# Flexible Infrastructure Free Navigation for Vehicles in Underground Mines

Johan Larsson, Mathias Broxvall, Alessandro Saffiotti

***Abstract*—During the last decade, mining companies and mobile equipment manufacturers have pursued improved efficiency, productivity, and safety in underground mining operations by automating some of the functions of underground vehicles. The work presented in this paper is the result of an effort to develop new flexible infrastructureless guidance system for autonomous tramming of center-articulated underground mining vehicles.**

***Index Terms*—mobile robots, mining robotics, reactive navigation.**

## I. INTRODUCTION

This paper describes our work towards a new infrastructureless guidance system for autonomous tramming of underground load-haul-dump (LHD) mining vehicles. We also present the results of experiments that have been performed on two different mobile robots, one research robot and one LHD vehicle, to verify the functionality of the system. Different parts of the autonomous navigation system presented here have previously been described in [1], [2] and [3]. Despite previous efforts by several parties to automate the tramming function of underground machines, widespread adoption of such technologies has yet to occur in the minerals industry. It has been reported by mine operators that this is, at least in part, due to poor reliability and lack of robustness in these technologies. Thus, the purpose of the described project was to design a flexible, reliable, and robust "autotramming" technology that does not require the installation of fixed infrastructure throughout the mine. Several factors make autonomous tramming in underground mines a challenging task, but two main requirements for commercial navigation systems for LHDs can be defined:

- To enable autonomous navigation the robot has to be capable of precise and real-time underground localization without dependency to specific infrastructure.
- With the rising demand of metal and minerals most mines are progressing rapidly creating a constantly changing environment. The high production rate also affects the rock surrounding the desirable ore and cave-ins or other unplanned events force the mines to re-plan their operation continuously. Therefore to be commercially successful, an autonomous navigation solution has to be flexible and easy to adapt to the changes in its working environment.

So far several solutions to autonomous tramming of LHDs have been suggested. Some of them suffer from relying on dedicated infrastructure like inductive wires [4], light-ropes [5] and reflexive tape [6]. A few solutions for autonomous tramming that are truly infrastructure free can also be found in the literature.

Fig. 1. LHD vehicle prepared with sensors and control system for autonomous navigation.

These systems, which all utilize laser range scanners and are based on a record and playback principle, span from feature-based/topological [7], feature-based metric [8] to occupancy gridmap based [9]. Currently, the two first systems are commercialized and in operation in underground mines. However, all of these systems suffer from either the need for infrastructure or that the paths to traverse have to be taught to the system, both shortcomings that are desirable to overcome.

In this paper, we present our approach to develop a flexible navigation system for operation of autonomous LHD vehicles in underground mines. The suggested system uses a coarse topological map for localisation, a fuzzy behaviour-based approach for navigation and is neither relying on infrastructure for localisation nor the need to teach a path to the system. In our work we have designed, developed and validated novel feature detection algorithms to enable reliable tunnel following as well as functions for topological localisation, path planning and navigation. To be able to quickly perform online experiments we started from an existing framework for autonomous navigation of research robots in indoor environments, and gradually changed the system to enable autonomous navigation of articulated vehicles in underground mines.

Our approach to develop an autonomous navigation system for a mine vehicle is divided in two main phases. In the first phase, with focus on fast and reliable feature detection, laser based feature detection algorithms and a topological navigation system was developed on a small research outdoor robot, starting from an existing framework for autonomous navigation. The experiments in this phase were performed in long corridors inside a building, and in a model of a mine with a drift width slightly less than two meters. In the second phase, the navigation system is ported to the existing control system of a real LHD vehicle, Figure 1, manufactured by Atlas Copco, on which experiments have been performed in an underground mine.

The feature detection algorithms that operate on data from a laser scanner to extract features such as tunnel center line and intersections have been described previously [2] [3] and will therefore not get any further attention here. Instead we focus on the overall architecture of the system and report on experiments performed to assess the performance of the full navigation system.

This work was partly funded by the Swedish organisation *Robotdalen*, and partly by the Swedish graduate school "Intelligent Systems for Robotics, Automation and Process Control" (RAP).

