EUROPEAN COMMISSION

SEVENTH FRAMEWORK PROGRAMME
FP7-SST-2013-RTD-1
GA No. 605598

Cargo handling by Automated Next generation Transportation Systems for ports and terminals

Deliverable No.	Cargo-ANTs D2.5	
Deliverable Title	High-level simulation results	
Dissemination level	Public	
Written By	Maurice Kwakkernaat (TNO) Jos Elfring (TNO)	[08-01-2015] [25-02-2015]
Checked by		
Approved by		
Status	Draft	[08-01-2015]

Please refer to this document as:

Kwakkernaat, M. (2015) Cargo-ANTs Deliverable 2.5 "High-level simulation results" of the EU FP7 Project Cargo-ANTs (www.cargo-ants.eu)

Disclaimer:

Cargo-ANTs is co-funded by the European Commission, DG Research and Innovation, in the 7th Framework Programme. The contents of this publication is the sole responsibility of the project partners involved in the present activity and do not necessarily represent the view of the European Commission and its services nor of any of the other consortium partners.

Executive Summary

The Cargo-ANTs project aims to create smart Automated Guided Vehicles (AGVs) and highly Automated Trucks (ATs) that can cooperate in shared workspaces for efficient and safe freight transportation in main ports and freight terminals.

The objective of this deliverable is to demonstrate the AT and AGV capabilities in simulations. The results include simulations at different levels. The first level simulates the expected capabilities and operation of the two individual ANTs (AGV and AT). The second level simulates the operations of both ANTs combined at a higher level to show how performance and capacities of a terminal can be improved. The third level of simulations includes more technical details and focusses more on the technical capabilities that are developed in the project for the AT and AGV specific. The simulations are based on both prototypical open-source software and dedicated commercial software. The results presented in this deliverable are used in WP 4 and 5 and will be used for setting up the real-life demonstrations. The results will be disseminated in the next project workshop meeting in March 2015 and on the project website.

Table of contents

Introduction

The Cargo-ANTs project aims to create smart Automated Guided Vehicles (AGVs) and highly Automated Trucks (ATs) that can cooperate in shared workspaces for efficient and safe freight transportation in main ports and freight terminals.

The objective of this deliverable is to demonstrate the AT and AGV capabilities in simulations. The results include simulations at different levels:

1. Simulate the expected capabilities and operation of the two individual ANTs (AGV and AT) based on the use-cases defined in WP3.
2. Simulate operations of both ANTs combined at a higher level to show how performance and capacities of a terminal can be improved and how the overall automated transhipment system will contribute to CO2 and emission reduction in ports and terminals.
3. Simulate more technical details of the two individual ANTs (AGV and AT) showing the environmental perception, vehicle control and manoeuvring and accident avoidance.

The simulations of level 1 and 3 are based on PreScan [1] commercial software, whereas level 2 simulations are based on prototypical open-source software. The results of the simulations are used in both WP 4 and 5 and will be used for setting up the real-life demonstrations. The results will be disseminated in the next project workshop meeting in March 2015 and on the project website.

Simulation environments

This section describes the different simulation environments that are used for the different levels of simulation.

1.1 PreScan

PreScan is a physics-based simulation platform which is designed for testing Advanced Driver Assistance Systems (ADAS). It is a tool simulating typical automotive sensors such as radar, LiDAR, camera, GPS and supports designing and evaluating V2V and V2I communication applications. Using this simulation environment helps in aligning the use case definitions, defining the expected behaviour of the AGV and AT and development by different partners at different locations across Europe. Besides that, it limits the number of (costly) real world tests required in the development phase. Due to its physics-based nature and focus on ADAS, it will only be used for the level one and level three simulations.

Virtual world model

Within PreScan, a number of virtual worlds is created. Each of these worlds is tailored for one of the use cases defined in earlier deliverables; hence some differences will be present. The virtual world is meant to be representative for a real container terminal and therefore combines object models which are readily available within PreScan with terminal specific models downloaded from the online SketchUp database [2], such as differently sized containers. Figure 0-1 shows a typical world as it is used for the level one and level three simulations performed later.

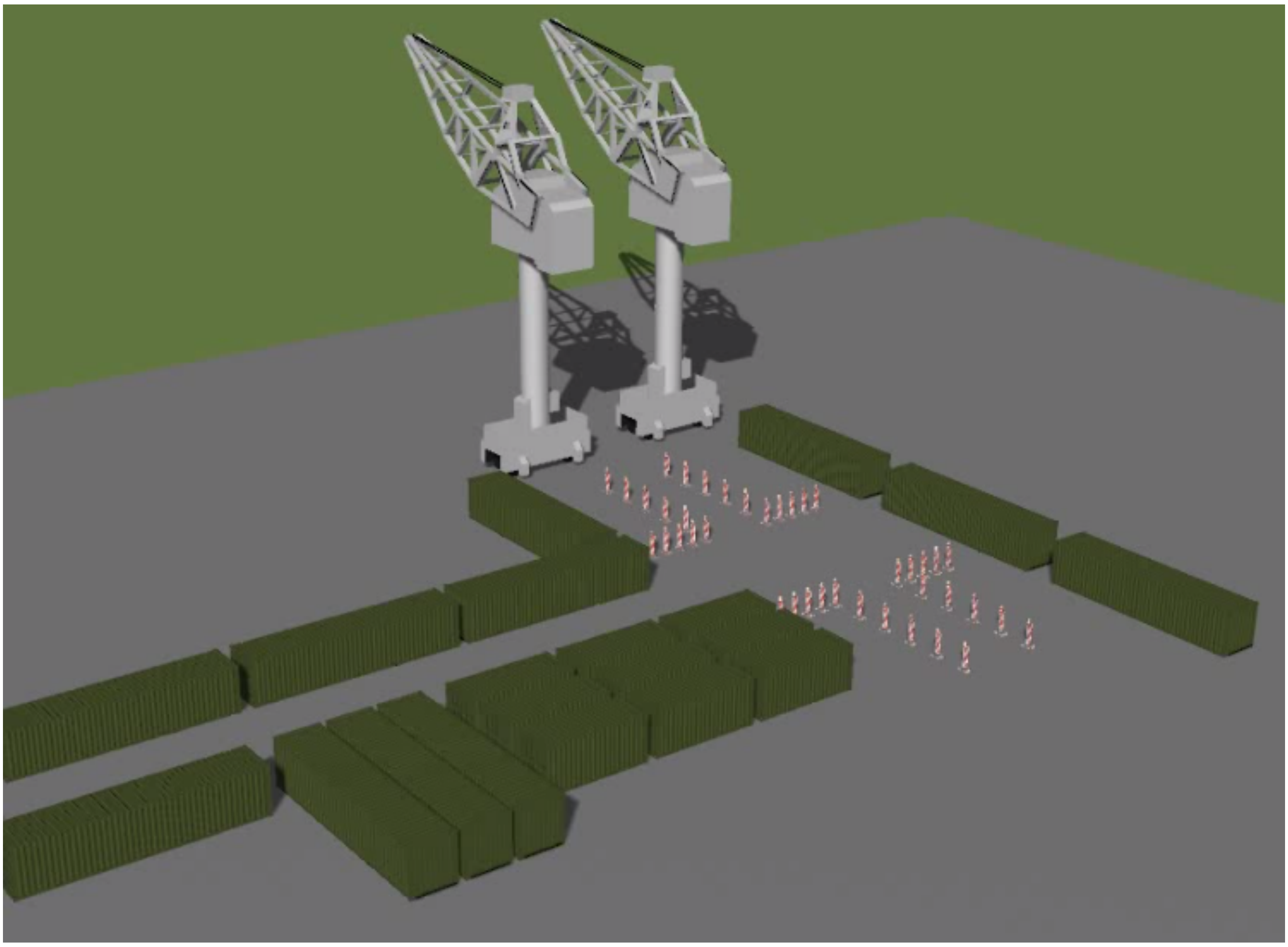

Figure 0-1 A typical PreScan virtual world

AT model

VOLVO

AGV model

The AGV model used is based on a geometrical description from an AGV downloaded from [2]. This model is enriched with the sensors as they will be connected to the actual AGV. The sensors are located according to the current ideas. Future experiments might lead to new insides and new sensors locations. However, updating the locations in PreScan can be done easily as will be shown below.

