Chapter

# Controlling a Fleet of Autonomous LHD Vehicles in Mining Operation

*Alexander Ferrein, Gjorgji Nikolovski, Nicolas Limpert, Michael Reke, Stefan Schiffer and Ingrid Scholl*

## Abstract

In this chapter, we report on our activities to create and maintain a fleet of autonomous load haul dump (LHD) vehicles for mining operations. The ever increasing demand for sustainable solutions and economic pressure causes innovation in the mining industry just like in any other branch. In this chapter, we present our approach to create a fleet of autonomous special purpose vehicles and to control these vehicles in mining operations. After an initial exploration of the site we deploy the fleet. Every vehicle is running an instance of our ROS 2-based architecture. The fleet is then controlled with a dedicated planning module. We also use continuous environment monitoring to implement a life-long mapping approach. In our experiments, we show that a combination of synthetic, augmented and real training data improves our classifier based on the deep learning network Yolo v5 to detect our vehicles, persons and navigation beacons. The classifier was successfully installed on the NVidia AGX-Drive platform, so that the abovementioned objects can be recognised during the dumper drive. The 3D poses of the detected beacons are assigned to lanelets and transferred to an existing map.

**Keywords:** autonomous vehicles, fleet control, planning, mining, computer vision, machine learning

## 1. Introduction

Along the lines of digitalisation and transforming industries towards Industry 4.0, also the mining industry as being a rather conservative industry is moving into this direction. For instance, in [1] the authors envision Mining 4.0 as a future concept of mining operations. The labour of a future mine worker will be smarter, more collaborative, more connected including augmented/virtual reality (AR/VR) technologies. In that sense, technology will also transform this sector into the Industry 4.0 direction. This has also an impact on the degree of automation deployed in future mines. In [2], mining is connected with key technologies such as intelligent systems, machine learning, and AR/VR. Additionally, there is a trend of the European raw materials industry towards changing from open pit mining to underground mining to reduce the environmental footprint of the mine. Digitalisation and automation are key technologies for further transforming mining operations into a decarbonised and more sustainable operations (see, e.g. [3–5]). This leads to hybrid mines, where parts of the mine are still open pit and parts are underground mines. This poses, in particular, additional challenges for the automation process. Whilst in the open pit part of the mine, methods from autonomous driving using GPS etc. can be deployed, in the underground part, methods from mobile robotics need to be used in an ever-changing environment with the need to continuously monitor and track the changes in the robots' maps.

Under certain limitations such as loaders following a prerecorded path, automated LHD vehicles are already commercially available [6]. However, the problem of fully automated guided load-haul dumpers remains a challenging problem. This results in a large body of related works that focus on the core problems like navigating and localising LHD vehicles. Some recent works are, for instance, [7], [8], or [9]. The former proposes topological navigation for underground haulage vehicles. The basic idea is that in underground mines with many tunnels, a highly precise localisation pose in the tunnel is not of importance, the LHD vehicle could localise at crossings or other important waypoints of a topological map. In our work, we also follow the idea of using topological maps, specifically in the Lanelet2 format [10], which is a common map format in autonomous driving.

In [8], a robust localisation system integrating cameras, LiDARs, and odometry information for underground LHD vehicles is proposed. The system is tested on mining datasets and shows good accuracy with mean errors below 1 meter. The methods deployed in this work are very similar to our approach. An interesting addition to common sensor cues is proposed in [9], where IMU data are used to record ground ripples as an additional localisation information. For measuring similarities and dissimilarities in the recorded data in order to recognise ground patterns, the dynamic time warping algorithm is deployed. The basic idea behind this approach is not to rely on visual landmarks such as visual tags to keep the extra infrastructure required in the mine as little as possible. On the other hand, extending the infrastructure in mines such as installing Wi-Fi at least in parts of the mine is no longer out of question. Some approaches make use of vehicle-to-vehicle (V2V) communication for establishing communication networks in underground mines [11]. The purpose of this work is to localise other vehicles underground deploying V2V communication. But localising other vehicles underground is not the only interest for mining operations as tracking mine workers underground is mandatory in many countries. Seguel et al. [12] overviews relevant positioning technologies to track mine workers underground. Finally, fleet control is one of the tasks that needs to be solved for automated haulage and artificial intelligence (AI) approaches are being deployed. Bnouachir et al. [13] overviews some approaches to intelligent fleet management in mining operations, whilst [14] proposes a real-time scheduling algorithm based on flow-achieving scheduling trees to overcome shortcoming of off-the-shelf software which often is based on myopic heuristics. In our case of fleet management, we make use of a planning approach which is based on hierarchical task networks [15].

Many related works concentrate on particular open problems in automating mining operations focusing mainly on localisation and navigation challenges or fleet-level planning focusing either on underground or open pit mines. Hybrid mines, which combine underground and open pit mine operations, pose a particular challenge for autonomous vehicles.

In this chapter, we report on the current state of affairs in our endeavour to automate hauling operations in hybrid mines. This work is based in part on our previous works [16–22]. In particular, we report on (1) the hardware setup of our fleet of robot vehicles including the LHD vehicles and a tracked exploration robot, (2) the overall ROS 2-based system architecture, (3) a model-based predictive control approach for controlling the articulated LHD vehicles, (4) an HTN-based approach to the tour planning of the fleet of vehicles.

The rest of this chapter is organised as follows. We present the hardware and sensor setup of the articulated LHD vehicles as well as of our exploration vehicle in Section 2 before we show the overall ROS 2-based software architecture (Section 3). Section 4 addresses the low-level navigation approach for controlling the articulated LHD vehicles. In Section 5, we outline the fleet and mission control system. Then, Section 6 reports on our approach to classifying drive ways in the mine and to mapping the changing mining environment. In Section 7, we show some experimental results. Then we conclude.

## 2. Hardware setup

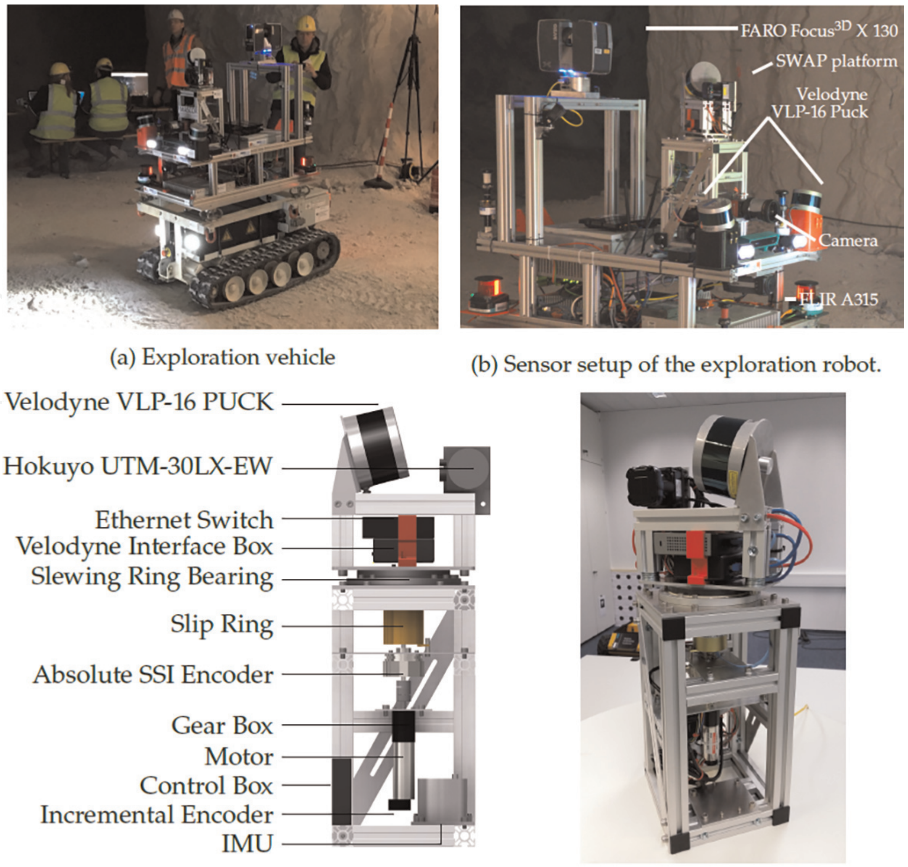
In our mining automation projects, we are using two different platforms. One platform has been deployed for exploring and mapping the mine environment (Section 2.1), the other one is smaller-scale articulated dumping vehicle from Wacker Neuson that we turned into autonomous vehicle (Section 2.2).

### 2.1 Exploration vehicle

As mining environments are constantly changing, also the maps required for autonomous LHD vehicles need to be updated constantly. In our previous work [19, 23], we developed an exploration vehicle for mapping underground mines with dense 3D point clouds. As we can only give a brief overview of this work here, we refer to our previous work for further details.

