IMPROVEMENT OF DELIVERY TIMES BY DYNAMIC VEHICLE ROUTING IN AUTOMATED MATERIAL HANDLING SYSTEMS FOR SEMICONDUCTOR INDUSTRY

Fabian Böttinger, Christian Fischmann, Stefan Weidele, Roland Wertz

Fraunhofer Institute Manufacturing Engineering and Automation
Nobelstrasse 12, 70569 Stuttgart, Germany

boettinger@ipa.fraunhofer.de (Fabian Böttinger)

Abstract

In the last couple of years, a high level of automation was integrated in the factories of semiconductor industry. This implies the whole production process as well as the material delivery process. An automated material handling system (AMHS) is such an integrated automated transport solution of today's 300mm semiconductor waferfabs. These AMHS are based on a direct delivery approach (unified model) with alternative flexible storage systems to handle surges of lots in process or unplanned events. As the 300mm fabs are very large and the number of process and handling steps is rising, a fast and reliable transport system with short delivery times is required which, in addition, minimizes its footprint inside the expensive clean room. Currently, the implemented transport solutions are based on overhead hoist vehicles (OHV) which have to cope with several problems. On one hand there is an increasing traffic on the whole track system caused by the increasing number of process steps. On the other hand there is a higher number of hoist-actions as a result of new storage strategies. Instead of large central stockers, distributed tool assigned Under Track Storages (UTS) are used. Both issues mentioned can result in vehicles blocking track sections frequently. This can cause accumulations which lead to high variances in delivery times and make the delivery time for loads in this system less predictable. An additional effect might be the increased idle times of expensive process equipments, which need to be minimized.

This paper describes an approach of dynamic vehicle routing and identifies simulation based improvements of the AMHS performance. Furthermore different vehicle routing strategies are presented and compared.

Keywords: Dynamic Vehicle Routing, Fastest Path, Delivery Performance, AMHS Simulation, Transport Optimization.

Presenting Author's biography

Fabian Böttinger is working as a scientist at Fraunhofer Institute for Manufacturing Engineering and Automation IPA in Stuttgart. His key activities are automated transport systems and the simulation of processes in logistics and automation in the semiconductor and photovoltaic industry.

1 Introduction

The effective use of expensive material transport systems in a high technological, dynamic production environment requires a sophisticated system control. In semiconductor 300mm wafer fabs this material supply is called Automated Material Handling System (AMHS). To optimize the performance of the AMHS, an approach to improve the predictability of moves and actions in an AHMS while reducing the average delivery time for lots at the same time has been developed.
As a typical AMHS seeks the shortest path for routing its vehicles and lots from source to its destination, an algorithm was used. A decision is made, whether the shortest path - subject to traffic and actions taking place on the planned route - is also the fastest one. If at any junction there can be found an alternative path, it is made sure that the vehicle uses the fastest path to its destination. If necessary, vehicles are rerouted to an alternative path.
Therefore the AMHS controller uses a database which is updated dynamically with real-time data of all track segments in the AMHS. These updates are caused mainly by

- vehicles entering or leaving a defined track section
- number of vehicles (traffic volume) on track segments
- upcoming vehicle actions like hoist or park
- random unplanned events like downtime etc. can be taken into consideration.

The AMHS discussed in this paper has an unified system architecture similar to models of already existing wafer fab-segments with bays and parallel one way main line tracks.

For simulation and statistical evaluation this system for dynamic vehicle routing was built up as a simulation model in Applied Materials AutoMod™ 12.1 software. The database containing all real-time-data, track section information and the algorithm for calculating the fastest path to destination was realized in Java™.

The paper describes the new dynamic routing approach by comparing static models selecting the vehicle route by using the shortest path with models using dynamic vehicle routing algorithm. The derivation of rules for this dynamic routing is discussed as well as its influence on delivery times and their predictability for loads in this system.

2 Vehicle routing

The following chapters describe the current routing approaches in use and the dynamic routing algorithm developed and analyzed by Fraunhofer IPA.

2.1 Performance measurement

While analyzing delivery times, other metrics are of significance as well when evaluating the performance of an AMHS. These metrics discussed in this paper are according to Fischmann et al. [1] (among others):

- Transport Time
- Waiting Time (Wait for Vehicle)
- Utilization of Vehicles

These metrics are defined and calculated as explained in the following:

Delivery Time, Transport Time and Waiting Time

As shown in Fig. 1 the transport time is the amount of time a vehicle with a load on board needs to reach its destination and unload.
The waiting time can be seen from a tool's point of view after having finished processing a load. It is the amount of time it takes from calling a vehicle to this tool and loading the load on the vehicle.