Figure 0-2 shows how the location of the long range radar, associated with the red coordinate frame, is defined with respect to the AGV's coordinate frame (shown in blue). The red cone represents the field of view of the sensor. In a similar way the two short range radars and the three identical laser range finders are added to the AGV.

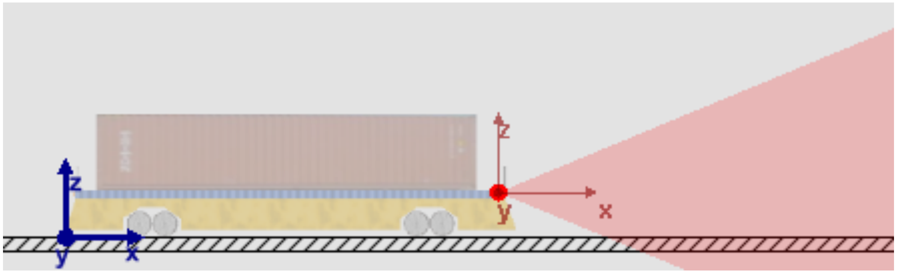

Figure 0-2 The relative position of the long range radar on the AGV

PreScan allows for configuring each sensor individually. The real sensor specifications are known from data sheets delivered by the sensor manufacturers and these specifications are copied to PreScan whenever possible. Figure 0-3 shows how the long range radar is configured.

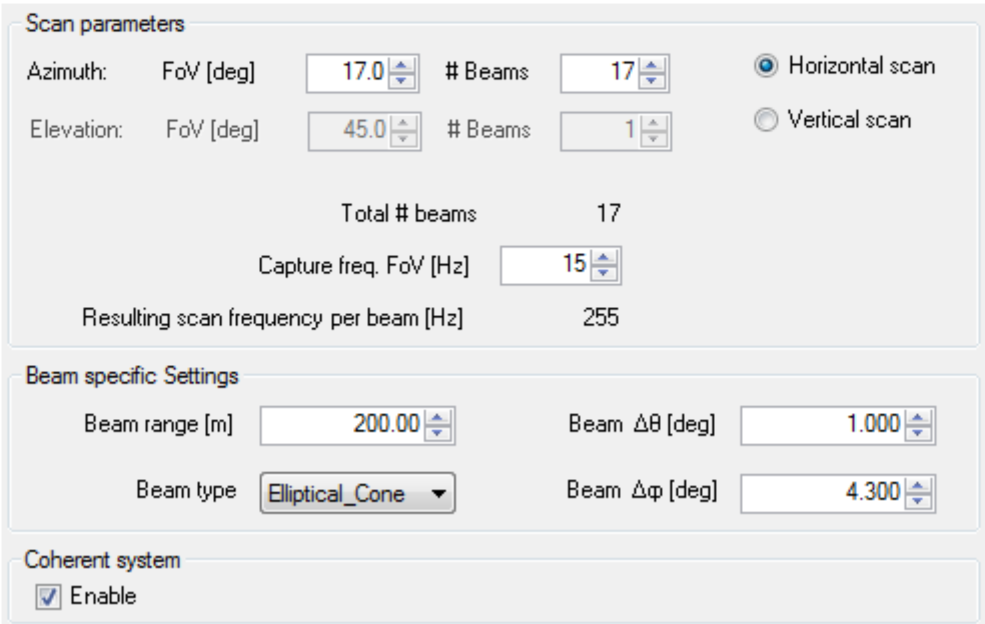

Figure 0-3 Long range radar scan parameters and beam specific settings

As can be understood from Figure 0-3, the field of view is 17 degrees, 8.5 degrees in both the positive and negative direction. The number of beams is 17 since the resolution of the radar is 1 degree. The number of measurement cycles per second is 15. Each beam has a measurement range of 200 m. More details on the scan parameters and the beam specific settings can be read from Figure 0-3 directly. The other sensors are configured according to their own characteristics in a similar manner. Figure 0-4 shows the locations of the three radars and laser range finders: two short range radars (red) each with a relative angle of 45 degrees with respect to the AGV, one long range radar (yellow) facing forward, one front laser range finder (cyan), one side laser range finder (green) and one back laser range finder (purple). The colour to sensor mapping adopted in this figure will be used throughout the whole document.

Figure 0-4 Sensor locations for the AGV (driving direction left to right): two short range radars (red) each with a relative angle of 45 degrees with respect to the AGV, one long range radar (yellow) facing forward, one front laser range finder (cyan), one side laser range finder (green) and one back laser range finder (purple)

The simulated AGV is now fully equipped with the sensors similar to how the sensors will be mounted to the real AGV during the real world experiments later. The sensor 'beams' can be visualized, however, when visualizing all 'beams' from all six sensors the figures look chaotic and are difficult to interpret. For that reason, only a limited number of sensors is visualized in the videos and screen captures which are presented in later sections. It is important to keep in mind that despite the fact that only a limited number of sensors is visualized, *all* sensors are simulated, i.e., only a part of the simulated sensor data is visualized. In order to get an idea of all sensor data, the sensors which are visualized are varied.

The figures below show the radars and the laser range finders added to the AGV. In Figure 0-5, the short range radars are shown in red and the long range radar is shown in yellow. In Figure 0-6 the front laser is shown in blue (cyan), the side laser is coloured green and the back laser is shown in purple.