J. Larsson is with AASS, Örebro University, Örebro, Sweden and Atlas Copco Rock Drills, Örebro, Sweden. E-mail: johan.larsson@oru.se

M. Broxwall and A. Saffiotti are with AASS, Örebro University, Örebro, Sweden E-mail: {mbl, asaffio}@aass.oru.se

### A. Outline of this Paper

In the next Section we present our navigation system that fulfills the requirements mentioned above. Sections III and IV describe experiments performed on a small research robot respective on a real LHD vehicle. Section V concludes.

## II. THE NAVIGATION SYSTEM

The control system we report on in this paper is designed according to the principles of reactive navigation, as opposed to absolute navigation. Using the concept of reactive navigation means that the control system follows the path defined by the natural environment, for instance a mine tunnel, instead of a metric path related to some global coordinate system as is the case with absolute navigation systems.

Using reactive navigation in this sense does not mean that the control system has to be implemented as a set of reactive behaviours according to the Reactive paradigm, even though such an approach might seem natural to use. Compared to absolute navigation the reactive navigation relaxes the global localisation function since the global localisation in a reactive navigation system is only needed for making high level decisions such as which way to go in a junction.

Our control system is implemented according to the Hybrid Deliberative/Reactive paradigm and based on an existing framework for autonomous navigation, the Thinking Cap. The *Thinking Cap* is an integrated, layered sensing and control system for autonomous robots based on fuzzy logic [10] [11], and is related to the the Saphira architecture. In our system we use a coarse topological map to represent the mine, and a behaviour based approach to navigate inside the mine using a sequence of reactive follow-tunnel behaviours. Data from a laser range scanner is used to maintain the robots relative position and orientation inside each tunnel, and a junction recognizer assess the vehicles topological position in the map. The junctions are mainly recognized as topological structures. By extracting and comparing specific features from the laser data, in practice the openings to side tunnels, to the expected topological structures in the map the system is able to keep track of its location. By using the natural structures the localisation and navigation is performed in pretty much the same way as a human operator would do it, e.g. "follow this tunnel approximately 200 m while passing two tunnels on the left side and then turn right in the third intersection".

### A. Thinking Cap

The *Thinking Cap* is implemented according to a two layered structure as shown in Figure 2. The lower level contains the sensori-motoric functionalities for perception and behaviour based control, while the cognitive functionalities for world modeling and goal-oriented planning belong to the higher layer. The glue that brings the two layers together is the LPS (Local Perceptual Space). The LPS is a geometric representation of the robots working environment. As such, it is designed to accommodate not only *a priori* information from sources such as maps, but also pure perceptions and interpretations of sensor information.

*1) The lower layer:* In the lower level the Thinking Cap is behaviour based. These behaviours are based on fuzzy logic and form small control units with capabilities such as corridor following or obstacle avoidance. However, unlike many other behaviour based architectures the behaviours do not form a strict "Sense – Act" pair. In the case of Thinking Cap the behaviours are related to the LPS rather than to specific sensors. In this sense the LPS is not only the central locus

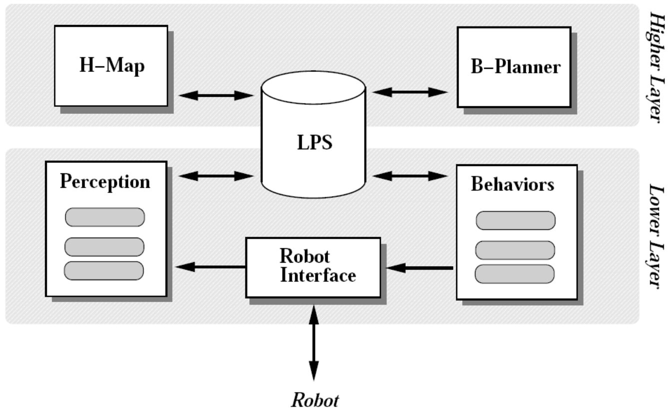


Fig. 2. Overall view of the Thinking Cap architecture

of integration between the higher and lower layer of the architecture. It is also the media through which the perception functions and the control behaviours communicate, forming a very flexible link where new sensors easily can be integrated in the control system.

As seen in Figure 2 the lower layer of Thinking Cap can be divided in four different modules, Perception, Behaviours, robot interface and LPS.