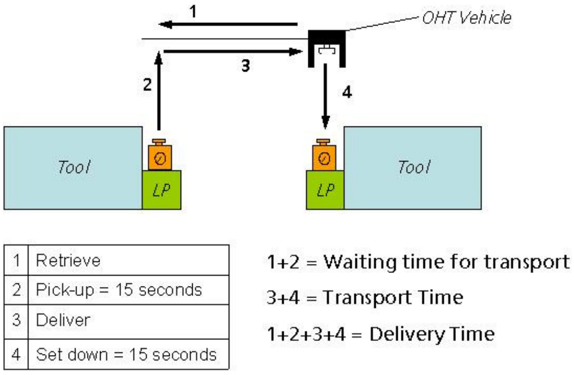


Fig. 1 Explanation of time segments

Transport time is the travelling time with load on board plus the unloading time.
The delivery time is the sum of transport- and waiting time.
These definitions are valid for all move types that occur in this simulation studies including moves to and from storages.

Vehicle Utilization is, following Sturm et al. [2], calculated as in Eq.1

$$Util_v = \frac{\sum_{j=1}^{J}(TFIN_{j,v} - TALLOC_{j,v})}{TSIM} \qquad (1)$$

where

v	Vehicle index
J	Number of Jobs for vehicle v
j	Index of jobs for vehicle v
$TFIN$	Finish time of job j for vehicle v
$TALLOC$	Time of allocation of vehicle v for job j
$TSIM$	Simulated time

In addition to AMHS related metrics, factory data like work in process (WIP), throughput and cycle time are of concern, as they help to analyze robustness and performance of the fab.

WIP is defined as the sum of all loads in the simulated system at a certain time *t* regardless of their current location.

Throughput (TP) is defined as shown in Eq. (2)

$$TP = \frac{\sum LPO}{TMON} \qquad (2)$$

where
LPO Lot passed Output in time span monitored
TMON Time span monitored

Cycle time is the difference between the time of load creation at the input and the time of leaving the system through the output. The average cycle time is calculated as in Eq. (3)

$$AVG\_CT = \frac{\sum_{i=1}^{I} CT}{I} \qquad (3)$$

where
i Index of the lot
I Number of all lots considered

To gain this data a logging mechanism is implemented into the simulation model writing periodically and event triggered timestamps, events and relevant information into log files that are evaluated and analyzed afterwards.

2.2 Shortest path routing

Finding the shortest path for a given source and destination location requires having all layout and path data like length, directions and crossings available. With this information there are a large number of algorithms that can be used to find the shortest route. Solving this problem for each possible node-node connection in a network leads to a static From-To Matrix which will be valid as long as there is no change in the layout. The static calculation works as follows: at the time, a vehicle gets a job, the shortest path to its destination is determined by two factors. Each part of the track has its own cost settings (factor 1). This coefficient is multiplied by the length of the respective track (factor 2). The decision, which route will finally be taken, is based on the product of all those extensions. Hence, the shortest path is determined up to fixed values which cannot take into consideration the dynamic behavior of an automated material handling system.
The matrix containing these calculated values will perform better in terms of simulation time than calculating the route on demand, as each combination has to be calculated only once.

2.3 Shortest time routing

In opposite to the shortest path the shortest time routing takes into account the physical behavior of vehicles travelling inside the network. Vehicles move with different speed values in straight lines and curves and need time for acceleration and deceleration. This approach leads to a more accurate static From-To Matrix and better results in delivery times but in return requires more effort in pre-simulation calculation.

2.4 Dynamic routing

2.4.1 Approach

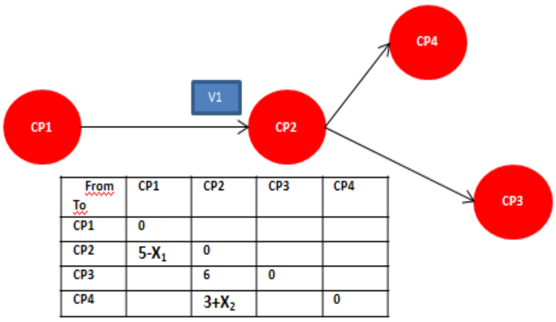
Both approaches, shortest path and shortest time are based on the assumption that a vehicle can travel on its route without any delay time. This represents reality in an inadequate way as in real fab operation the AMHS state changes frequently [3], resulting in delays on tracks e.g. caused by hoisting vehicles, vehicles with downtimes or several travelling vehicles in close proximity.
This can lead to route calculations which are not optimal in terms of travelling times and to less predictable arrival times of vehicles. To overcome this obstacle the dynamic routing approach which takes possible delay times into account while identifying travel routes has been developed.
A shortest From-To-Matrix of the whole networks control points (CP) as shown exemplary in Tab. 1 serves as a basis for this approach.