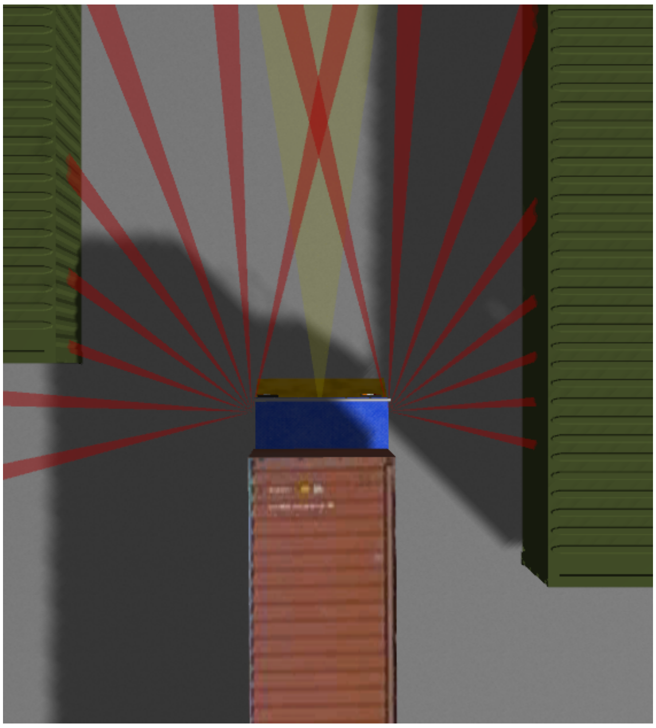

Figure 0-5 The sensor beams of the radars mounted to the AGV

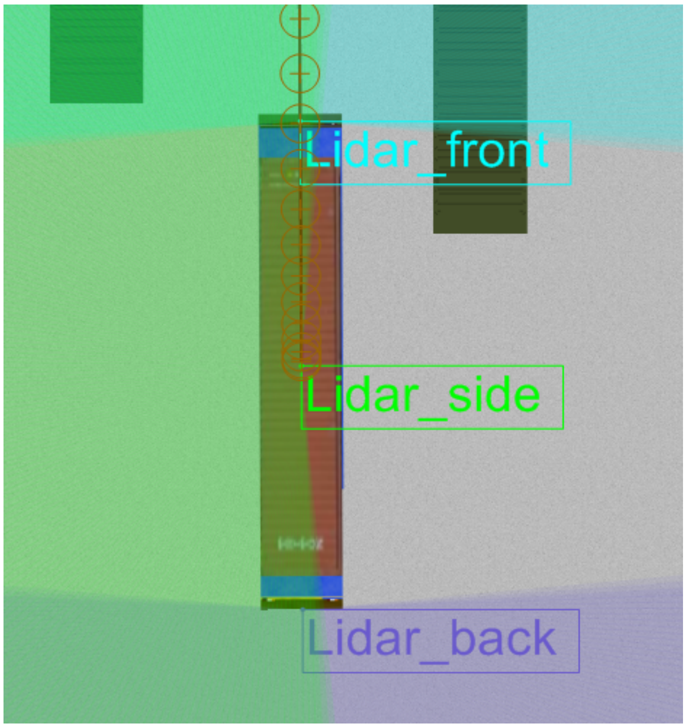

Figure 0-6 The fields of view of the three laser range finders

1.2 Site simulation

HALMSTAD

The site simulator serves for simulating the assignment of ANTs to containers (task assignment) as well as for visualizing the site-working environment, i.e. it provides a top view of containers and ANTs. It also provides a means for stopping the vehicles via the ROS interface (an emergency stop). Since the simulator is web-based, it can be visualized in any web browser, thus making it independent with respect to the employed operating system and platform.

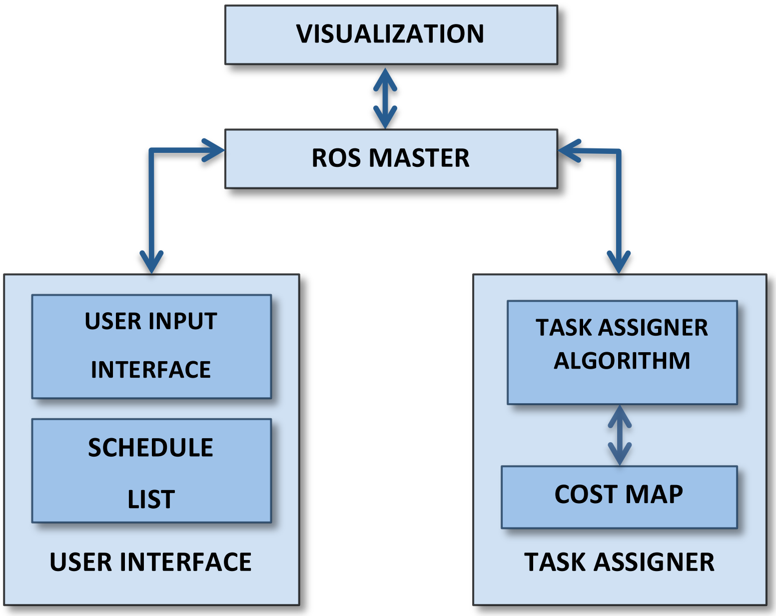

Figure 2-7: System architecture of the site simulator.

Figure 2-7 shows the architecture of the site simulator. Each part of simulator is connected through the ROS master. The ROS Master[1] provides naming and registration services to the rest of the processes (nodes) in the ROS system. It tracks publishers and subscribers to ROS topics as well as ROS services. The role of the ROS Master is to enable individual ROS nodes to locate one another. Once these nodes have located each other they communicate with each other peer-to-peer.

The site simulator is comprised of three main modules, which are linked together through the ROS Master:

- Visualization.
- User interface.
- Task Assigner.

The **Visualization** module provides an online-visualization of the site-working environment in the form of a top view of a map with all containers and ANTs. The **User interface** module is in charge of defining the input to the **Task Assigner**, such an input is defined by two sub-modules: a *user input interface* module that allows to specify the number of ANTs and containers to schedule and the *scheduler list* module, which contains a pre-defined list of available ANTs and containers with priority numbers.

---

[1] http://wiki.ros.org/Master

The **Task Assigner** module takes as inputs the ANTs and containers to be scheduled and gives a valid assignment. To compute such an assignment, it first converts the site map into a cost map using a distance transform and then applies an implementation of a combinatorial optimization algorithm to solve the assignment problem.

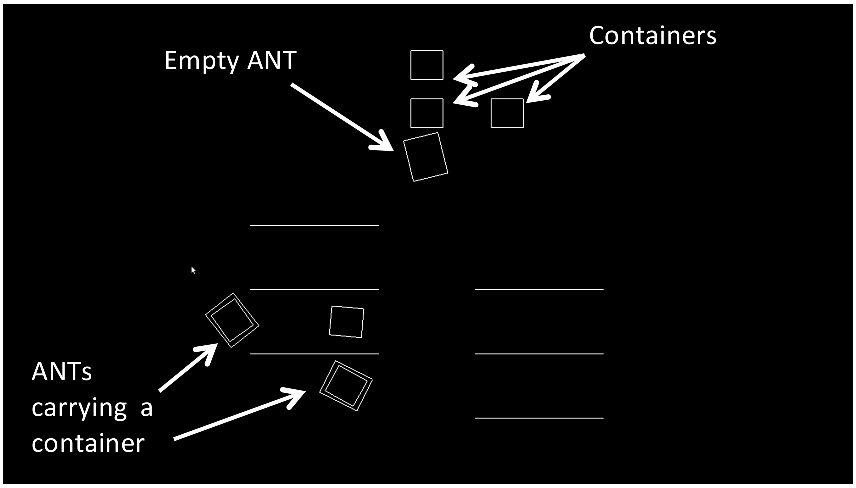

Figure 2.8: Example of site simulation visualization.

Figure 2.8 shows the output from the **Visualization** module of the Site simulation. As described above, it consists of a top view of containers and ANTs. As the goal is to visualize only the scheduling process, the visualization is kept as simple as possible and we represent containers and ANTs with smaller and larger rectangles, respectively. The horizontal lines correspond to simulated obstacles in the environment. In Figure 2.8, three ANTs are shown, two of them are loaded with containers, while the third one is empty and is about to pick up a container.

The simulator also publishes ROS messages with the status of each container. These are custom-made ROS messages employed by the site simulator. Figure 2.9 shows the printouts from these messages for the task assignment depicted in the lower part of the figure. In the printouts it is shown the status of each container, i.e. it shows the identification name of each container, the ANT that is carrying it, the location where it should be picked up and the place where it should be taken. These locations are expressed in site map coordinates.

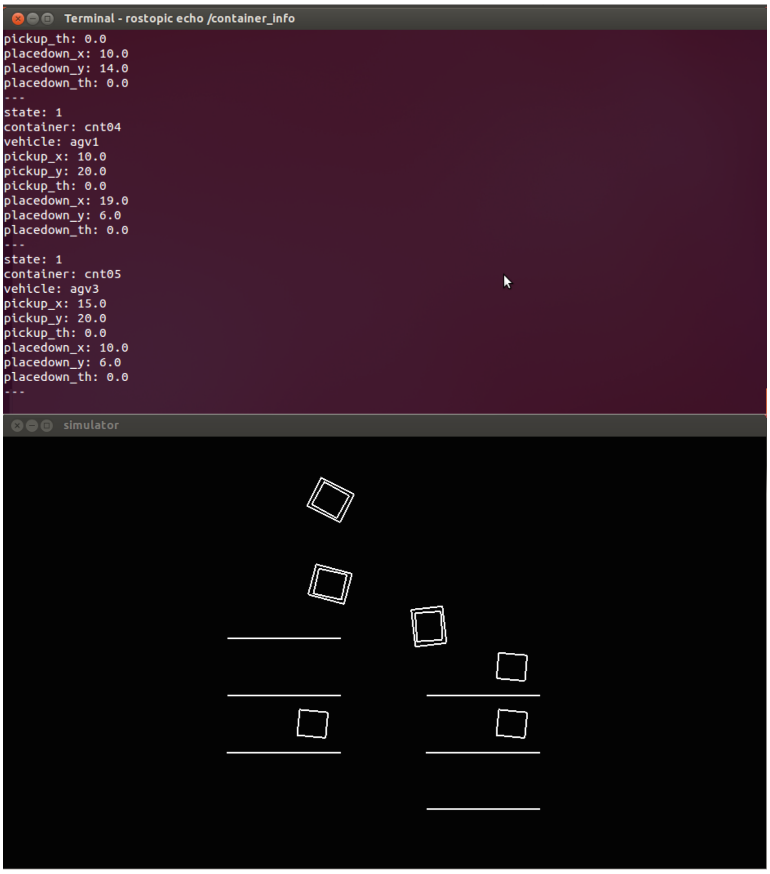

Figure 2.9: Example site simulator and printouts of example ROS messages.