The original perceptual routines can be classified in two different groups, sonar based and vision based. Sonars are mainly used to detect obstacles, linear features and doors. Vision is used to recognize objects within a given set of types, and to measure their properties such as: color, shape, size and position relative to the robot.

The LPS gives the robot an awareness of its immediate environment, and is critical in the tasks of fusing sensor information, planning local movement and integrating map information. The perceptual and control architecture make constant reference to the LPS as it acts as a short term memory of the location and properties of the objects around the robot, and the location of the objects in robot-centered coordinates. The position of each object in the LPS is updated by three mechanisms:

- Perception, whenever the object is detected and measured by the robot's sensors
- Global information, whenever the robot knows its location in the global map and the object is stored in the map (e.g., corridors and doors)
- Odometric clamping, whenever the robot moves

The behaviours are based on fuzzy logic, and so are also the blending functions for combining behaviour output. Control behaviour output is typically forward velocity and angular orientation. Basic behaviours take their input from the LPS. Simple reactive behaviours, like obstacle avoidance, have rules whose antecedent depends on low-level information, like occupancy information, which is quickly available. However, behaviours can also use more complex data structures in the LPS. In this way certain *Goal–seeking* behaviours can be implemented. The Goal-seeking behaviours take their input from *artifacts*, i.e. objects in the LPS that have additional properties other than just the location, examples are doors and corridors which both have the additional properties *Width* and *Direction*. Thanks to this, purposeful behaviours such as "Cross–Door" and "Follow–Corridor" can have the same form as purely reactive behaviours, e.g. "Keep–Off".

*2) The Higher layer:* The higher layer contains the functionalities needed for reasoning globally about time and space.

As seen in Figure 2 the higher layer includes two modules in addition to the LPS, a Hybrid map and the *B-planner*.

The original framework utilizes a hybrid patchwork map [12]. The patchwork map contains a global topological map, representing the connectivity of sectors, and local geometric information of each sector. Each sector map is a Cartesian representation of a limited area of environment, like a room, a hall or a corridor. These local geometric maps contains not only approximate information about the boundaries of the environment, but can also include *Landmarks*. The landmarks specify the position and orientation of specific objects used for robot navigation. Examples are doors and corridor junctions. Landmarks are seen as intrinsic parts of the environment which do not change with time, and the system needs to be provided with the geometrical information of these landmarks a priori. When the map is available to the system the information is stored in the LPS as objects and artifacts, ready to be operated on by the behaviours, the perception and the planner.

The planner of the original framework has the ability to combine available behaviours to a *Behavioural Plan*. A Behavioural Plan is a set of behaviours combined to a plan by *Context–dependent blending* [11]. Such a plan includes a set of behaviours, each having its individual context of applicability in the form of a “Situation ⇒ Behaviour” rule.

### B. The mine navigation system

In our navigation system major changes have been made to the original structure of the Thinking Cap. From the original framework only the lower level and the LPS are used. New functionality has also been added to the perception module to support laser range scanners. This includes both feature extraction of tunnel properties as well as intersection detection and occupancy information for obstacle avoidance.

Initially it might seem strange to replace the higher layer of the original framework. However, the original map used in the Thinking Cap is a hybrid patchwork map with local metrical maps. Since no metric information can be assumed to be provided in advance for the mine navigation, and the original planner is dependent on the hybrid map the entire top level is discarded. Instead a topological map and a simple path planner based on standard search techniques have been implemented. The output from the path planner is a list of nodes to be traversed in order to reach the goal. Before the vehicle starts to navigate the path is transformed to a behavioural plan that can be executed by the behaviour module.

In the behaviour module we find a set of fuzzy behaviours. Fuzzy behaviours are easy to define and they provide robustness with respect to sensor noise, and to modelling errors and imprecision [13]. The behaviours that we use were originally developed for indoor, low-speed navigation [10].

The main behaviour used in our system is the “Follow” behaviour, which takes as input the attributes (orientation and lateral position) of the tunnel extracted from the laser data. Other behaviours used in our development include “Avoid” to perform obstacle avoidance, and “Orient” to orient in the direction of the tunnel when entering a new one.