Tab. 1 FromTo-Matrix with travel times [s]

From To	CP A	CP B	CP C	CP D
CP A	0	10.5	1.5	35.3
CP B	5.3	0	22.3	1.7
CP C	15.3	8.5	0	8.6
CP D	10.3	2.5	25.1	0

Control points in this coherence are all tool load ports, stocker load ports, UTS and nodes of the track (path) network.

During simulation this matrix is updated constantly. Whenever a vehicle approaches any control point it re-calculates the route to its destination based on the current state of the matrix. Depending on the job which the vehicle has to execute in the next track segment it updates the matrix for it with a cost factor plus *X*. When approaching the next control point on its route it updates the track segment just passed with a cost factor minus *X*.



From To	CP1	CP2	CP3	CP4
CP1	0			
CP2	5-X_1	0		
CP3		6	0	
CP4		3+X_2		0

Fig. 2 Example of matrix update

In Fig. 2 vehicle V1 approaches control point CP2. A route is calculated from CP2 to the vehicles destination, in this example CP4. That is why the matrix (CP2,CP4) is updated with cost X_2.When arriving at CP2 the matrix (CP1,CP2) is updated with the negative cost X_1 it was updated with before. Approaching CP4 the value of matrix (CP2,CP4) is reduced by X_2.

2.4.2 Cost factor X

The cost factor X is the sum of constants which are defined before the simulation runs. In the current configuration there are two constants defined:

- X_{hoist}: a constant defining the additional cost caused by vehicles that will hoist (load or unload) at the related track segment
- X_{travel}: the additional cost caused by travelling at the related track segment

The cost constants used for the simulations, presented in this paper, are determined in a Design Of Experiments and are specified in chapter 4.1.

Additional cost factors e.g. for parking positions on tracks or down events of tracks and vehicles are not taken into account in this paper and might be considered in future research.

2.4.3 Drawbacks

Beside the drawbacks mentioned within Shortest Time routing the dynamic routing approach requires more simulation time due to its frequent route calculations on demand. The authors compensate this disadvantage for this network by a distributed IT architecture.

Additional challenges exist in determining appropriate cost X constants. Simulation experiments show a sensitive behavior of the whole system related to these constants.

2.4.4 Similar appliances

The dynamic routing approach mentioned can basically be compared with navigation systems typically found in automotive sector. These systems are able to calculate the shortest time routes and reroute a driver in case of traffic jams on its current route.

3 Set up of system

3.1.1 Layout

For testing, analyzing and comparing the results of the commonly used systems to our approach of dynamic routing, a simulation model in a unified architecture was set up.

The layout of this model as shown in Fig. 3 is related to sections of currently operating semiconductor fabs.

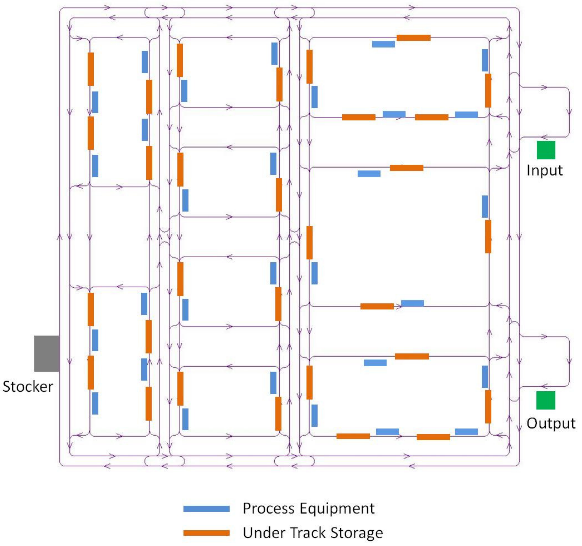


Fig. 3 Layout of the model used

Distributed over the whole system which has a dimension of 46x46 meters there are several tools with varying numbers of load ports. A defined number of Under Track Storages (UTS) are assigned to these load ports (compare [4]) in order to buffer incoming loads in case no free tool load port is available. These UTS are always located directly under the track close to the tool in front of its load ports.