The exploration vehicle is shown in **Figure 1a** and **b**. It is a skid-steered tracked robot based on a platform similar to a mini excavator but using a suspended undercarriage. It is equipped with a number of different sensors used for navigation, localisation, and mapping. The robot reaches speeds of up to $3\ \mathrm{m\ s^{-1}}$ and is controlled via the ROS [24] Movebase. For navigation, collision avoidance, and terrain classification, two Velodyne VLP-16 Puck LiDARS are mounted at the front. For mapping, we equipped the exploration vehicle with a custom-built rotating sensor platform shown in **Figure 1c** and **d**. It allows to acquire a (nearly) complete sphere around the vehicle. A Velodyne VLP-16 PUCK LiDAR with 16 scan lines, a vertical opening angle of 30°, and a horizontal range of 360° is mounted with a 14° inclination to the vertical axis. For short-range measurements, we additionally equipped the device with a 2D Hokuyo UTM-30LX-EW range scanner mounted in a 90° angle to the rotation plane. For teleoperation during mapping missions, an Allied Vision GT6600C high-resolution camera with a wide-angle lens is mounted at the front of the robot. As a safety feature, we mounted a FLIR A315 thermal camera at the front of the robot in order to be able to detect persons even when not sufficient light is available. Additionally, an IMU for providing the orientation of the platform with respect to the ground is mounted on the vehicle.

For registering the 3D point clouds from the scanning device, we deploy the Iterative Closest Point (ICP) method [25] based on the Point Cloud Library implementation [26]. To minimise the errors of pairwise registration of many point clouds, all point clouds are registered globally making use of the GraphSLAM [27] algorithm



(a) Exploration vehicle

(b) Sensor setup of the exploration robot.

(c) Components of the platform.

(d) Photo of the platform.

**Figure 1.**
*Exploration robot developed for mapping underground mining sites (from [23]).*

after Lu and Milios [28]. It creates a graph of the connections between all point clouds with overlaps and minimises the alignment errors of all connections simultaneously. Finally, for an internal representation of the 3D map, Octomap [29] is used. This offers the advantage of being able to query the information in various resolutions and to map the distinctions that are important for navigation between free, occupied, and unknown cells.

We present an example map from an underground mine in Krölpa, Germany, in Section 7.

## 2.2 LHD vehicles

The base vehicle we use is an articulated haul-dumper from Wacker Neuson (Model 1501).[1] Our fleet consists of three of those vehicles. **Figure 2** shows an exemplary prototype. This chapter is based on a paper that has been previously published. We refer to [18] for further details.

The dumper is an off-the-shelf model that can handle loads of up to 1500 kg. The control of the brakes and the angle of the articulated joint of the vehicle is realised by a hydraulic system, which depends on the hydraulic pressure generated by a diesel engine. In order to automate the vehicle, we installed electric linear actuators with our

[1] https://www.wackerneuson.de/produkte/dumper/raddumper/raddumper-1501/

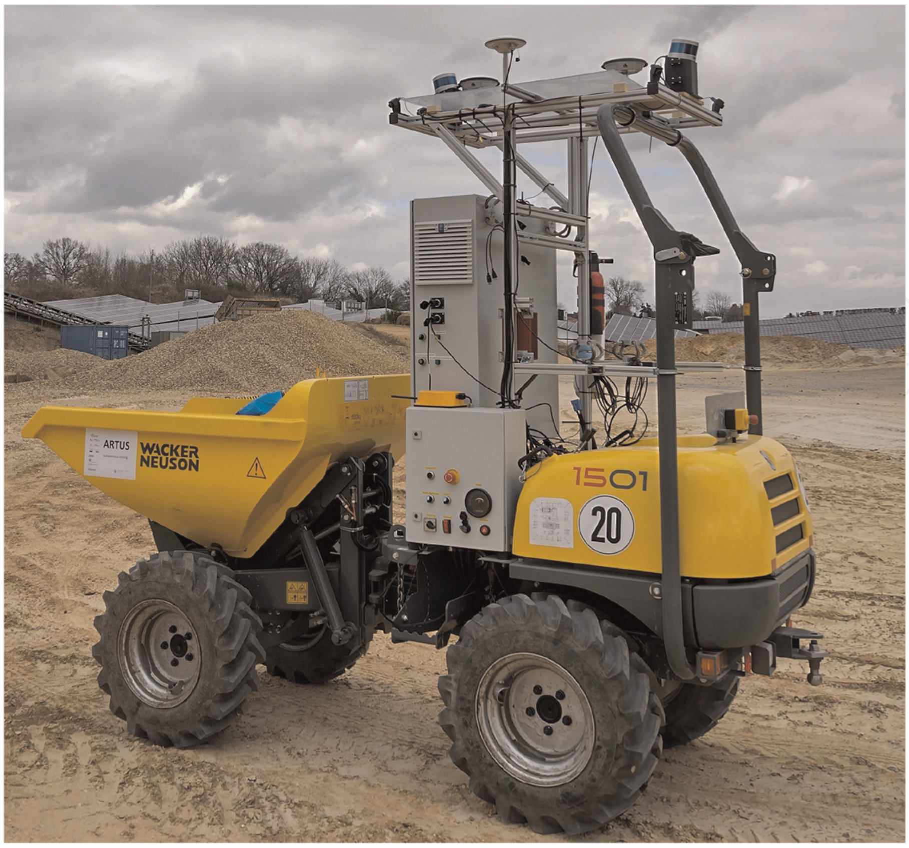


**Figure 2.**
*Picture of one of the articulated haul-dumpers at the test site.*

project partner Fritz Rensmann GmbH & Co. KG to control the brake and the throttle. Additionally, we attached a rotational servo motor to the steering axis and replaced the manual valves by electromagnetic valves to control the skip. We also installed hall sensors on each wheel to obtain information on the vehicle's state.

The control system comprises the different components shown in **Figure 3**. These parts are presented in the following. The software architecture of the vehicle will be discussed in the next section. As can be seen in the figure, each dumper can be controlled via our high-level control stack, where the tasks for the individual vehicles are generated by the overall mission planner for the whole mine to fulfil the daily production goals. The high-level control system is described in Section 5. Additionally, each vehicle can be controlled via remote control. In particular, this comes in handy when the vehicle needs to be steered onto a low loader trailer for transport.

*Real-time Controller*. The low-level real-time controller runs on a programmable logic controller by Beckhoff. To communicate with the motors and the PLC, we make use of a CAN interface. The PLC serves as a message filter, that prevents unsafe commands to be propagated to the motors, and as a kill-switch manager for controlling a number of kill-switches, which are installed on the vehicle including a radio kill-switch. The PLC runs a PID controller regulating the angle of the articulated joint and the engine speed by properly actuating the motor moving the steering axis and the motor opening and closing the throttle valve. For more detail, we refer to [30].

*Compute Nodes*. As described below (Section 4), we implemented a GPU-based model-predictive path-follower, a high-level control system, and various computer vision algorithms. All the software modules run on two dedicated compute devices.

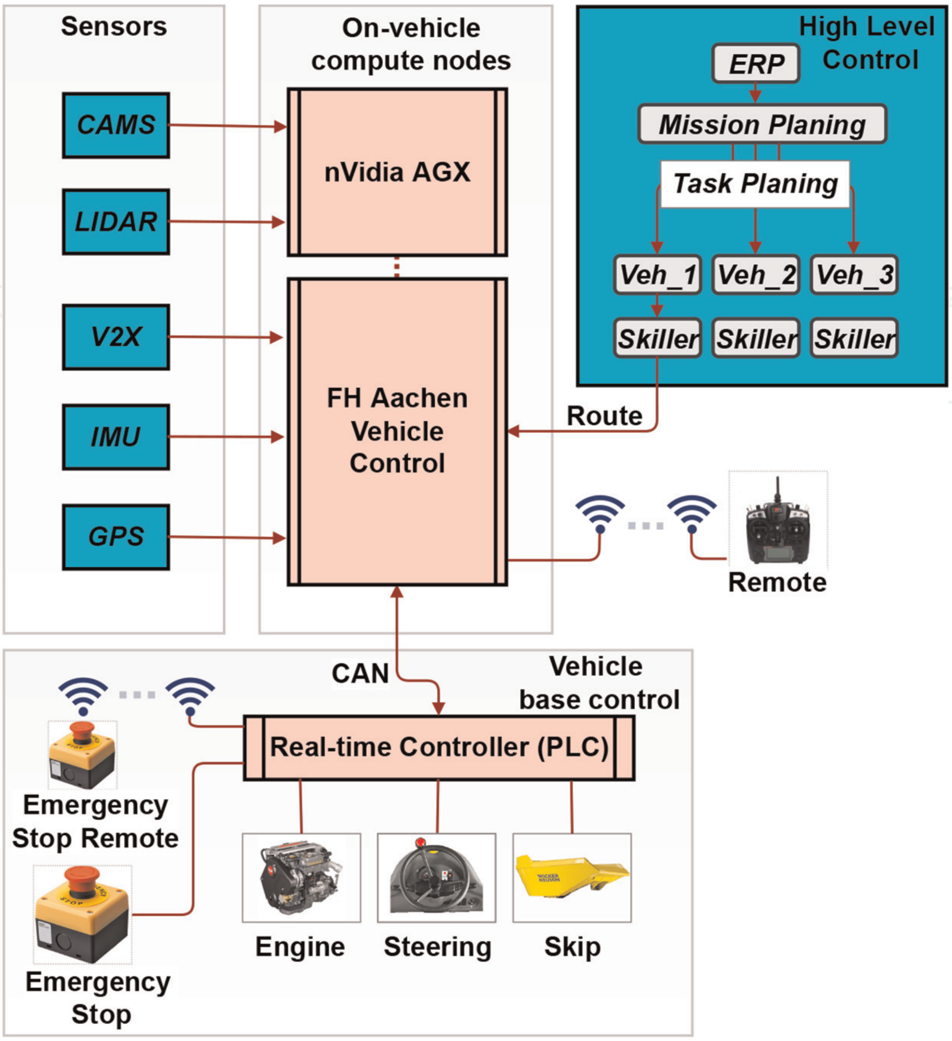


**Figure 3.**
*System architecture of haul-dumper control system (From: [18]).*

We make use of a Zotac ZBOX Magnus One including an NVIDIA GPU to run the path-follower, the high-level control, and semantic and life-long mapping algorithms. In addition, the computer vision tasks are deployed on an NVIDIA Drive AGX Xavier. All high-bandwidth sensors such as the cameras are interfaced to the AGX unit, which directly processes the data. Both compute devices run Linux as its operating system.

