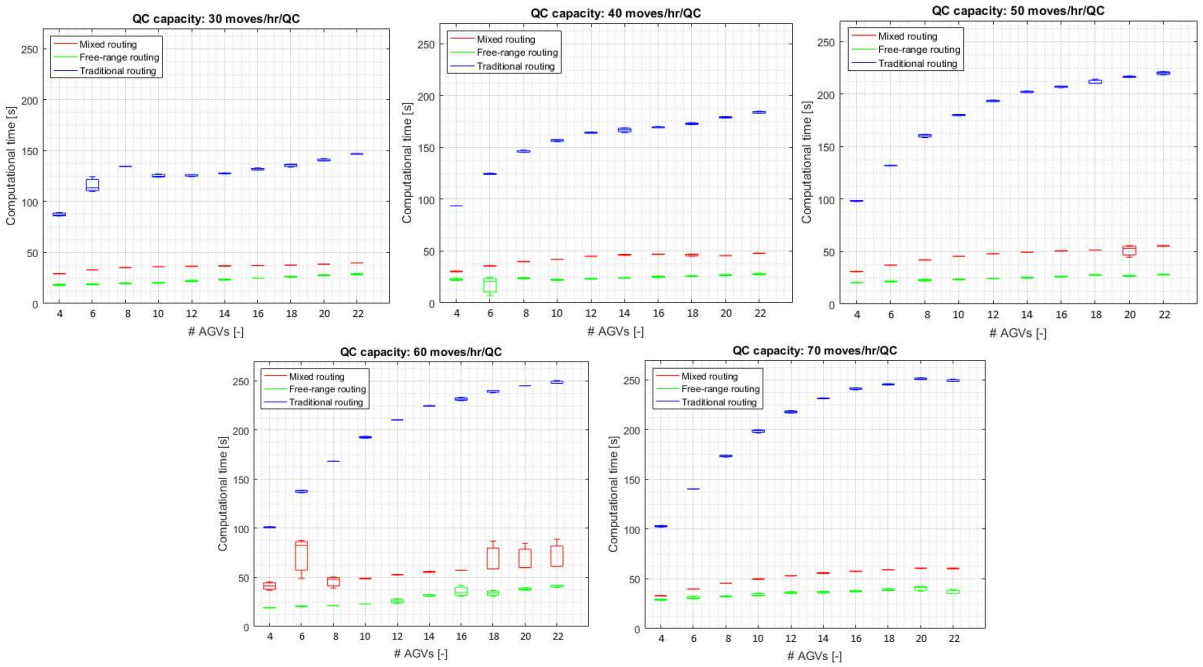


Figure 57 - Computational times for different QC capacities.

Terminal and routing performance

Table 54 - Terminal performance for different QC capacities.

Number of AGVs	30			40			50			60			70		
	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed
4	0.668	0.775	0.678	0.535	0.658	0.555	0.451	0.563	0.469	0.387	0.486	0.398	0.342	0.433	0.353
5	0.773	0.870	0.790	0.639	0.760	0.655	0.542	0.659	0.560	0.471	0.571	0.480	0.416	0.511	0.426
6	0.861	0.935	0.872	0.735	0.836	0.744	0.629	0.736	0.638	0.544	0.642	0.551	0.479	0.578	0.491
7	0.928	0.967	0.928	0.811	0.902	0.820	0.698	0.806	0.708	0.611	0.707	0.621	0.544	0.640	0.547
8	0.956	0.978	0.961	0.874	0.940	0.881	0.764	0.860	0.774	0.672	0.762	0.674	0.599	0.687	0.601
9	0.974	0.987	0.979	0.920	0.961	0.925	0.823	0.902	0.822	0.723	0.808	0.724	0.648	0.722	0.647
10	0.985	0.989	0.985	0.946	0.982	0.949	0.865	0.930	0.865	0.772	0.841	0.764	0.690	0.756	0.684
11	0.995	0.990	0.993	0.965	0.987	0.969	0.905	0.951	0.898	0.811	0.871	0.802	0.727	0.784	0.716
12	0.991	0.993	0.994	0.986	0.990	0.981	0.937	0.970	0.926	0.843	0.896	0.832	0.758	0.801	0.746
13	0.993	0.992	0.992	0.988	0.996	0.985	0.956	0.978	0.947	0.877	0.911	0.855	0.783	0.822	0.762
14	0.991	0.995	0.992	0.990	0.993	0.994	0.970	0.986	0.958	0.896	0.928	0.881	0.806	0.837	0.786
15	0.998	1.000	0.993	0.992	0.990	0.989	0.977	0.987	0.974	0.918	0.941	0.895	0.821	0.843	0.804
16	0.996	0.996	0.994	0.992	0.994	0.997	0.986	0.991	0.981	0.926	0.948	0.911	0.837	0.853	0.815
17	0.997	0.995	0.999	0.995	0.999	0.991	0.988	0.995	0.986	0.944	0.957	0.926	0.850	0.855	0.829
18	0.991	0.996	0.992	0.994	0.994	0.998	0.988	0.992	0.986	0.951	0.964	0.934	0.849	0.857	0.836
19	0.995	0.991	0.989	0.996	0.998	0.988	0.994	0.993	0.993	0.962	0.968	0.947	0.864	0.868	0.839
20	0.994	0.997	0.991	0.995	0.992	0.996	0.995	0.994	0.996	0.968	0.968	0.952	0.866	0.863	0.850
21	0.989	0.996	0.992	0.992	0.995	0.996	0.990	0.994	0.995	0.976	0.974	0.959	0.833	0.871	0.851
22	0.993	0.994	0.991	0.995	0.992	0.991	0.993	0.993	0.991	0.979	0.976	0.965	0.863	0.870	0.855
23	0.991	0.991	0.998	0.992	0.997	0.993	0.996	0.994	0.991	0.979	0.979	0.968	0.872	0.875	0.845