Tab. 2 Tool settings

Tool Group	Tools	Load-ports (LP)	UTS	Process times /deviation [min]
1	5	3	4	8,9 / 2
2	8	3	3	30 / 9
3	2	2	6	4,6 / 0,4
4	7	2	4	15 / 4
5	6	2	4	10,5 / 3

Tab. 2 shows the number of available tools per tool group, the number of load-ports per tool and the number of assigned UTS per tool. In addition to this, one large scaled stocker is available to handle a temporary high number of loads in system, which can be used if no free UTS or tool load ports are available.

3.1.2 Load distribution

Loads are created with a uniform distribution at the input-queue where they enter the system. The start-rate of loads per hour is used as a mean; half of this rate is used as standard deviation to amplify system dynamics.

3.1.3 Vehicles

In order to distribute loads between input, tools, storages and output the system provides 25 vehicles. These vehicles have ten fixed parking-locations inside the system. The vehicles are going to these park-positions in case there are no jobs available at this moment. If any vehicle without a job blocks the track of a moving vehicle, it is forced to swap its current location with the location of the trailing, moving vehicle.
The vehicles are configured as follows:

- Speed (straight tracks): 3 m/s
- Speed (curve/turn): 1 m/s
- Acceleration / Deceleration: ±1.5 m/s²
- Hoist time (load/unload): 15 s

3.1.4 Downtimes

To simplify the comparison of a model with static vehicle routing to our approach of dynamic routing, downtimes for tools or vehicles have not been implemented.

3.2 Implementation of dynamic parts

The implementation of dynamic routing is divided in two parts:

- The first part at the beginning of the simulation runs is a short phase of initialization which is required for automated building of the From-To-Matrix.
- The second part deals with this matrix which has to be updated during the simulation whenever any vehicle enters or leaves a track segment.

The simulation model is separated from the path finding-algorithm. This allows changing the algorithm without having to adapt the source code in the AutoMod mode. Besides that any other more powerful and flexible programming language supporting socket connections can be used. An additional benefit of the separation is the possibility to use multi-core CPUs or additional hardware to improve the available computing power.

As algorithm to calculate the fastest path between a vehicles source and its destination, the Dijkstra-Algorithm has been implemented as it is very popular in today's route planner as well as IT-network-routing protocols.
The logic to find the fastest path uses the basic model data provided by the From-To-Matrix and is implemented in a Java-application. To link the AutoMod Simulation model with Java a socket-connection is established and used.

3.2.1 Initialization

During the initialization phase one vehicle travels along each track in the simulation model. Control points at any junctions subdivide these tracks in smaller segments. While traveling, the time between these control points passed by the vehicle is measured and stored in the From-To-Matrix. This basic matrix is valid until the model layout is changed.

3.2.2 Update and Path finding during the Simulation

As described in chapter 2.4, the From-To-Matrix is updated permanently, whenever a vehicle enters or leaves a track section. When arriving at a junction where the vehicle can decide between two or more different routes, the Dijkstra-Algorithm - using the current updated matrix - is executed to calculate the current fastest path for this vehicle. This allows all vehicles to avoid high traffic sections if the possibility of an alternative, reasonable route is given.

4 Execution of simulation experiments

As a reference model a simulation model without dynamic vehicle routing is set up. The ramp-up time is set to 10 hours, the simulation time to 10 days. Load start-rate is adjusted to obtain the models range of limited conditions:

- average vehicle utilization >= 60%
- average WIP-Rate during the considered simulation duration with steady behavior

This leads to an average input rate of 92.2 loads per hour. Based on the parameters determined in the static model, a Design Of Experiments (DOE) is started using a model with dynamic vehicle routing. Both, the additional cost caused by vehicles that hoist (load or unload) at any track segment (X_{hoist}) and the additional cost caused by vehicles travel on any track segment (X_{travel}) are modified to investigate their influence to the whole system and to find the optimum value for the simulation model.

4.1 Results

In the following chapter this paper discusses the results of the simulation studies. In general there are results for three different types of routing:

- Shortest path routing
- Shortest time routing
- Dynamic routing

4.1.1 Delivery time

As shown in Fig. 4, the X factor settings of the model affect the delivery times of the vehicles in the simulation model intensely. As expected before starting the DoE, the results comparing shortest path routing and shortest time routing are similar to each other. While the simulation run with shortest path routing, which is also used as reference model, shows an average delivery time of above 111 seconds, the simulation runs with different settings for X_{hoist} and X_{travel} have values around 103 seconds.