*Sensors*. In the open pit part of the hybrid mine, GPS localisation is possible. For this task, we deployed an OxTS RT3000v3 dGPS on one vehicle and equipped the two remaining vehicles with two OxTS xNav650 from Oxford Technical Solutions. The RT3000 shares its correction data via the xNAV650 devices. Each dGPS contains an IMU. Especially, in the underground part of the hybrid mine, where no GPS is available, we scanned the environment and made use of two VLP-16 Lidars and six cameras for localising and navigation. A mesh network spanning over multiple local stations in the testing area is used for vehicle interaction within the fleet and vehicle interaction with loading and unloading stations (V2X in **Figure 4**). Local access points are available on each of the vehicles that are connected to the mesh. For identification of the vehicles and direct addressing, we make use of ROS2 namespacing.

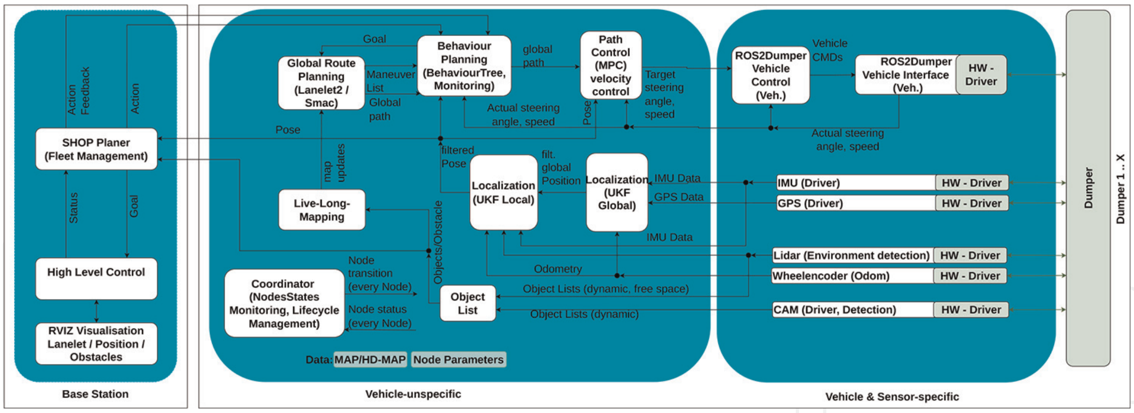

Action Feedback
Action
SHOP Planer (Fleet Management)
Status
Goal
High Level Control
RVIZ Visualisation Lanelet / Position / Obstacles
Base Station
Goal
Global Route Planning (Lanelet2 / Smac)
Maneuver List
Global path
Behaviour Planning (BehaviourTree, Monitoring)
global path
Path Control (MPC) velocity control
Target steering angle, speed
Actual steering angle, speed
Pose
Pose
map updates
filtered Pose
filt. global Position
Localization (UKF Local)
Localization (UKF Global)
IMU Data
GPS Data
IMU Data
Live-Long-Mapping
Objects/Obstacle
Odometry
Node transition (every Node)
Node status (every Node)
Coordinator (NodesStates Monitoring, Lifecycle Management)
Object List
Object Lists (dynamic, free space)
Object Lists (dynamic)
Data: MAP/HD-MAP Node Parameters
Vehicle-unspecific
ROS2Dumper Vehicle Control (Veh.)
Vehicle CMDs
ROS2Dumper Vehicle Interface (Veh.)
HW - Driver
Actual steering angle, speed
IMU (Driver) HW - Driver
GPS (Driver) HW - Driver
Lidar (Environment detection) HW - Driver
Wheelencoder (Odom) HW - Driver
CAM (Driver, Detection) HW - Driver
Dumper
Dumper 1 .. X
Vehicle & Sensor-specific

**Figure 4.**
*Diagram of the software architecture (From: [18]).*

## 3. Software architecture

In [21], we proposed a ROS 2-based architecture for self-driving cars. For our LHD vehicle, we adopted the architecture to fit the needs of the vehicle. Figure shows an overview of our architecture. The architecture also allows for using some common ROS packages such as Navigation2 [31] and robot_localisation [32]. In the following, we give an overview of the different parts of the architecture.

*Centralised Mission Management Block*. At the top-most level in the hierarchy of decision-making and control of the fleet is the high-level control block (leftmost block in **Figure 4**). In the LHD context, it means that a certain tonnage that should be hauled is defined and the high-level system is to find a plan for all the vehicles in the fleet to reach the tonnage. We deploy a SHOP3 [33] planning system for high-level planning. An additional fleet manager distributes the plans and dispatches the actions to the vehicles. Whilst the vehicles execute these actions, they continuously update their current positions and these of their locked resources. The world model gets information of the actual status of all agents and the hauled tonnage. If needed, in case of bigger differences to the original plan, a re-planning for the whole fleet is initiated.

*Vehicle Unspecific Block*. The vehicle unspecific section of the architecture consists of mid-tier functionalities, that are similar to three-tier robotics architectures with modules for localisation or path planning. A global route is planned via a free-space planner like smac or map-based planner using a Lanelet2 HD map. The vehicle follows this path utilising a path-following module, including a feedback loop [34, 35]. As path-following algorithm, we use a model-predictive controller (MPC) which uses a GPU-based grid-search on a set of predicted trajectories achievable by the vehicle's kinematic model. We introduce the details of the MPC in Section 4.

*Vehicle Specific Block*. The vehicle specific block (rightmost block in **Figure 4**) mostly consists of drivers and vehicle communication modules such as drivers for cameras, IMUs, GPS units, LiDARs, or the wheel encoders. For object detection, we have implemented an object detection with cameras using YOLOv5 to aid in the semantic and life-long mapping that is being presented in Section 6. We also implemented a modular way of integrating state-of-the-art deep neural networks for 3D object detection in point clouds. The detailed presentation of the latter is part of previous work presented in [36]. Our presented architecture is implemented using ROS 2 and deployed to each vehicle of the fleet. The underlying DDS network of ROS 2 is also used for the communication with the centralised high-level control system.

Next, we address the model-predictive controller used on the vehicle, before we discuss the high-level control software in Section 5.

## 4. Model-predictive control

In this section, we show the low-level control system based on a model-predictive control approach. We first introduce the kinematic model of the dumper, before we discuss the software implementation. Further details of the approach can be found in [17].

The kinematic model that has been used to model the haul-dumper in the MPC is described in [37] and shown in **Figure 5a**. This model has been chosen, as it models a centre-articulated platform and steering is done by changing the angle of the active joint. This model fits our haul-dumper well. The model is continuous and therefore the equations need to be discretised for our use case. The equations in the discrete form are as follows:

$$
\begin{aligned}
x_{t+1} &= x_t + \Delta_t{}^* (v\cos\psi) \\
y_{t+1} &= y_t + \Delta_t{}^* (v\sin\psi) \\
\psi_{t+1} &= \psi_t + \Delta_t{}^* \left( \frac{\sin\phi}{l_2 + l_1\cos\phi} v + \frac{l_2}{l_2 + l_1\cos\phi} \omega \right)
\end{aligned}
$$

The constant $l_1$ describes the length of the front part, $l_2$ the rear part of the vehicle, $x$ and $y$ are the current position values within the Cartesian coordinate system, and $v$ the current velocity of the vehicle. The current steering angle is $\phi$, $\omega$ denotes the angular velocity of the joint, and $\psi$ is the heading of the front part within the Cartesian coordinate system as shown in **Figure 5a**. $\Delta_t$ represents the time interval between two control steps within the prediction process. These equations are used to predict the travel of the vehicle given the different inputs for each iteration of the optimisation.