The **User interface** module is shown in Figure 2.10. It is part of the web-based application that allows the user to specify the number of ANTs and containers to be scheduled. In the example shown in this figure, the user likes to schedule the containers 4, 5, and 6, which will be assigned to the currently available vehicles (Vehicles 0, 2, 4, and 6). After the selection of the containers to be scheduled, the user should press the "Task schedule" button to start the task planning procedure. In the example, the goals to each container are predefined and they are displayed in the "Container goals" box after the "Task schedule" button is pressed.

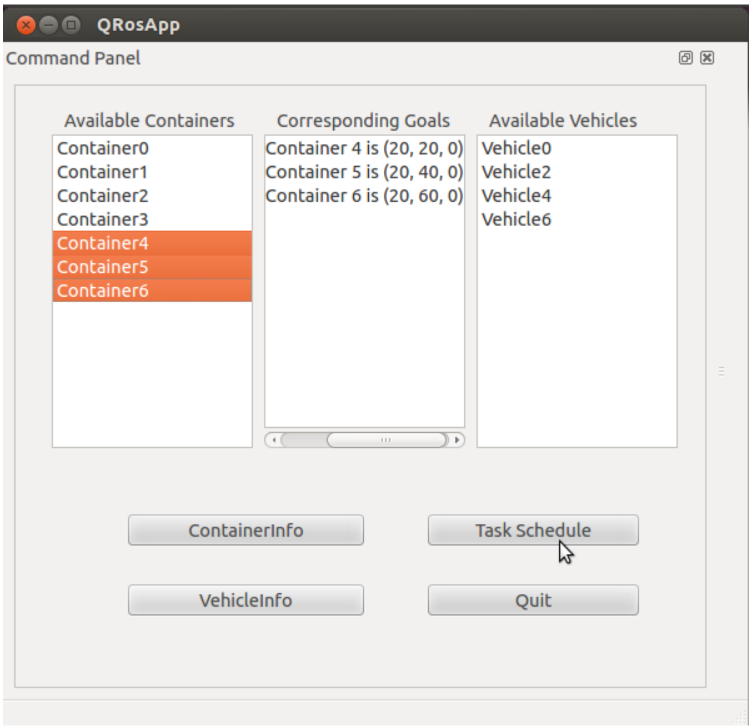

Figure 2.10: Example of the User interface.

Additionally, as the custom-made ROS messages that the Site simulator publishes can only be visualized in machines running ROS, the **User interface** makes also available this information through the web application, i.e. by pressing the "Container Info" button the user is able to visualize the status of each container (its identification name, the ANT that is carrying it, the location where it should be picked up and the place where it should be taken). Similarly, by pressing the "Vehicle Info" button the user will be able to receive the status of each vehicle, i.e. the ANT name, the container name that is transporting, its current location with respect to the site map and the ANT status (i.e. whether it is idle or it is picking or placing a container).

Level 1: Expected capabilities of ANTs in defined use cases

The level one simulations are meant to show the expected capabilities of ANTs in the defined use cases. Whenever a use case is identical for the AGV and the AT, only one of the two is shown, since the other one ANT should behave similar. The sensors are not explicitly included as this is part of the level three simulations. In the sections below screen shots of the use cases as implemented in PreScan are given, the videos showing the complete use case can be found at the project repository in the videos folder which is in the same folder as this document. Note that the videos show a loaded AGV, i.e., an AGV with container whereas the AT is not loaded. At this point, the question whether or not an ANT is loaded is considered to be irrelevant for the level one and level three simulations.

1.3 UC_1.1: Go to way-point

In the first use case a path must be planned and followed to a way-point or final destination. The use case is defined for both the AGV and the AT and no lines or magnetic grids can be used for path planning or path following. Figure 0-1 shows a screen shot of the video displaying the desired behaviour. The path which is followed will be determined by the site.

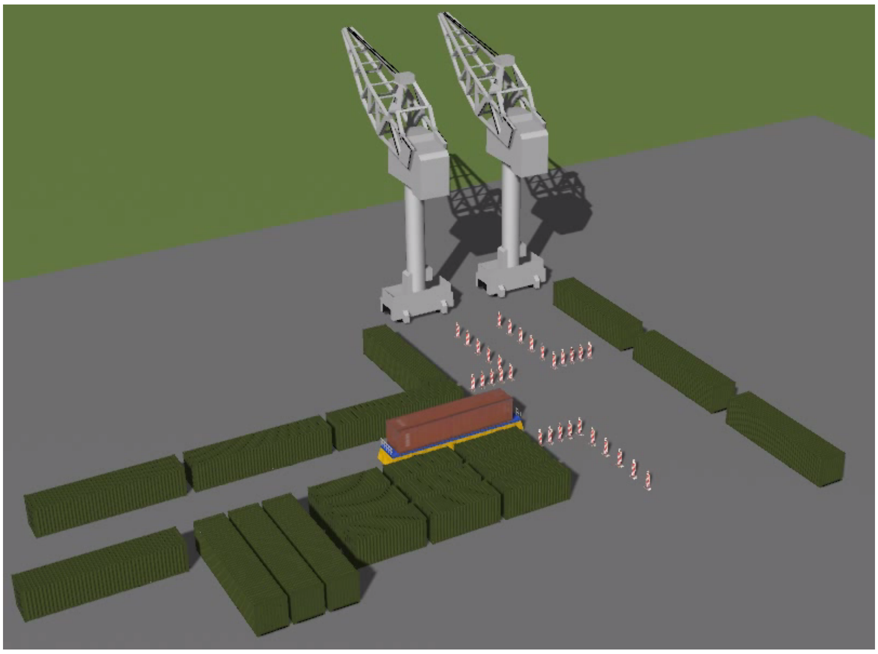

Figure 0-1 Use case 1 for the AGV

1.4 UC_1.2: Backward parking

VOLVO AT only

1.5 UC_2.1: Dynamic path planning: unexpected standing object on path

Use case 2.1 initially is similar to use case 1.1, however, when driving the path as it is planned by the site an unexpected standing object is observed on the path. As a result, the path must be re-planned. The new path should take constraints implied by the environment, e.g., drivable and non-drivable areas, into account. Again, no lines or magnetic grids should be used while driving. During this use case no other dynamic objects are assumed to be present on the terminal area. Figure 0-2 shows how the ANT drives around the static obstacle (in this case a container); the video shows the complete scenario.