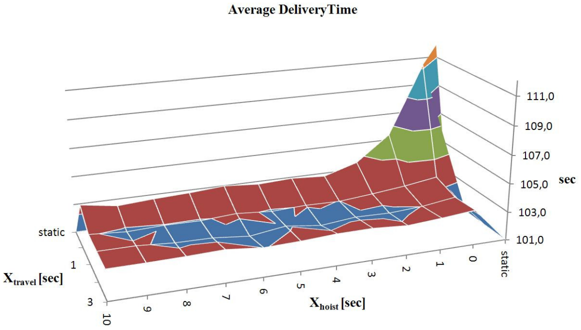


Fig. 4 Average Delivery Times

Comparing the shortest path routing approach to the best result of dynamic routing the improvement of time sums up to 8% in average. At the same time, the standard deviation on delivery times was reduced by 19 %. The settings for the best scenario in this configuration are:

- X_{hoist}: 6 seconds
- X_{travel}:2 seconds

These values are the basis for the results presented in the following.

4.1.2 Waiting Time

As the delivery time is the sum of waiting time and transport time, a closer look on these values is necessary.

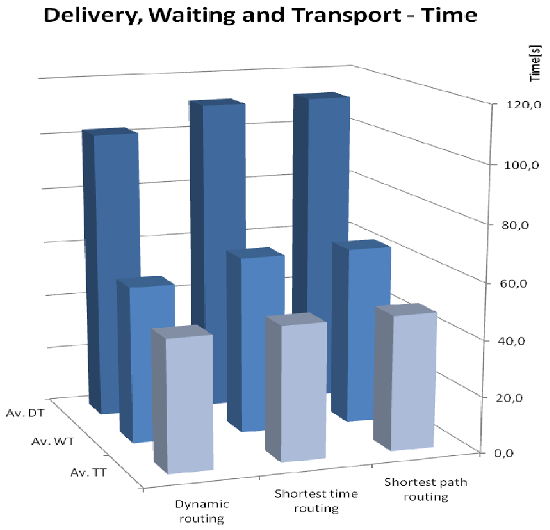


Fig. 5 Comparison of DT, WT and TT

Fig. 5 allows comparing the three different model settings. The diagram shows a significant improvement in waiting time of approximately 12 % for Dynamic Routing. Beside this, the measurement of standard deviation of waiting times shows an improvement of 29 %.

4.1.3 Transport Time

The results in Fig. 5 also show an average improvement of delivery times of about 3 % (standard deviation improves about 11 %).

The difference in improving the delivery times by 12 % and the transport times by 4 % only creates further questions. A possible explanation for this can be found by taking a look at the distances travelled for the different job types. While vehicles often have to travel large distances when they are allocated to a load which is to be picked up, the distance for delivering loads is quite short. This is particularly the case for dispatching a load from a tool dedicated UTS to a tool load port. In all cases these jobs have no alternative route at all. This, however, is intended to ensure equipment to be utilized all the time. Moreover this shows the potential that can be achieved with an intelligent empty vehicle balancing.

Additionally, these results can imply that the transport times might improve more when considering a larger network where more long distance moves are required. This could be e.g. inter-bay moves (tool to tool or tool to UTS) in a wafer fab.
Besides absolute values, the standard deviations of the key metrics are important. The achieved reduction of the deviation helps to gain more precise predictions of arrival times at tools and storages. This in return supports the planning process of dispatchers in the factory.

4.1.4 WIP

Fig. 6 shows the WIP in the whole system over time.

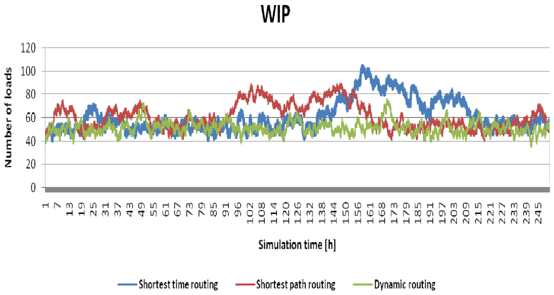


Fig. 6 WIP

For the dynamic routing the results show a lower and more uniformly distributed WIP with an average of 51 to 61 loads for both static variants. The lower WIP leads to a more constant usage of factory equipment and minimizes the usage of the additional implemented stocker which is only used to catch temporary high number of loads.