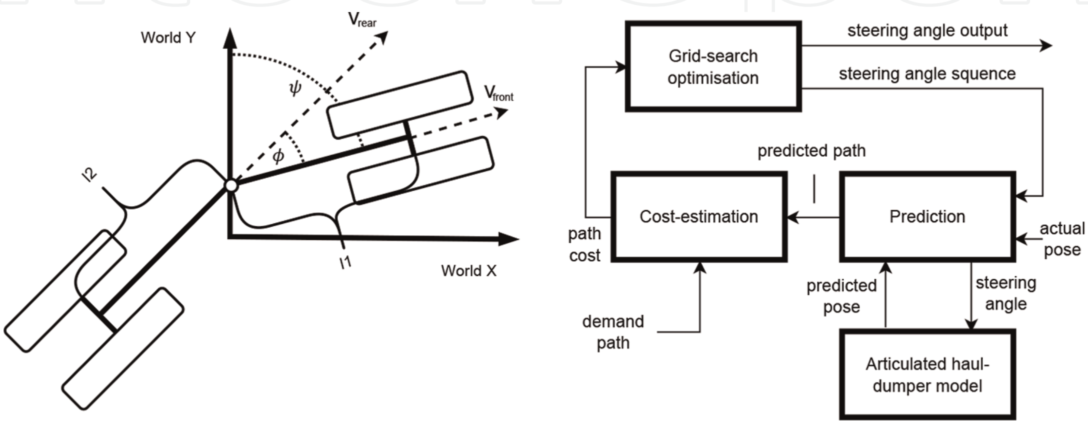


**Figure 5.**
*Dumper kinematics and model-predictive control approach (From: [17]).*

**Figure 5b** shows the MPC control cycle. Based on the actual pose and the articulated haul-dumper model (**Figure 5a**), a predicted path is calculated for a set of steering angle sequences. Subsequently, for each predicted path, the costs are estimated by three weights:

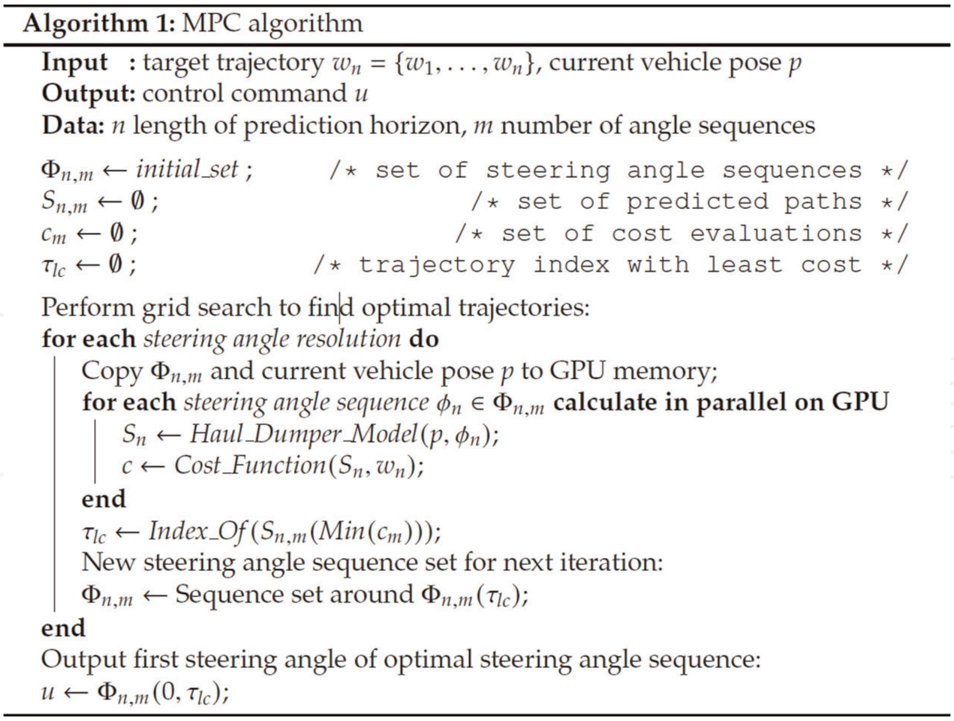
1. the *lateral error*, which weighs the difference in the vehicle's heading on the predicted path in contrast to the target trajectory;

2. the *orientation error*, which weighs the difference in the vehicle's lateral position on the predicted path in contrast to the target trajectory;

3. the *joint angle change*, which weighs the needed change of the joint angle from step to step.

Finally, in a grid-search optimisation, the output is determined as the first steering angle from the sequence of the steering angles for which the lowest cost was estimated.

The control result of this approach is very good, however, the calculation costs are very high. We therefore chose a GPU-based implementation of the controller based on our experiences described in [38]. The advantage of this approach is that the predicted paths and their individual costs can be calculated for all steering angle sequences in parallel in one step of the GPU. This leads to an improved control result at efficient calculation costs as we showed in [38].



**Algorithm 1:** MPC algorithm

```
Input  : target trajectory $w_n = \{w_1, \dots, w_n\}$, current vehicle pose $p$
Output: control command $u$
Data: $n$ length of prediction horizon, $m$ number of angle sequences
$\Phi_{n,m} \leftarrow$ initial_set ;        /* set of steering angle sequences */
$S_{n,m} \leftarrow \emptyset$ ;                      /* set of predicted paths */
$c_m \leftarrow \emptyset$ ;                         /* set of cost evaluations */
$\tau_{lc} \leftarrow \emptyset$ ;              /* trajectory index with least cost */
Perform grid search to find optimal trajectories:
for each steering angle resolution do
    Copy $\Phi_{n,m}$ and current vehicle pose $p$ to GPU memory;
    for each steering angle sequence $\phi_n \in \Phi_{n,m}$ calculate in parallel on GPU
        $S_n \leftarrow$ Haul_Dumper_Model$(p, \phi_n)$;
        $c \leftarrow$ Cost_Function$(S_n, w_n)$;
    end
    $\tau_{lc} \leftarrow$ Index_Of$(S_{n,m}(Min(c_m)))$;
    New steering angle sequence set for next iteration:
    $\Phi_{n,m} \leftarrow$ Sequence set around $\Phi_{n,m}(\tau_{lc})$;
end
Output first steering angle of optimal steering angle sequence:
$u \leftarrow \Phi_{n,m}(0, \tau_{lc})$;
```

The algorithm of the controller is shown in Algorithm 1, where the separation of GPU and CPU calculations is shown. To get an optimal trajectory resolution, multiple iterations are executed. The whole algorithm is calculated at a period of 20 ms. To adapt the original algorithm from [38] to the here-used long-haul-dumpers, we only had to change the kinematic model of vehicle and had to re-calibrate the control parameters (cf also to [17] for details).

## 5. High-level fleet planning

For larger-scale mining scenarios, several vehicles need to work together in a fleet. Such a fleet then needs to be coordinated for the work to be organised and distributed amongst the different vehicles in an optimised fashion. To realise the high-level fleet coordination system, we implemented a planning server embedding of SHOP 3 [33]. SHOP 3 is a domain-independent planning system based on ordered task decomposition. Ordered task decomposition is a modified version of hierarchical task network (HTN) planning [39] in which the planning order respects the actual order of execution of each task [33]. The interaction between the embedding sever and our vehicles is done by first communicating or loading a world model into the planner. Afterwards, a query for a day plan can be sent to the planner as a string, which the planner then processes. The query is a problem statement describing the resource allocation, available agents, and goals. As a result, the planner produces a string containing the day plan for all vehicles. This plan can then be distributed to the fleet. Each agent can then parse the plan and start executing the assigned actions.

As an example, **Figure 6** visualises a resource assignment. The blue circles in the figure represent resource sources. Above each source, we show the type of resource which can be loaded from that source and its quantity (rounded corners rectangles). The resource quantities of ores are modelled as infinite. The waste resource is constrained to 500 tons. The white circle represents a stockpiling point, which would be used to store resources if direct hauling to the unloading points would lead to congestion. In black, we colour the waste dumping points which can be used to unload waste from sources S1 and S2. The circles in yellow are fuel stations, that agents need to drive to fill their fuel tanks. Each agent in the problem statement is defined with a fuel state which is being used up when the agents transport resources. The arrows in the visualisation show possible direct load/unload relations. For example, the arrow from S1&2 to G13 means that the agent loads from S1 or S2 and unloads at the goal G13.