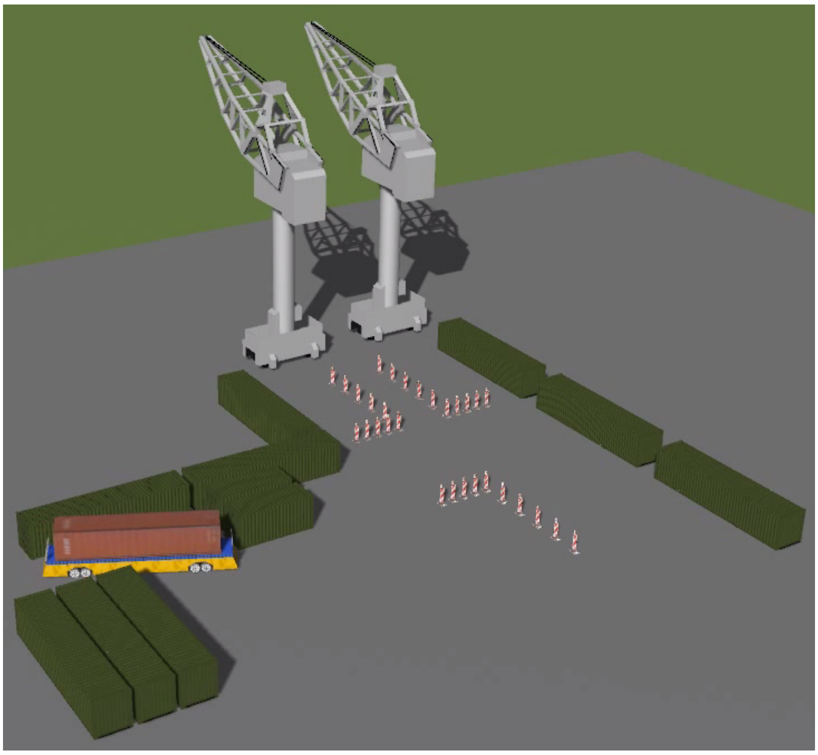

Figure 0-2 The AGV driving around a group of static obstacles (use case 2.1)

1.6 UC_2.2: Collision avoidance for moving object

In use case 2.2 again some complexity is added to the scenario. The starting point is an ANT navigating autonomously to a way point without using lines or magnetic grids for localizing itself. However, this time an unexpected moving object crosses the path of the ANT. The AGV or AT detects the object and the potential collision. Contrary to what is dictated by the planned trajectory, the ANT decides to break to avoid a collision. Once the path is free and the collision is avoided, the ANT resumes with the path as it was planned by the site unit. The waiting AGV is shown in Figure 0-3.

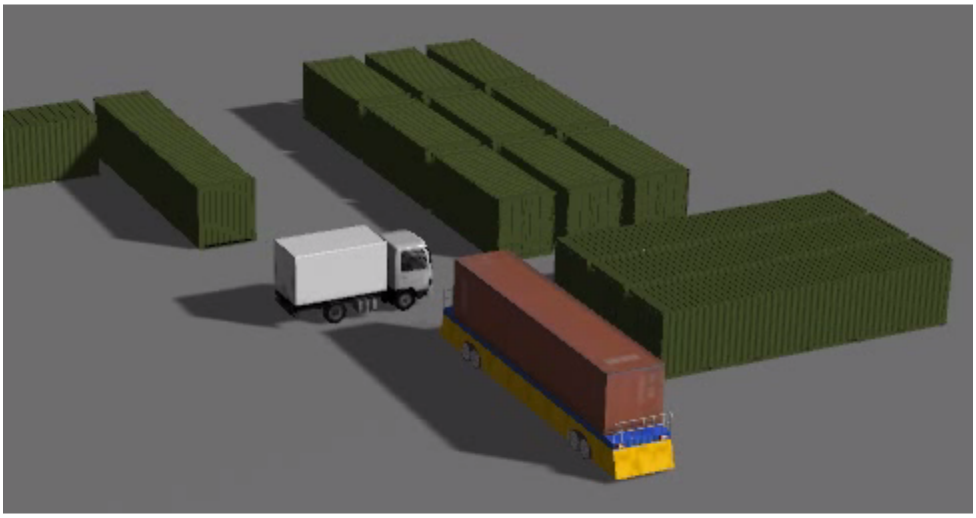

Figure 0-3 The AGV just stopped for an unexpected moing obstacle as specified by use case 2.2

1.7 UC_2.3: No braking in situation which might seem a hazard (false-positive)

In this use case again the ANT is driving a trajectory planned by the site and leading to a given way point. Like in the previous use case, both the ANT and one other dynamic object are present in the closed environment. The ANT and the other dynamic object are driving in opposite directions on the same line and, therefore, are approaching each other. The dynamic object is detected by the ANT. The ANT decides to continue driving its trajectory since this trajectory is curved and bends away from the other dynamic object before a collision can occur. The trajectories are visualized by the coloured lines in Figure 0-4.

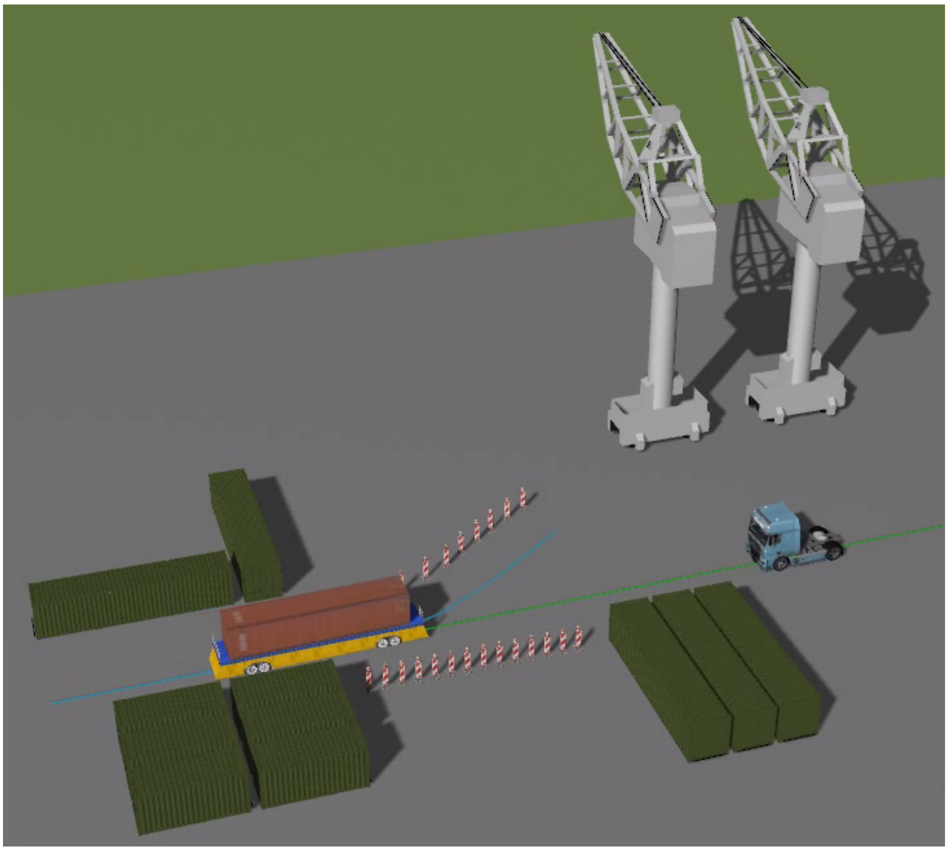

Figure 0-4 The AGV detects the movind obstacle but its path is curved (blue line) hence the ANT decides to continue driving

1.8 UC_3.1: Prioritizing

The prioritizing use case 3.1 involves two ANTs in a closed environment. The trajectories planned from the ANTs partially overlap. Once the ANTs detect each other, they exchange priority information and based on the priority information they decide who should go first. Based on the outcome, the lower priority ANT slows down (or stops) and, that way, makes room for the other ANT to go first. As a result, the higher priority ANT can follow its path without slowing down. Figure 0-5 shows a screen capture of the scenario. Once this higher priority ANT is at a safe distance from the lower priority ANT, the lower priority ANT resumes driving its original path.