4.1.5 Vehicle Usage

In Fig. 7 a change in the total vehicle usage during the whole simulation is pointed out.

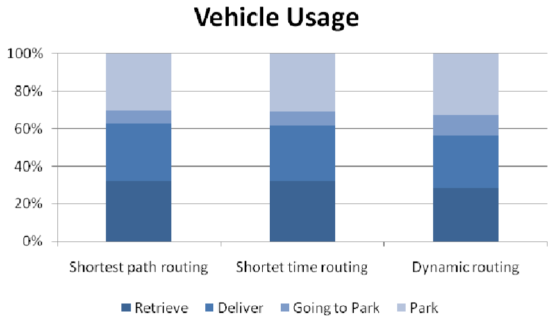


Fig. 7 Vehicle Usage

The vehicle utilization (retrieve-jobs + deliver-jobs) is reduced by 6 % in the Dynamic routing approach. This can be explained on the one hand by the reduced number of moves to and from the stocker and on the other hand by the reduced time needed for delivering and retrieving loads.

4.1.6 Cycle Time

As stated above the input rate is set to a 92.2 loads per hour. As all simulation scenarios are able to cope with this load factor in a stable state this directly leads to similar throughput values which are close to the input rate.

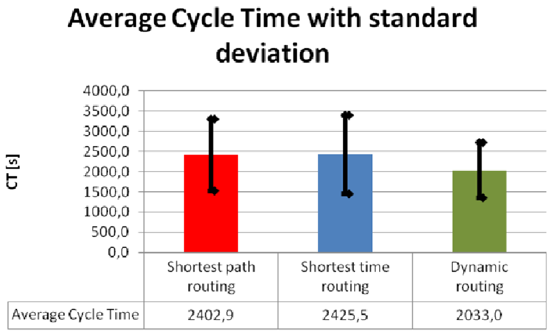


Fig. 8 Cycle time and standard deviation comparison

Analyzing the cycle times shows an improvement with the dynamic approach of about 15% in comparison to the shortest path scenario. Additionally the standard deviation can be reduced significantly.

When combining these results with the lower total vehicle utilization discussed in the preceding chapter a future investigation concerning throughput maximization is reasonable.

5 Conclusions

Due to the increasing complexity of automated material handling systems in semiconductor industry, the vehicle routing becomes a very critical point. The rising number of transports in modern 300mm fabs demands an effective routing of the vehicles in use. The overall fab performance depends on a sophisticated vehicle routing.
The shortest path is not always the fastest. Reasons for this might be congestions based on hoist operations executed by a leading vehicle or too many vehicles running over a branch or N-Shunt.
The approach developed by the authors points out that a dynamic evaluation of the fastest possible route instead of the static one helps to reduce the delivery time sustainable. To achieve this goal several facts are taken into consideration. Each time a vehicle is able to choose an alternative route (depending on the layout) the different possibilities are compared, based on the updated length of the track to destination, the density of that track and other time critical factors like hoist operations. Various additional experiments documented that a dynamic calculation of the route without updating the mentioned factors cannot improve the system performance lasting.
A closer look shows the main effect appearing at the waiting time. Also the transport time could be reduced; especially the average waiting time was decreased by more than 11%. In addition to that, the standard deviation could be reduced as well.
In future, the whole vehicle routing concept has to consider both: routing of empty vehicles and vehicles in operation in conjunction with a sophisticated empty vehicle distribution. This fact leads to the conclusion that a further investigation in advanced rerouting of empty and idle vehicles might reform the throughput of an automated handling system significantly.

6 References

[1] C. Fischmann, R. Muckenhirn, R. Sturm. Approach for Artificial Load Generation for Acceptance Test of AMHS. *Proceedings of the 12th IFAC International Symposium, Volume 2,* P. 685-690*, 2006.*

[2] R. Sturm. Modellbasiertes Verfahren zur Online-Leistungsbewertung von automatisierten Transportsystemen in der Halbleiterfertigung. *Dissertation, University of Stuttgart,* P. 82, 2006.

[3] M. Fukunari, S. Rajanna, R. J. Gaskins, M.E. Sparrow. Data-based node penalties in a path-finding algorithm in an automated material handling system. *Proceedings of the Winter Simulation Conference*. Volume 2, P. 1383-1386, 2002.

[4] K. Rust, N. Patel, P. Qu. Modeling Practical Methods of Material Storage in a 300 mm Fab. *IEEE International Symposium on Semiconductor Manufacturing*, P.24-28, 2005.