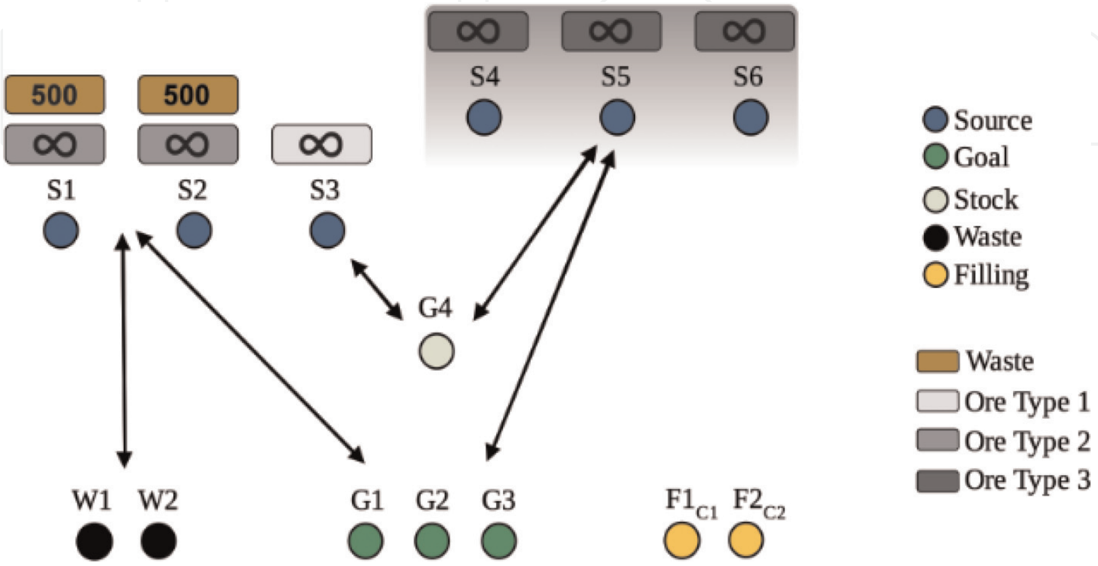


**Figure 6.**
*Schematic of one of the problem statements evaluated in our research.*

| Module | KPIs | | | | | |
|---|---|---|---|---|---|---|
| Model Predictive Control | Mean lateral error in curves | Mean lateral error on straight section | | | | |
| | 0.9 (m) | 0.1 (m) | | | | |
| System Delays | Mean steering delay | Mean engine RPM delay | | | | |
| | 200 (ms) | 100 (ms) | | | | |
| Mission Planning (SHOP3) | Idle-time (min) of fleet after 8 h in simulation | | | | | |
| | Scenario Name | Number of vehicles simulated | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 4 |
| | Simulation Scenario 1 | 75 | 25,782 | 26,096 | 26,046 | 25,938 |
| | Simulation Scenario 2 | 40 | 2093 | 5343 | 3512 | 3084 |
| | Overall hauled resources (t) | | | | | |
| | Scenario Name | Number of vehicles simulated | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 4 |
| | Simulation Scenario 1 | 75 | 18,550 | 18,550 | 18,550 | 18,550 |
| | Simulation Scenario 2 | 40 | 13,865 | 12,685 | 13,080 | 10,305 |
| Object Detection | Model | mAP@0.5 Person | mAP@0.5 Wheel-dumper | mAP@0.5 Car | mAP@0.5 Beacon | Detection Frequency [hz] |
| | PointPillar (3D)[a] | 0.34 | 0.4 | 0.4 | — | 20 |
| | YOLOv5m (2D) | 0.994 | 0.978 | — | 0.966 | 30 |

[a] *The results exclude performance for beacons, because the LiDAR sensor did not measure many points on the target beacons. Detection of beacons was, therefore, unreliable with the given hardware.*

**Table 1.**
*Table on the quantified key performance indices for some of our modules we have researched by now (From: [18]).*

Another problem statement we defined for the evaluation of the high-level control is less complex with only 5 sources with infinite resources and 2 goals. The meaning of sources and goals is the same as for **Figure 6**. The evaluation presented as an aggregation of key performance indicators in **Table 1** in Section 7 shows the results when different criteria are imposed on a plan. A plan can be made to take into consideration waiting times of the vehicles in a fleet, the time a vehicle needs to fulfil a load-unload cycle accumulated for a whole fleet, or idle time of the fleet as a whole. For comparison, we show the results for a plan that is created by generating plans randomly and selecting the first valid plan for a problem. This is the fourth strategy as presented in the evaluation. The plan that takes into consideration the waiting time of the agents is strategy 1. Strategy 2 takes the accumulated cycle time for all agents into account and tries to minimise it. The last strategy, strategy 3, minimises the time the vehicles in the fleet spend idling.

In our scenario, SHOP 3 instructs the navigation system to execute an action called DRIVE_TO. The action's main argument is an ID to a lanelet [10] in a map representing the routing graph. Lanelet is an HD map format which is an extension of OSM. With its integrated extensions for routing, it can support the execution of many tasks in navigation. Obstructions in the global map are represented within the routing graph and can be derived from an occupancy grid, which is implemented in the costmap ROS package. Given the goal, the action server runs a behaviour-tree [40] which computes the centre line of a route calculated by the lanelet_planner. This centre line is referenced to a world-fixed frame (i.e. the entrance of the mine) which is in turn passed on to our MPC to follow the path. Regular actuation commands are calculated by the MPC by calling its respective action server from within the behaviour-tree at a certain frequency.

## 6. Life-long mapping

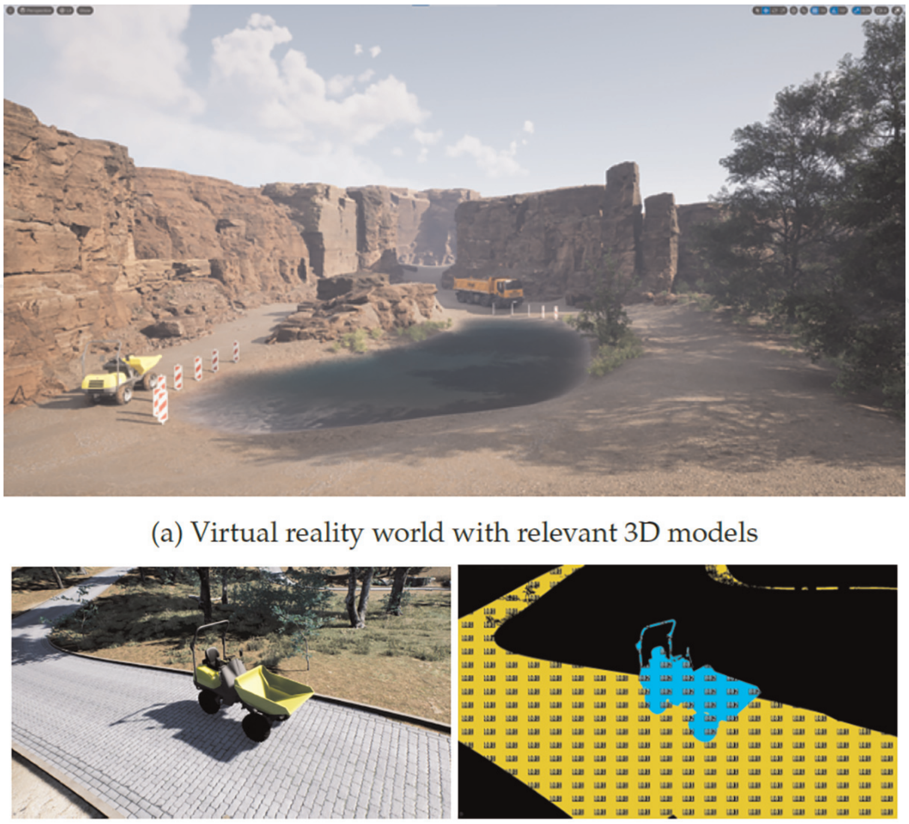
With life-long mapping, we refer to the concept of mapping the environment and ensuring that changes in the environment can be detected and processed into a map update. To achieve this, we identified two approaches. First, one can use representations of a traffic network as maps and use static delimiting objects in the environment as markers that one can infer a change in the environment. Secondly, one can use shape representations of the environment (such as a point cloud) to extract a traffic network representation of the environment from the shape. To accommodate for changes in the environment, one can simply repeat the process of generating the traffic network from the environment.

For the first approach, three major components are needed: the object detection and classification of static and dynamic objects, the localisation of static object in a unified coordinate frame, and the integration of the information from the 2 previous components into an HD map. In the following, we explain this in more detail:

*Object Detection*. The artificial intelligence framework known as YOLOv5 was used for object recognition. The focus was on the recognition of humans, navigation aids such as beacons marking the way, and different types of vehicles commonly used in mining. To facilitate the recognition of previously unknown objects using this YOLOv5 mechanism, a three-part training methodology was devised:

Firstly, a solution using synthetic data was developed to automate the annotation process for training deep learning networks. By creating a realistic 3D mining world with Unreal Engine and capturing annotated images from virtual cameras on dumpers, see **Figure 7**, these synthetic datasets serve as cost-effective training data for the YOLOv5 network, addressing the challenges of manual annotation for new objects and environments.

As second stage, synthetically generated data for neural network training often lacks impurities found in real-world images, such as noise or blur. To address this, we developed an image augmentation tool. This tool introduces variations into the training images by adding noise, modifying lighting, saturation, image resolution, and horizontal alignment. By incorporating these variations, the synthetic training data becomes more realistic and better aligns with real-world conditions. Thirdly, the training data is supplemented by data from real journeys which are manually annotated and augmented.