The original behaviours only needed to be modified slightly in order to make them work in our setup and to navigate at the research robots top speed 1.7 m/s, or about 6 Km/h — the original behaviors were tuned for top speeds of about 0.3 m/s. However, major changes to the behaviours are needed when the navigation system is ported to the real LHD vehicle, which is characterized by more complex dynamics and kinematics, less clearance on the sides, and speed up to 30 Km/h.

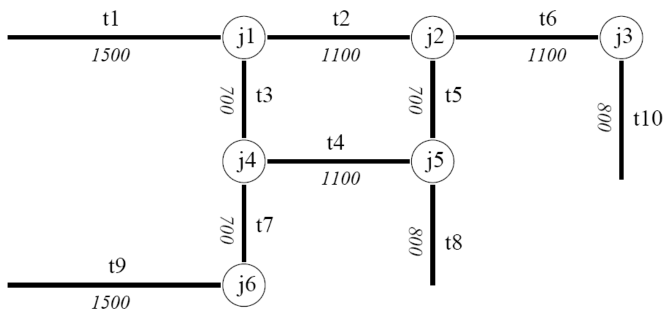


Fig. 3. Example of a topological map augmented with tunnel lengths.

At the top level, the navigation planner relies on the topological localization.

Like the path planner the original localisation functions of the Thinking Cap are dependent on the hybrid map. Therefore, new localisation functions based on our topological map have been developed. This localisation is a simplified single hypothesis tracking function [14] that is only able to localise itself within the planned path. This simplification is justifiable since the target vehicle has to stop immediately if it departs from the planned path.

To exemplify the operations of the complete system we consider the topological map from Figure 3. Assume that the vehicle starts at at the junction `j6` facing in the direction of tunnel `t7` and is given the goal to move to `j5`. The topological navigation planner will then generate the following behavioural plan which will be executed:

```
IF obstacle_near                         THEN Avoid()
IF nextNode(j4) AND NOT oriented(t7)     THEN Orient(t7)
IF nextNode(j4)                          THEN Follow(t7)
IF nextNode(j5) AND NOT oriented(t4)     THEN Orient(t4)
IF nextNode(j5) AND oriented(t4)         THEN Follow(t4)
IF nextNode()                            THEN Still()
```

Avoid, Orient, Follow and Still are fuzzy behaviours, activated according to the fuzzy predicates obstacle_near, nextNode and oriented. j4, j5, t4 and t7 are control system representations of objects in the map, for details see [10].

Data from the laser scanner is fed to the avoid-obstacle behaviour, and is also used to update the parameters of the tunnel artifacts for the follow behaviours. As the vehicle moves these two behaviours make the vehicle follow the center of the tunnel `t4` as long as the topological localization does not signal that the junction `j4` has been reached. When the junction `j4` is detected another tunnel is added to the local expectations on the right side, and the laser is used for localizing its exact position and the orient behaviour starts up. This behaviour uses first odometric information and eventually the laser readings to orient toward tunnel `t4` and when oriented this new tunnel is traversed by the follow behaviour.

## III. Evaluation of the mine navigation system on a research robot

Since the functionality of a control system based on fuzzy logic can not be fully evaluated with formal analysis this experiment was performed to empirically evaluate our control systems ability to navigate in mine like environments. The tests were performed in a model of a small mine consisting of approximately 2 m wide corridors, covering a 10x10 meter large area. In the experiment we used a small research robot to simulate the operation of a LHD vehicle in a mine, where the vehicle travels back and forth between the draw point and dump point. The used ATRV-Jr robot is displayed in Figure 4.


Fig. 4. The ATRV-Jr research robot used to evaluate the mine navigation system.

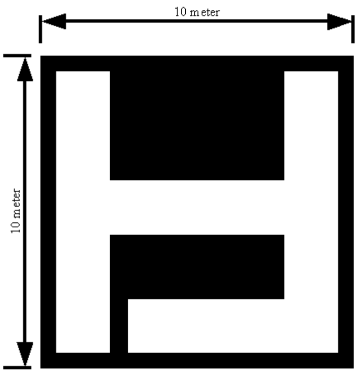


Fig. 5. Metric map of the environment used in the experiment

### A. Experimental Setup

To get good visibility of the robot no roof was fitted to the "tunnels", and as the setup was also intended for demonstration purposes the plywood board walls was kept as low as possible. A metric map of the test area is displayed in Figure 5.