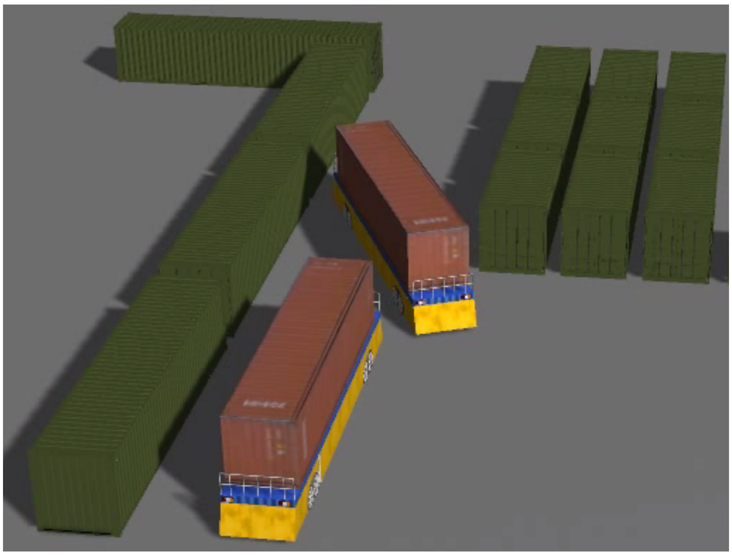

Figure 0-5 One ANT stops to give room to the ANT which has a higher priority as specified by use case 3.1

1.9 UC_3.2: Vehicle following

The last of the use cases is on vehicle following. The AT and the AGV share a path. The AGV follows the AT at a safe but short distance. The AT communicates the relevant information to the AGV. Again, no lines and magnetic grid or other moving objects are present in the closed site environment. Figure 0-6 shows a screen shot of the PreScan experiment of this scenario.

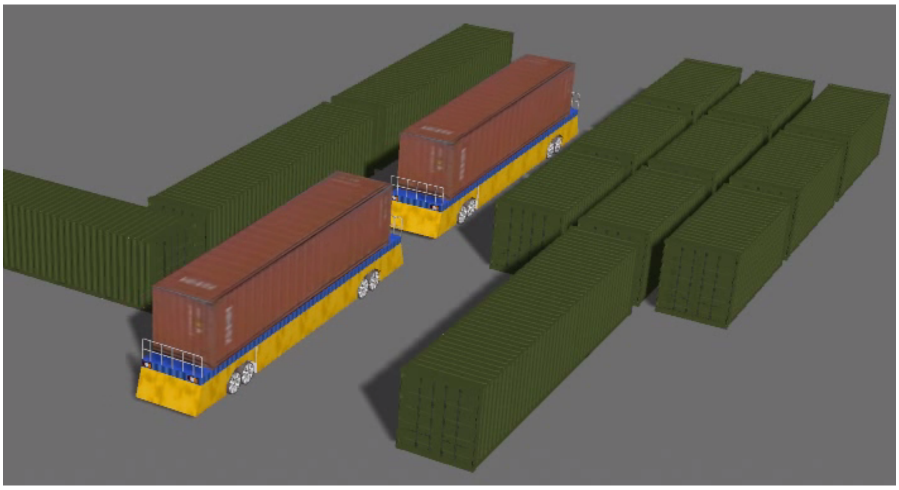

Figure 0-6 Two ANTs in vehicle following mode (use case 3.2)

Level 2: Expected terminal performance

HALMSTAD

The site simulator described in Section 1.2 allows measuring the performance of different task planning approaches. To this end, the **Task Assigner** module (see figure 2.7 from Section 1.2) can be instantiated with different implementations that solve the assignment problem.

In Cargo-ANTS, the terminal performance is measured from the throughput and flow of the containers. It depends on the number of containers currently on dock, the estimated dwell times for each container and the path length for the containers to reach their specific destinations. Hence, the expected terminal performance of our planning approach is to reduce the time taken as well as the path length for each container. This can be seen as a transportation/assignment problem from the operations research area [3]. With a specific schedule of the list of containers, the site simulator runs an assignment algorithm that provides the ANTs and ATs with their destination and routes.

The expected terminal performance for the task planning in Cargo-ANTs can be expressed as achieving the assignment that satisfies the following constrained optimization problem:

$$\begin{aligned} \text{minimize } & f = \sum_{i,j} c_{ij} x_{ij}, \\ \text{subject to } & \sum_{j} x_{ij} = 1, \quad \forall i, \\ & \sum_{i} x_{ij} = 1, \quad \forall j, \\ & x_{ij} \geq 0, \quad \forall (i,j), \end{aligned} \tag{1}$$

The variable x_{ij} represents assignment of container *i* to vehicle *j* subject to the constraints that only one vehicle is assigned a container and vice versa. The variable c_{ij} represents the cost of the assignment of container *i* to vehicle *j.*

In the ongoing implementations developed as part of Task 5.1, the cost c_{ij} is currently defined by the total length of the path the ANT should traverse to pick up and place the container from one place to another. It thus depends on the path planner approach. In our current experiments we employ the E-star [4] method to solve the path planning part. Unlike A-star, which constrains movements to graph edges, E-star produces smooth trajectories by interpolating between edges. In addition, it also supports dynamic re-planning after local path cost changes. The above corresponds to the work that is currently being done as part of Tasks 5.1 and 5.2.

As part of the ongoing work in Task 5.1, we have implemented and compared two approaches to solve the task assignment: a feedback controller [5] and the Hungarian (Kuhn-Munkres) Algorithm [6]. In the first approach the containers are allotted on a first-come first-serve basis, while the second approach consists of a combinatorial optimization algorithm that solves assignment problems in polynomial time. In our implementation both algorithms employ E-star to compute the paths to pick up and place the containers from one place to another.

Next, we present the simulations results obtained by comparing the Hungarian Algorithm and the feedback controller. For this comparison we simulated six scenarios by combining different number of combinations of available ANTs and containers to schedule. For our simulations we employed the same site map, the identical predefined goals for each container and the same initial location for ANTs and containers. Table 1 summarizes the simulations results obtained from the site simulator. The first column in Table 1 indicates each simulated scenario, i.e. the six different combinations of available ANTs and containers to schedule. The second and third columns show the resulting assignment given by the Feedback Controller and Hungarian Algorithm, respectively. The arrows indicate that the ith-container (**C**$_i$) is assigned to the jth-ANT (**A**$_j$). The last columns show the total path length of the assignments obtained by the Feedback Controller and Hungarian Algorithm, respectively.

Number of ANTs and Number of Containers (CNTs)	Feedback Controller Assignment	Hungarian Algorithm Assignment	Total Path Length (meters)	
			Feedback controller	Hungarian Algorithm
1 ANT – 4 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_1$ $C_3 \rightarrow A_1$ $C_4 \rightarrow A_1$	$C_4 \rightarrow A_1$ $C_3 \rightarrow A_1$ $C_2 \rightarrow A_1$ $C_1 \rightarrow A_1$	24.08	24.08
2 ANTs – 4 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_3 \rightarrow A_1$ $C_4 \rightarrow A_2$	$C_4 \rightarrow A_1$ $C_3 \rightarrow A_2$ $C_2 \rightarrow A_1$ $C_1 \rightarrow A_2$	85.12	45.17
2 ANTs – 8 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_3 \rightarrow A_1$ $C_4 \rightarrow A_2$ $C_5 \rightarrow A_1$ $C_6 \rightarrow A_2$ $C_7 \rightarrow A_1$ $C_8 \rightarrow A_2$	$C_8 \rightarrow A_1$ $C_7 \rightarrow A_2$ $C_6 \rightarrow A_1$ $C_5 \rightarrow A_2$ $C_4 \rightarrow A_1$ $C_3 \rightarrow A_2$ $C_2 \rightarrow A_1$ $C_1 \rightarrow A_2$	112.2	51.73
3 ANTs – 6 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_3 \rightarrow A_3$ $C_4 \rightarrow A_1$ $C_5 \rightarrow A_2$ $C_6 \rightarrow A_3$	$C_6 \rightarrow A_1$ $C_5 \rightarrow A_2$ $C_4 \rightarrow A_3$ $C_3 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_1 \rightarrow A_3$	182.36	72.33
3 ANTs – 3 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_3 \rightarrow A_3$	$C_3 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_1 \rightarrow A_3$	142.42	64.24
4 ANTs – 8 CNTs	$C_1 \rightarrow A_1$ $C_2 \rightarrow A_2$ $C_3 \rightarrow A_3$ $C_4 \rightarrow A_4$ $C_5 \rightarrow A_1$ $C_6 \rightarrow A_2$ $C_7 \rightarrow A_3$ $C_8 \rightarrow A_4$	$C_8 \rightarrow A_1$ $C_7 \rightarrow A_2$ $C_6 \rightarrow A_3$ $C_5 \rightarrow A_4$ $C_4 \rightarrow A_1$ $C_3 \rightarrow A_2$ $C_2 \rightarrow A_3$ $C_1 \rightarrow A_4$	202.11	92.55

Table 1: Simulation results of comparing a feedback controller [5] and the Hungarian (Kuhn-Munkres) Algorithm [6].