(a) Virtual reality world with relevant 3D models

(b) Simulated image

(c) Labeled image

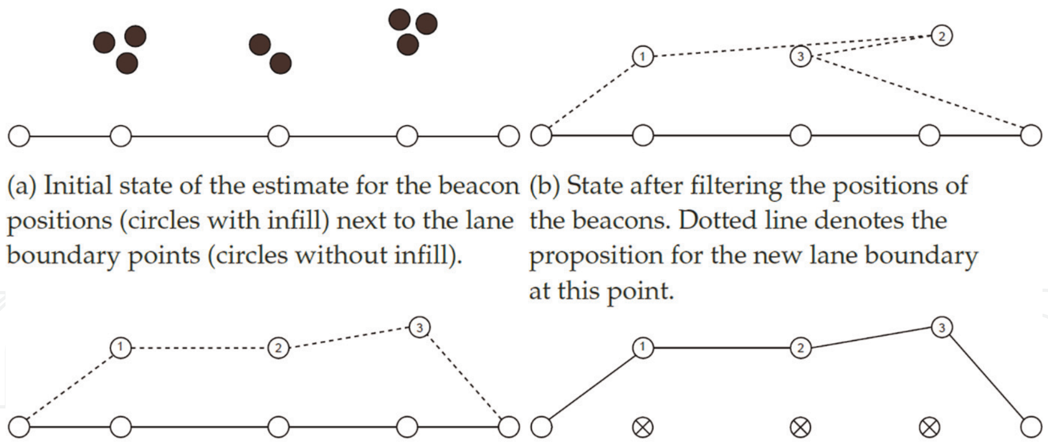
**Figure 7.**
*Virtual reality world with 3D models created with unreal engine 5 to get annotated training data from simulated driving.*

The classifier, trained with a blend of synthetic and real images over 299 epochs, attains 0.9835 mAP@0.5 and 0.9836 mAP@0.95 with YOLOv5m and effectively detects partially obscured objects. After converting from PyTorch to ONNX, then to TensorRT, it is deployed on Nvidia AGX Drive, with an inference time near 35 milliseconds.

Lane detection. Beacons serve as static boundary markers for navigable roads. We calculate the 3D position of these beacons by projecting the midpoint of the lower boundary box onto a ground plane, considering the camera's extrinsic parameters relative to the vehicle centre point. Finally, we derive the global UTM coordinates from the estimated vehicle-relative positions.

*Boundary Matching and Map Correction*. In order to correlate the positions of the beacons with a lane boundary, a filtration procedure is implemented. A binning filter reduces measurement noise by averaging close positions. A logical constraint filter ensures feasible interpolation of the lane boundary. Positions are then re-indexed based on their medial distance from the lane boundary segments. The updated lane boundary state is shown in **Figures 8** and **9**.

*LiDAR Map To HD Map*. Our approach for generating an HD map from a LiDAR map involves segmenting the navigable ground segment from the point cloud, calculating the concave hull, creating a Voronoi graph [41], finding the longest chain of vertices in the graph, smoothing the vertices, and converting the trajectory into a lane. The process includes steps such as elevation map creation, filtering of sample points based on a simple morphological filter from [42],



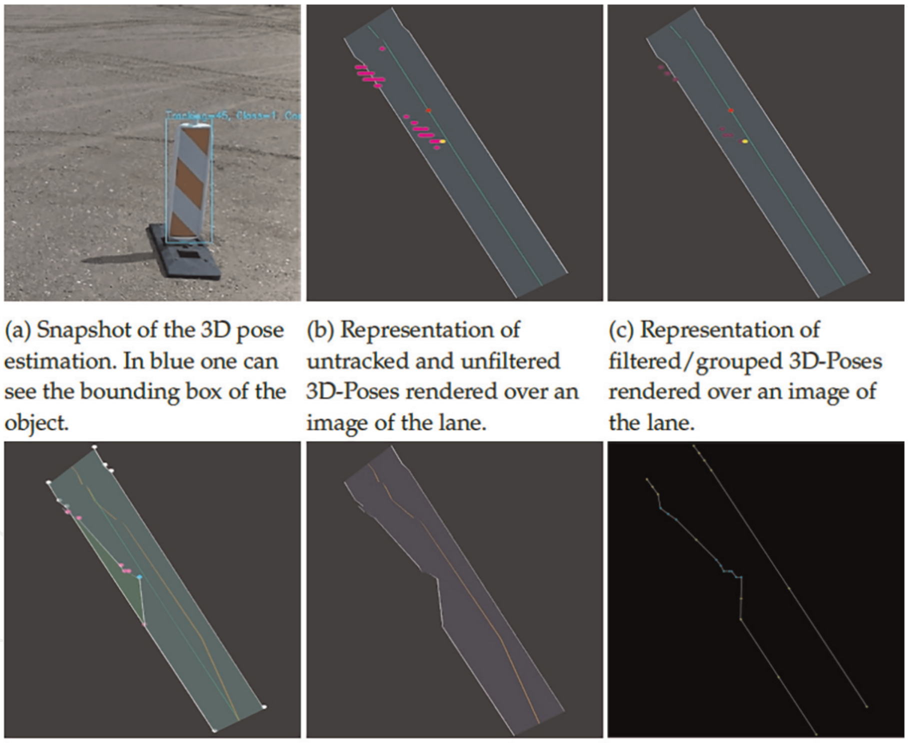
(a) Initial state of the estimate for the beacon positions (circles with infill) next to the lane boundary points (circles without infill).

(b) State after filtering the positions of the beacons. Dotted line denotes the proposition for the new lane boundary at this point.

(c) State after re-indexing the beacon-position groups.

(d) In the final stage, outdated lane boundary points are discarded to form a new lane boundary.

**Figure 8.**
*State of the lane boundary throughout the four stages of the lane boundary adjustment process.*



(a) Snapshot of the 3D pose estimation. In blue one can see the bounding box of the object.

(b) Representation of untracked and unfiltered 3D-Poses rendered over an image of the lane.

(c) Representation of filtered/grouped 3D-Poses rendered over an image of the lane.

(d) Update proposal rendered on top of image of the lane with changed centre-line.

(e) Image of the updated lane.

(f) Updated and valid lane rendered with the editing tool JOSM.

**Figure 9.**
*Visualisation of the update process and its result.*

calculating a spanning polygon, applying Delaunay triangulation, and constructing a lane with a consistent width. **Figures 10** and **11** show a visualisation of each step.

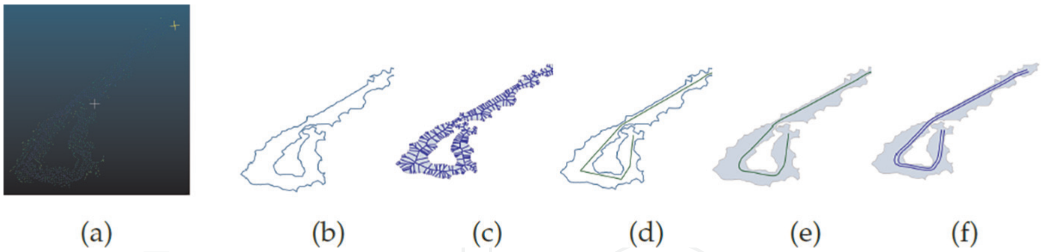


**Figure 10.**
*Processed point cloud map after each step of the conversion from point cloud map to HD map.*

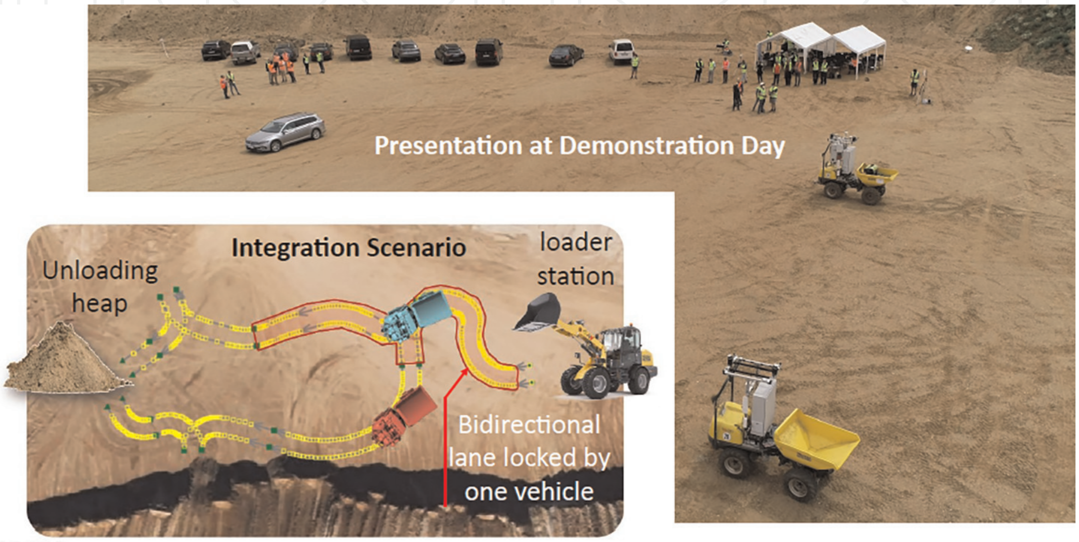

Presentation at Demonstration Day
Integration Scenario
Unloading heap
loader station
Bidirectional lane locked by one vehicle