To be able to order the robot to travel between two locations in the test environment a topological map of the test area was designed and provided to the robot, see Figure 6. The map consists of seven nodes, where four of them (N1, N4, N6 and N7) are dead ends, and the other three (N2, N3, N5) represent two different types of intersections.

In a mine LHD vehicles are mainly used to transport ore from a draw point to a dump point. To mimic the task of travelling back and forth between the draw and dump points, our navigation system was extended with functionality to continuously move between two nodes. This was implemented in a very simple way by just toggling between two target nodes. Every time the robot reached its goal the target node was switched and the path re-planned.

A real LHD vehicle is designed to be able to travel both backwards and forwards on equal conditions since it has no means of turning around in a mine. In other words, a LHD would typically move forward when approaching the draw or dump point and backwards after the loading or dumping operation has been done. Unfortunately our ATRV-Jr is only equipped with one laser scanner, which means that the laser

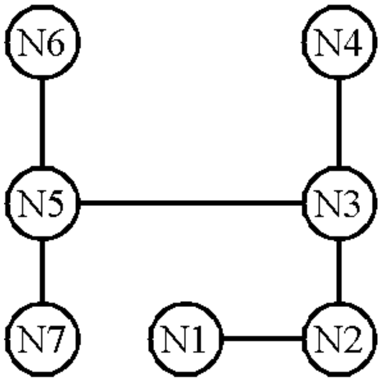


Fig. 6. The topological map of the environment in the experiment.

based autonomous navigation can only be performed in the forward direction. Therefore the robot had to turn around each time a target node had been reached, and the path was re-planned to a new target.

### B. Result

The experiment includes several hours of autonomous navigation, where the robot moved between two of the nodes in the test area. The main bulk of the test time were carried out during an exhibition, that covered about 3 hours of active navigation and 88 missions[1]. At the exhibition the spectators had the opportunity to choose two nodes between which the robot should travel. Most of the time, about 80% of the missions, the navigation system were able to navigate the robot between the two target nodes for many iterations until it was stopped manually and ordered another route. 18 of the missions failed. However, the lion share of these failures are related to the experimental setup and will not occur in an implementation on a mine vehicle. The failures can be divided in three groups:

1) Unable to turn around when reaching the target and the robot is ordered to go to a new location, 55% of the failures (11% of the runs)
2) Obstacles, 39% of the failures (8% of the runs)
3) Unable to detect intersection, 6% of the failures (1% of the runs)

The first error type occurred when the robot had to turn around 180° when receiving a new goal. The explicit cause was that the obstacle avoidance behaviour stopped the turn when the robot got too close to the tunnel walls when turning on the spot.

The second error type occurred due to obstacles in the "tunnels" of the test area. During many of the test runs humans entered the test area and blocked the way for the robot just to show that the navigation system can handle obstacles in its path, and recover to finish the task when the obstacles was removed. However, at a few occasions the obstacles caused the navigation to fail.

The third error type occurred once, and the reason was that the right opening at node N5 when approaching from N3 was detected to late. Only in the two last scans before getting into the intersection was the right side tunnel detected, and only two consecutive occurrences of a side tunnel is not enough for the localisation to consider it to be valid.

The conclusion of the experiment is that the navigation system is capable of performing autonomous navigation in a

[1]A mission is here defined as the task of moving to a first goal node, turn and then move to a second goal node.