From the simulation results shown in Table 1, we can note that since in the first case there is only one ANT that does all the work, the total path cost remained the same for both algorithms. However, in the rest of the scenarios, where there are more than one ANT the "Hungarian Algorithm" provided the overall lowest cost for all of the containers that had to be scheduled. On the other hand, with the feedback controller we obtained the largest cost for such scenarios. This is due to fact that in the feedback controller the path length is not considered to assign containers to ANTs, it does it on a first-come first-serve basis. On the other hand, the "Hungarian Algorithm" explicitly considers the path length to decide on the assignments of containers to ANTs as this method provides a solution to the optimization problem defined by Equation 1, i.e. it always yields optimal assignments.

Level 3: Expected capabilities of ANTs in defined use cases

The simulations at this level are based on the use case simulations presented in Section 0, however, this time the sensors are added. The level three simulations again are developed in PreScan. The experiments do not include the complete sensor data fusion and vehicle control algorithms at this point since these still are in the development phase. As a result, the simulated ANTs are not exact replicas of the AGV and AT, however, the simulated ANTs are as similar as possible. Obviously, the simulation platform can be updated as each of the functions gets matured.

It is good to realize that *all* sensors are simulated, however, the screen captures and videos show only a fraction of the sensor data to keep the presentation understandable. The fraction of the simulated data which is presented is varied in order to give the reader a complete understanding of typical sensor data. Again, the videos showing the complete scenarios can found in the videos folder.

1.10 UC_1.1: Go to way-point

Figure 0-1 below shows the AGV driving towards a predefined waypoint. The red cones represent measurements performed by the two short range radars, the yellow cones represent the front radar measurement. Due to the high resolution of the long range radar, the yellow field of view looks like a single cone. It does, however, consist out of a set of beams, just like the short range radar. Both radars have different ranges. Remember that the laser range finders are simulated but not visualized in order to keep the videos and screen shots understandable.

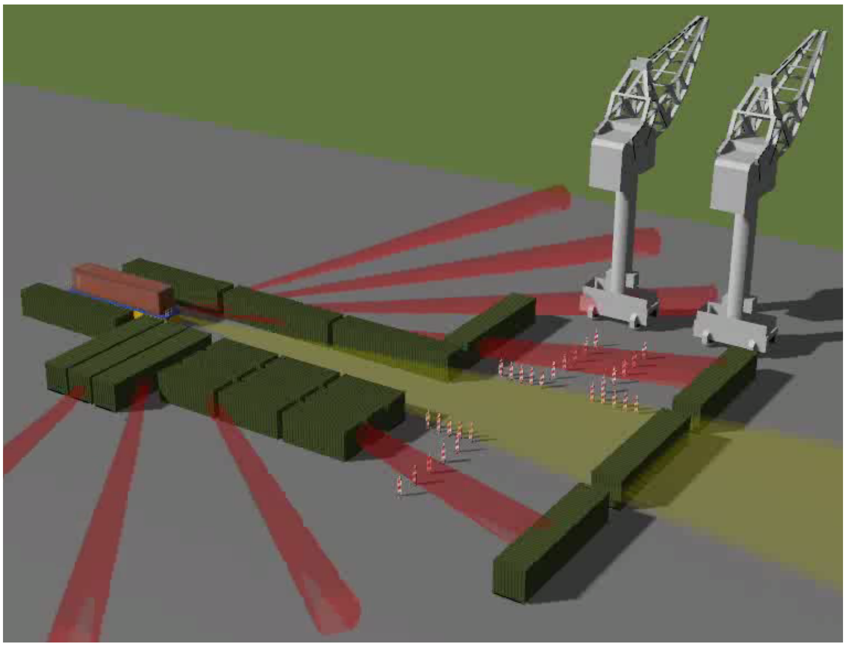

Figure 0-1 Use case 1.1

Additions VOLVO

1.11 UC_1.2: Backward parking

Additions VOLVO

1.12 UC_2.1: Dynamic path planning: unexpected standing object on path

Figure 0-2 shows the dynamic path planning around an unexpected static object. The three laser range finders are visualized, each in a different colour. The fields of view partially overlap. The maximum ranges at which objects are detected are sufficient to detect the static object before a collision occurs.

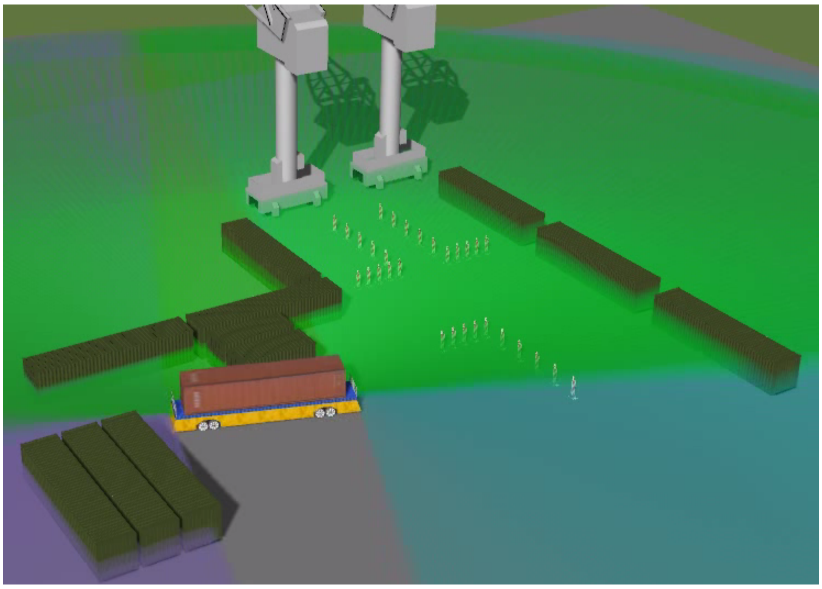

Figure 0-2 Use case 2.1 with simulated laser range finder data

Additions VOLVO

1.13 UC_2.2: Collision avoidance for moving object

In this use case the AGV detects a moving obstacle and stops in order to avoid a collision. In Figure 0-3, the simulated data from the side laser range finder and the two short range radars are shown. The figure shows the point in time at which the moving obstacle just has been detected. At this point in time, the time to collision is sufficiently large to allow for a smooth deceleration in order to avoid a collision, as can be seen in the video which is available for download on the project's repository.