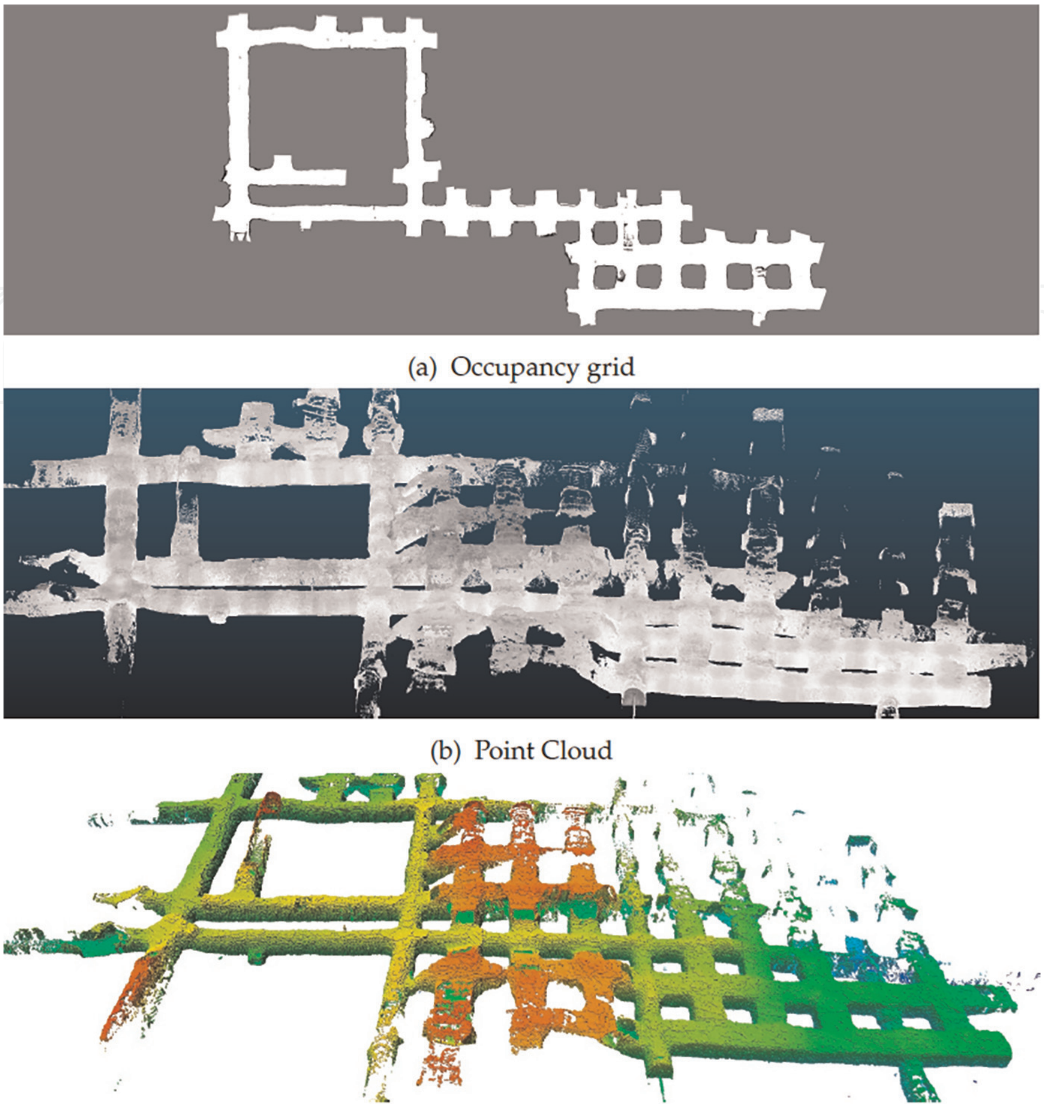
**Figure 11.**
*Visualisation of the integration scenario, which was presented at a demonstration day.*

## 7. Real-world experiments

In this section, we show some experimental results of our work.

3D Mapping. Regarding underground mapping with the exploration vehicle described in Section 2.1, we show maps from the MAXIT underground anhydrite mine in Krölpa, Germany in **Figure 12**. The map was recorded by exploration vehicle in a stop-and-scan fashion taking a full spherical 3D scan every 10 m. The recording of the map took place in a teleoperated manner, the point cloud data and the odometry data were stored in a rosbag file and were processed offline following the procedure described in detail in [19, 20]. **Figure 12a** shows a 2D occupancy grid of the mapped part of the mine. **Figure 12b** visualises the point clouds themselves and **Figure 12c** a 3D Octomap from the point clouds. Whilst the results show that precise 3D maps of the underground mining environment can be produced with the exploration vehicle, one has to admit, on the downside, that the process is not fully automated and human expertise is required to avoid errors in registering the different point clouds into a consistent map. The overall map size of the map is about 800 m.

*Vehicle Automation.* The haul-dumper's hardware setup has proved reliable throughout our project, with hundreds of testing hours and no system-wide failures. The most frequent issue involves power supply to the vehicle's components. The installed PLC and sluggish system caused by the hydraulic mechanism lead to an average latency of 200 ms from command issuance to actuators. There is also latency


(a) Occupancy grid

(b) Point Cloud

(c) Octree

**Figure 12.**
*Parts of the MAXIT anhydride mine in Krölpa, Germany.*

in emergency stops, taking 1.5 seconds to fully halt after triggering the switch. The path-following feature shows a mean lateral deviation of 0.1 m on straight sections and 0.9 m on curves.

*Fleet-level Planning.* Fleet-level planning was tested through coarse simulations that mimic mining processes and through integration in the lower modules of the architecture. Two simulation scenarios were created with different levels of complexity: one with simple targets and resources, the other emulating a hybrid mine with multiple resource types. In each scenario, four strategies were analysed, each examining a different heuristic: Minimising the idle time of all vehicles in the fleet, minimising the average duration of loading and unloading the dump truck, minimising the idle time of the dump truck without moving loads, and maximising the haulage mass from randomly generated plans. Idle time and total haulage mass were observed in an 8-h simulation. The results are shown in **Table 1**.

*Long-term Mapping.* We carried out a functional test on live map correction by manually altering the map and monitoring the global route planner and vehicle response. This often involved tweaking the lane sections leading to the loading station, as demonstrated in the integration scenario **Figure 11**.

*Integration Scenario.* In an integration test, we utilised loading and unloading stations to handle pick-up and unload payloads. The unloading heap has two access points, whereas the loading station has only one. There are two bidirectional lanes to the unloading sites and a single bidirectional lane to the loading site.

The daily transport schedule of fleet management includes repeated loading and unloading operations for the vehicle fleet, with only one vehicle having access to the loading point. The high-level controller coordinates resource access to avoid conflicts. Resource blocking is implemented by the operations scheduling system. The first vehicle to reach the fork in the middle of the traffic network whilst the loading resource is free secures the lock and blocks the lane to the resource. Following vehicles must wait for fleet management to release the resource. This mechanism proved robust in an 8-h operation and successfully implemented the daily schedule.

## 8. Conclusion

In this chapter, we present our results from automating load-haul-dump (LHD) operations in hybrid mines. Hybrid mining operations are mines where mining in part is done in an open pit fashion, and parts are underground. As for the autonomous vehicles deployed in such a mine, it means that the vehicles cannot simply rely on GPS data for localising themselves over ground, but also need classical approaches to mapping the environment. We reported on our exploration vehicle that is equipped with a rotating LiDAR scanner to produce detailed 3D point clouds which then are integrated into Octree maps that the LHD vehicles could use for localising themselves in the mine.

The haul-dumpers that we deploy are modified off-the-shelf articulated dumpers which were turned into autonomous vehicle. The vehicles are equipped with GPS, LiDAR, and camera sensors. The software architecture is based on ROS2 and was also deployed in similar projects related to autonomous driving including a modified version of a model-predictive control (MPC) algorithm. It had to be adapted to the articulated kinematics of the dumper. Further, to control a fleet of LHD vehicles (three in our case), we made use of the HTN-based planner SHOP3, which generates a global mission plan for the vehicles based on the required haulage capacities. Each vehicle is following this global plan for generating their local missions. As another important contribution, we introduced Lanelet2-based maps for navigating the LHD vehicles, including dedicated drive ways and right of way rules. Lanelet maps are commonly used for self-driving cars and not so much in robotics applications. Using this approach facilitates the coordination of a fleet of vehicles to a great extend. Finally, we showed our approach to object and drive-way detection including the automatic generation of the aforementioned Lanelet drive ways.

The presented approach has been tested in real-world scenarios underground and in open pit mines under controlled conditions. We showed how mapping successfully took place with our exploration vehicle in the Krölpa mine in Germany. The fleet of dumpers was tested in an open gravel pit in Buir, Germany. We could show that the low-level control such as MPC works well also on the dumpers with their articulated kinematics and how the fleet of dumpers could be coordinated on a mission level. Whilst the tests and experimental results show that the overall approach is working, next steps would be to deploy this work in real mining operations under the realistic hard mining environment conditions with limited communication means.

## Acknowledgements

We gratefully acknowledge the contributions of (in alphabetical order) N. Adam, A. Braining, D. Bulla, M. Claer, B. Decker, J. D. Eichenbaum, F. Engels, J. Grabe, K. Kramer, K. Krückel, M. Lubos, A. Lyrmann, T. Neumann, Y. Otten, J.C. Richter, D. Scholl, P. Shah, P. Stricker, and S. Weyer. The work presented in this chapter was funded with grants from the German Federal Ministry of Education and Research (BMBF) under grant numbers 033R126C, 033R126CN.