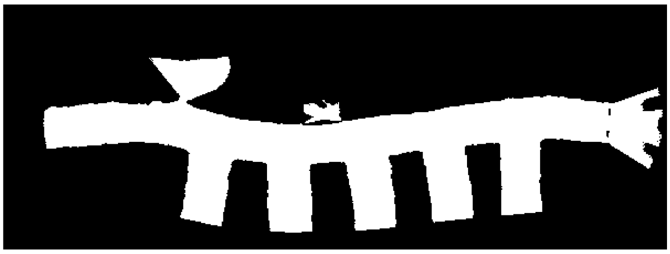

Fig. 7. Gridmap of the tunnel that is traversed in the experiment. Nominal drift width is 11 m.

mine like environment. Out of 88 missions, there was only one failure that is related to a situation that can occur in an underground mine. Most of the errors occurred due to the used robots inability to travel backwards, while an LHD can, and that the robot therefore had to turn around when receiving a new goal. The other error type that was found during the experiment was due to obstacles in the simplified "tunnels", and that the navigation system tried to recover and continue its movement when the obstacles were removed. In a real autonomous LHD application this type of error would not occur due to two reasons. First, the area where the LHD is operating would be sealed of and no humans would be allowed within the area when the machine is active. Secondly, for safety reasons the control system of an autonomous LHD would be designed to stop immediately when detecting an obstacle, and to report the stop to the control room or superior control system.

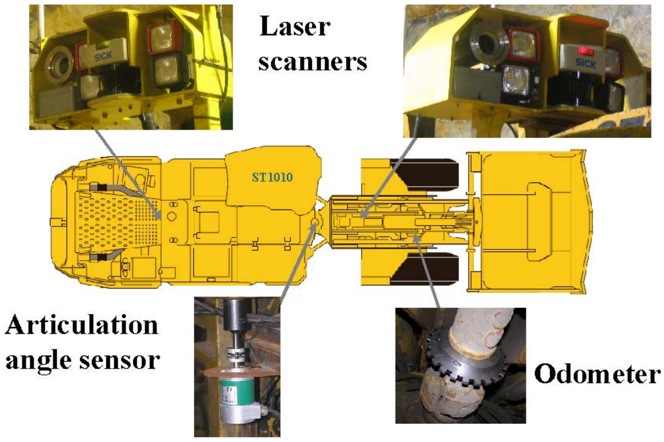


Fig. 8. The sensors added to the original control system

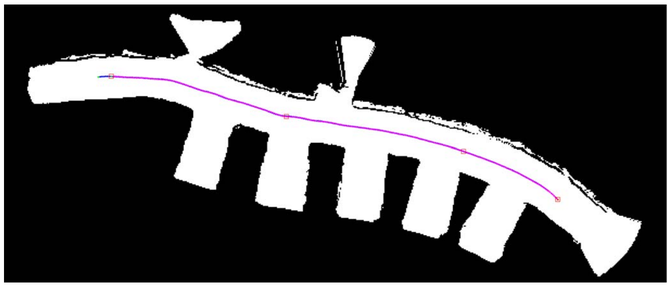

Fig. 9. Gridmap with vehicle path generated from the data recorded during the experiment.

## IV. Evaluation of tunnel following on a real LHD

Currently the navigation system described in this paper has been partly ported to the control system of a ST1010 LHD, manufactured by Atlas Copco. The parts that have been ported are the feature detection algorithms and the LPS. In addition a simple tunnel following behaviour has also been implemented. In this experiment the tunnel following behaviour is used to guide the LHD along a tunnel of a real underground mine. The tunnel that the vehicle is intended to traverse is intersected by several side tunnels, see Figure 7.

### A. Experimental setup

The ST1010C is a one of a kind prototype machine equipped with Atlas Copco's CAN-based Rig Control System (RCS), Figure 1, and is based on an Atlas Copco Wagner ST1010 LHD which has an empty mass of 26.3 tonnes and takes a payload of 10 tonnes.

The standard ST1010 is equipped with direct hydraulics for control of steering, brakes and bucket movement, and analogue interfaces to the transmission and engine, e.g. accelerator pedal and gear selector switch. In the ST1010C, all vehicle functions are interfaced via a CAN-based computerized control system. Since this machine is of an old type and one of a kind, it is not optimal for research on future products. However, when the project started this was the only available Atlas Copco LHD that was already equipped with the RCS. To enable autonomous tramming, the machine was equipped with some additional sensors, Figure 8, and an extra computational module. The main sensors added are: a drive shaft encoder to measure drive length; a hinge angle encoder; and two laser range finders, one for each direction of travel.

In the experiment the performance of the tunnel following behaviour is evaluated by navigating a section of a tunnel with several intersecting tunnels, see Figure 7.