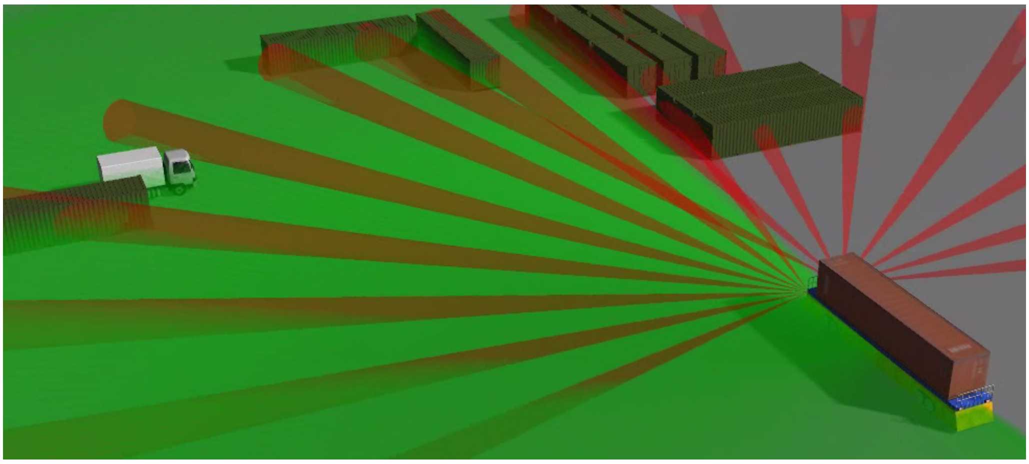

Figure 0-3 Use case 2.2

Additions VOLVO

1.14 UC_2.3: No braking in situation which might seem a hazard (false-positive)

In this use case the laser range finder data of the sensors on the front and back of the AGV is visualized. As shown in Figure 0-4, the moving obstacle is well within the (blue) field of view of the front laser range finder. Since the AGV's planned path is known to the AGV (and visualized by the blue line), it can safely decide to continue driving its path without the risk of a collision with the approaching dynamic obstacle.

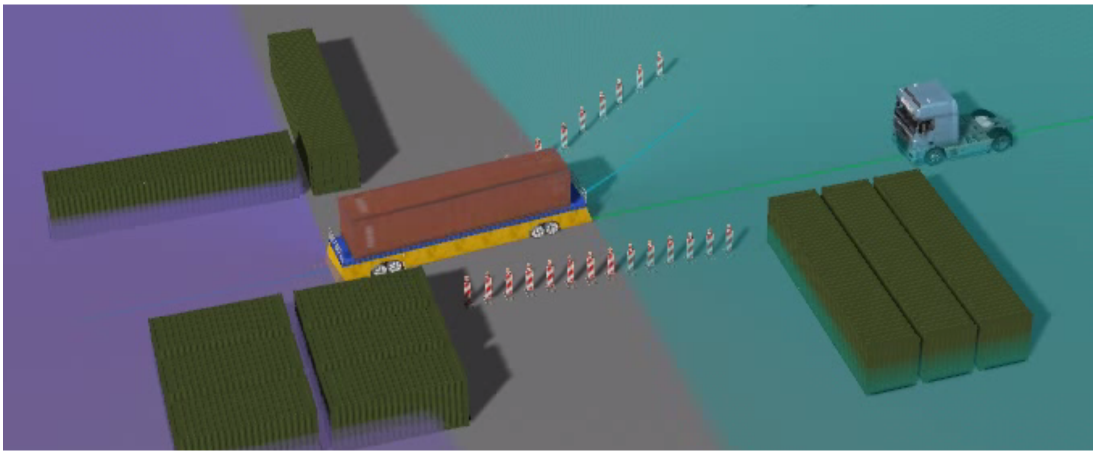

Figure 0-4 Use case 2.3 with laser range data visualized

Additions VOLVO

1.15 UC_3.1: Prioritizing

In this use case two ANTs are sharing a path. The radar 'beams' are visualized for both ANTs, the laser range finders are not. The AGVs in Figure 0-5 negotiated and based on priorities decided that the ANT coming from the side should go first. As a result, the other AGV waits until the AGV coming from the right has passed.

Figure 0-5 Use case 3.1 with two AGVs

Additions VOLVO

1.16 UC_3.2: Vehicle following

Figure 0-6 shows two ANTs, AGVs in this case, driving in vehicle following mode. The visualization shows the radar data of the first AGV and the laser range data from the following ANT. The following distance is small and the velocity profiles of the ANTs are as similar as possible.

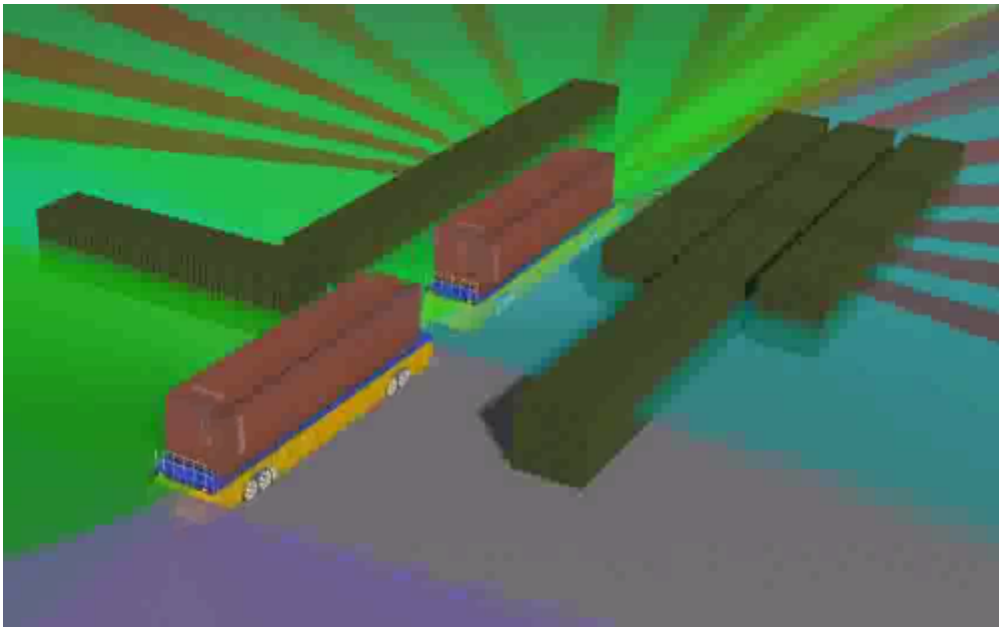

Figure 0-6 Use case 3.2: vehicle followin

Additions VOLVO

Conclusions

This deliverable presented the results of a series of simulations of use cases at different levels. The level one simulations showed the expeced behavior of the ANTs in the use cases specified in WP 3. By simulating all use cases, the expectations are aligned.

As part of the level two simulations, we described a site simulator developed using open-source software to test the task planning process in Cargo-ANTs. It allows simulating the operations of ANTs at a higher level and measuring the performance of each task assignment. In addition, we described the expected terminal performance of the task planning approaches tested on the site simulator. Moreover, to exemplify the utility of the site simulator we also included simulation results to compare two task planning approaches that are being tested as part of the work that is currently being done in Tasks 5.1 and 5.2.

The level three simulations have shown the same use cases as simulated in the first level, however, this time with sensor data. The effect of different sensor locations can easily be investigated and, therefore, the level three simulations helped in fixing the sensor locations on both the AT and the AGV for the first series of real world experiments.

REFERENCES

[1] https://www.tassinternational.com/prescan

[2] https://3dwarehouse.sketchup.com/

[3] Lantao Liu, Dylan A. Shell, An anytime assignment algorithm: From local task swapping to global optimality. Autonomous Robots 35: 271-286, 2013.

[4] Roland Philippsen, Roland Siegwart. Smooth and efficient obstacle avoidance for a tour guide robot. In Proc. IEEE Int. Conf. Robotics and Automation, 2003.

[5] W.C. Ng, K.L. Mak. “Yard crane scheduling in port container terminals”. Applied Mathematical Modelling, Vol.29, Issue 3, Pages 263–276, 2005.

[6] Lantao Liu and Dylan A. Shell. “Assessing optimal assignment under uncertainty: An interval based algorithm”. International Journal of Robotics Research, 30(7) 936-953, 2011.