## Author details

Alexander Ferrein*, Gjorgji Nikolovski, Nicolas Limpert, Michael Reke,
Stefan Schiffer and Ingrid Scholl
Mobile Autonomous Systems and Cognitive Robotics Institute, FH Aachen -
University of Applied Sciences, Aachen, Germany

*Address all correspondence to: ferrein@fh-aachen.de

## References

[1] Lööw J, Abrahamsson L, Johansson J. Mining 4.0—The impact of new technology from a work place perspective. Mining, Metallurgy & Exploration. 2019;**36**:701-707

[2] Faz-Mendoza A, Gamboa-Rosales NK, Medina-Rodríguez C, Casas-Valadez MA, Castorena-Robles A, López-Robles JR. Intelligent processes in the context of mining 4.0: Trends, research challenges and opportunities. In: 2020 International Conference on Decision Aid Sciences and Application (DASA). IEEE; 2020. pp. 480-484

[3] Batterham R. The mine of the future – Even more sustainable. Minerals Engineering, Sustainable Minerals. 2017; **107**:2-7

[4] Sánchez F, Hartlieb P. Innovation in the mining industry: Technological trends and a case study of the challenges of disruptive innovation. Mining, Metallurgy & Exploration. 2020;**37**(5): 1385-1399

[5] Clausen E, Sörensen A. Required and desired: Breakthroughs for future-proofing mineral and metal extraction. Mineral Economics. 2022;**35**(3):521-537

[6] Paraszczak J, Gustafson A, Schunnesson H. Technical and operational aspects of autonomous lhd application in metal mines. International Journal of Mining, Reclamation and Environment. 2015;**29**(5):391-403

[7] Mascaró M, Parra-Tsunekawa I, Tampier C, Ruiz-del Solar J. Topological navigation and localization in tunnels—Application to autonomous load-haul-dump vehicles operating in underground mines. Applied Sciences. 2021;**11**(14):6547

[8] Jacobson A, Zeng F, Smith D, Boswell N, Peynot T, Milford M. What localizes beneath: A metric multisensor localization and mapping system for autonomous underground mining vehicles. Journal of Field Robotics. 2021; **38**(1):5-27

[9] Stefaniak P, Jachnik B, Koperska W, Skoczylas A. Localization of LHD machines in underground conditions using IMU sensors and DTW algorithm. Applied Sciences. 2021;**11**(15):6751. DOI: 10.3390/app11156751

[10] Poggenhans F, Pauls J-H, Janosovits J, Orf S, Naumann M, Kuhnt F, et al. Lanelet2: A high-definition map framework for the future of automated driving. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2018. pp. 1672-1679

[11] Dong L, Sun D, Han G, Li X, Hu Q, Shu L. Velocity-free localization of autonomous driverless vehicles in underground intelligent mines. IEEE Transactions on Vehicular Technology. 2020;**69**(9):9292-9303

[12] Seguel F, Palacios-Játiva P, Azurdia-Meza CA, Krommenacker N, Charpentier P, Soto I. Underground mine positioning: A review. IEEE Sensors Journal. 2022;**22**(6):4755-4771

[13] Bnouachir H, Chergui M, Machkour N, Zegrari M, Chakir A, Deshayes L, et al. Intelligent fleet management system for open pit mine. International Journal of Advanced Computer Science and Applications. 2020;**11**(5):327-332

[14] Seiler KM, Palmer AW, Hill AJ. Flow-achieving online planning and dispatching for continuous transportation with autonomous

vehicles. IEEE Transactions on Automation Science and Engineering. 2020;**19**(1):457-472

[15] Erol K, Hendler JA, Nau DS. UMCP: A sound and complete procedure for hierarchical task-network planning. In: Proceedings of the Second International Conference on Artificial Intelligence Planning Systems (AIPS'94). AAAI Press; 1994. pp. 249-254

[16] Eichenbaum J, Nikolovski G, Mülhens L, Reke M, Ferrein A, Scholl I. Towards a lifelong mapping approach using lanelet 2 for autonomous open-pit mine operations. In: Proc. IEEE CASE 23. IEEE; 2023

[17] Nikolovski G, Limpert N, Nessau H, Reke M, Ferrein A. Model-predictive control with parallelised optimisation for the navigation of autonomous mining vehicles. In: 2023 IEEE Intelligent Vehicles Symposium, IV 2023. IEEE; 2023

[18] Ferrein A, Reke M, Scholl I, Decker B, Limpert N, Nikolovski G, et al. Towards a fleet of autonomous haul-dump vehicles in hybrid mines. In: ICAART (1). SCITEPRESS; 2023a. pp. 278-288

[19] Ferrein A, Scholl I, Neumann T, Krückel K, Schiffer S. A system for continuous underground site mapping and exploration. In: Reyhanoglu M, Cubber GD, editors. Unmanned Robotic Systems and Applications. London, UK, Rijeka: IntechOpen; 2019

[20] Donner R, Rabel M, Scholl I, Ferrein A, Donner M, Geier A, et al. Die Extraktion bergbaulich relevanter Merkmale aus 3D-Punktwolken eines untertagetauglichen mobilen multisensorsystems. Tagungsband Geomonitoring. 2019;**S**:91-110. DOI: 10.15488/4515

[21] Reke M, Peter D, Schulte-Tigges J, Schiffer S, Ferrein A, Walter T, et al. A self-driving car architecture in ROS2. In: 2020 International SAUPEC/RobMech/PRASA Conference. IEEE; 2020. pp. 1-6

[22] Macenski S, Foote T, Gerkey B, Lalancette C, Woodall W. Robot operating system 2: Design, architecture, and uses in the wild. Science Robotics. 2022;**7**(66):eabm6074

[23] Neumann T, Dülberg E, Schiffer S, Ferrein A. A rotating platform for swift acquisition of dense 3D point clouds. In: ICIRA 9834 of Lecture Notes in Computer Science. Cham: Springer; 2016. pp. 257-268

[24] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, et al. Ros: An open-source robot operating system. In: ICRA Workshop on Open Source Software. Vol. 3. Kobe, Japan; 2009. p. 5

[25] Besl P, McKay ND. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis & Machine Intelligence. 1992;**14**(2):239-256

[26] Rusu RB, Cousins S. 3D is here: Point cloud library (PCL). In: 2011 IEEE International Conference on Robotics and Automation. IEEE; 2011. pp. 1-4

[27] Thrun S, Montemerlo M. The graph SLAM algorithm with applications to large-scale mapping of urban structures. The International Journal of Robotics Research. 2006;**25**(5–6):403-429

[28] Lu F, Milios E. Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems. 1997; **18**(3):249-275

[29] Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W. OctoMap: An efficient probabilistic 3d

mapping framework based on octrees. Autonomous Robots. 2013;**34**(3):189-206

[30] Sürken D. Programmierung einer Basisautomatisierung für eine Wacker Neuson 1501 Modellmulde [Thesis]. FH Aachen University of Applied Sciences in German; 2021

[31] Macenski S, Martin F, White R, Ginés Clavero J. The marathon 2: A navigation system. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE; 2020

[32] Moore T, Stouch D. A generalized extended kalman filter implementation for the robot operating system. In: Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13). Cham: Springer; 2014

[33] Goldman RP, Kuter U. Hierarchical task network planning in common lisp: The case of SHOP3. In: Proceedings of the 12th European Lisp Symposium (ELS), April 1-2, 2019, Genova, Italy. Zenodo; 2019. pp. 73-80. DOI: 10.5281/zenodo.2633324

[34] Pivtoraiko M, Knepper RA, Kelly A. Optimal, Smooth, Nonholonomic Mobile Robot Motion Planning in State Lattices. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University; 2007

[35] Limpert N, Schiffer S, Ferrein A. A local planner for Ackermann-driven vehicles in ROS SBPL. In: 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). IEEE; 2015. pp. 172-177

[36] Nikolovski G, Reke M, Elsen I, Schiffer S. Machine learning based 3D object detection for navigation in unstructured environments. In: IEEE Intelligent Vehicles Symposium Workshops, IV 2021. IEEE; 2021. pp. 236-242

[37] Delrobaei M, McIsaac KA. Design and steering control of a center-articulated mobile robot module. Journal of Robotics. 2011;**2011**:621879

[38] Chajan E, Schulte-Tigges J, Reke M, Ferrein A, Matheis D, Walter T. Gpu based model-predictive path control for self-driving vehicles. In: 2021 IEEE Intelligent Vehicles Symposium (IV). IEEE; 2021. pp. 1243-1248

[39] Dvorak F, Barták R, Bit-Monnot A, Ingrand F, Ghallab M. Planning and acting with temporal and hierarchical decomposition models. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. IEEE; 2014. pp. 115-121

[40] Ghzouli R, Berger T, Johnsen EB, Dragule S, Wasowski A. Behavior trees in action: A study of robotics applications. In: Proceedings of the 13th ACM SIGPLAN International Conference on Software Language Engineering. New York, NY, United States: Association for Computing Machinery; 2020. pp. 196-209

[41] Bhattacharya P, Gavrilova ML. Voronoi diagram in optimal path planning. In: 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007). IEEE; 2007. pp. 38-47

[42] Pingel TJ, Clarke KC, McBride WA. An improved simple morphological filter for the terrain classification of airborne lidar data. ISPRS Journal of Photogrammetry and Remote Sensing. 2013;**77**:21-30