Table 55 - Routing performance for different QC capacities.

Number of AGVs	30			40			50			60			70		
	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed	Traditional	Free-range	Mixed
4	1.08	1.07	1.12	1.08	1.07	1.13	1.08	1.07	1.13	1.08	1.07	1.13	1.08	1.07	1.13
5	1.09	1.10	1.14	1.09	1.09	1.14	1.09	1.09	1.15	1.09	1.09	1.15	1.09	1.09	1.15
6	1.09	1.14	1.15	1.10	1.12	1.16	1.10	1.12	1.17	1.10	1.12	1.16	1.10	1.12	1.17
7	1.09	1.20	1.15	1.10	1.15	1.17	1.11	1.14	1.18	1.11	1.14	1.18	1.11	1.14	1.19
8	1.11	1.29	1.15	1.11	1.20	1.18	1.11	1.18	1.20	1.11	1.17	1.20	1.12	1.17	1.21
9	1.11	1.38	1.16	1.12	1.26	1.19	1.12	1.21	1.22	1.12	1.19	1.22	1.12	1.19	1.23
10	1.11	1.51	1.16	1.12	1.33	1.19	1.13	1.25	1.22	1.13	1.22	1.24	1.13	1.22	1.25
11	1.12	1.63	1.16	1.13	1.42	1.21	1.13	1.31	1.24	1.15	1.25	1.26	1.13	1.24	1.27
12	1.14	1.81	1.16	1.13	1.52	1.20	1.14	1.36	1.25	1.14	1.29	1.27	1.14	1.26	1.29
13	1.16	2.01	1.16	1.15	1.66	1.21	1.15	1.44	1.26	1.15	1.33	1.31	1.15	1.28	1.32
14	1.19	2.19	1.16	1.16	1.78	1.21	1.15	1.51	1.27	1.15	1.37	1.32	1.15	1.31	1.34
15	1.22	2.37	1.16	1.17	1.95	1.21	1.16	1.63	1.28	1.16	1.40	1.34	1.15	1.33	1.37
16	1.24	2.55	1.16	1.21	2.08	1.21	1.17	1.72	1.28	1.16	1.43	1.36	1.16	1.34	1.39
17	1.29	2.68	1.16	1.23	2.18	1.20	1.19	1.81	1.28	1.17	1.50	1.37	1.16	1.37	1.41
18	1.35	2.83	1.17	1.25	2.30	1.22	1.22	1.93	1.28	1.18	1.55	1.39	1.16	1.40	1.43
19	1.40	2.97	1.16	1.29	2.43	1.21	1.23	2.04	1.29	1.19	1.60	1.41	1.17	1.42	1.46
20	1.44	3.11	1.17	1.33	2.54	1.21	1.26	2.13	1.29	1.19	1.67	1.42	1.17	1.44	1.49
21	1.52	3.19	1.16	1.37	2.63	1.21	1.28	2.23	1.29	1.19	1.71	1.43	1.20	1.47	1.53
22	1.55	3.33	1.17	1.41	2.72	1.21	1.32	2.33	1.28	1.20	1.76	1.46	1.19	1.52	1.54
23	1.63	3.45	1.18	1.46	2.88	1.22	1.34	2.41	1.29	1.22	1.85	1.48	1.19	1.55	1.59