### B. Results

The experiment showed that the tunnel following behaviour is fully capable of guiding the vehicle in a tunnel. Figure 9 shows the path of the vehicle from one of the runs superimposed on a gridmap generated from the data retrieved during the tunnel following. Note that the tunnel in the map generated from the recorded data is curved, something that indicates that the articulation angle sensor was not properly calibrated during the test.

From the figure it is clear that the machine was able to track the center line of the tunnel, and accordingly that the path seems rather smooth.

However, when displaying the curvature of the path, Figure 10, it becomes apparent that the tunnel following behaviour is not completely stable. The oscillations were also visually observable when looking at the vehicle during the run. After a brief investigation of the implementation of the tunnel following behaviour, it became clear why the machine oscillated during tunnel following. The feedback of the articulation angle had in the follow behaviour efficiently been cut off by resetting the variable used to store the articulation angle. This means that the tunnel following was performed with open loop control, something that shows the robustness of the rest of our reactive tunnel following behaviour.

## V. Conclusions

In this paper we have described how an existing framework for autonomous navigation of research robots in indoor environments has been adapted to autonomous tramming of LHD vehicles in underground mines. The presented empirical evaluation of the full control system system show that it is capable of autonomous navigation in tunnel like environments. Moreover, we have also showed on a real LHD in an underground mine that the system is capable of tunnel following. In

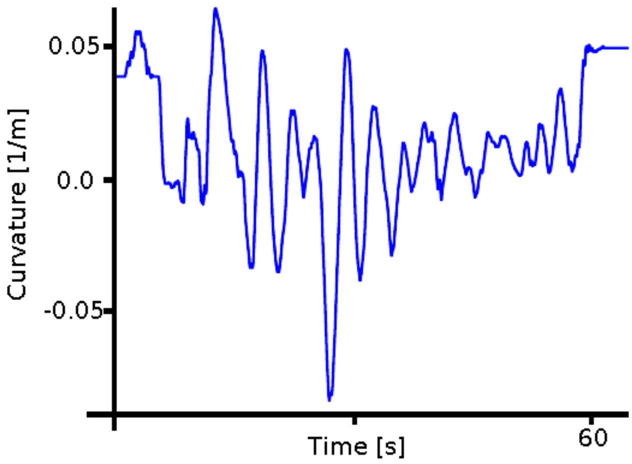


Fig. 10. The plot is showing the curvature of the travelled path from one of the tunnel following runs.

addition, the experiment on the real vehicle showed that the system is extremely robust as it was capable of controlling the articulated vehicle without feedback from the articulation angle sensor.

For the future, it is desirable to port the complete navigation system to the control system of the LHD. Another challenge is to explore the possibilities of semi autonomous tele-operation of mobile robots and in particular LHD vehicles in underground mines using modules from our navigation system.

## REFERENCES


[1] J. Larsson, M. Broxvall, and A. Saffiotti, "A navigation system for automated loaders in underground mines," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, P. Corke and S. Sukkarieh, Eds. Berlin, DE: Springer, 2006, pp. 129–140.

[2] J. Larsson, M. Broxvall, and A. Saffiotti, "Laser based corridor detection for reactive navigation," *Industrial Robot*, vol. 35, no. 1, pp. 69–70, 2008.

[3] J. Larsson, *Licentiate thesis*. Örebro, Sweden: Örebro University, 2006.

[4] G. Eriksson and A. Kitok, "Automatic loading and dumping using vehicle guidance in a Swedish mine," in *International Symposium on Mine Mechanisation and Automation, Colorado*, 1991, pp. 15.33–15.40.

[5] G. Brophey and D. Euler, "The opti-trak system. a system for automating today's lhds and trucks," *CIM Bulletin*, vol. 87, no. 984, pp. 52–57, 1994.

[6] "SALT4 övergripande beskrivning," Product manual, 2000.

[7] E. S. Duff, J. M. Roberts, and P. I. Corke, "Automation of an underground mining vehicle using reactive navigation and opportunistic localization," in *Australasian Conference on Robotics and Automation, Auckland*, 27-29 November 2002, pp. 151–156, http://www.araa.asn.au/acra/.

[8] H. Mäkelä, "Overview of LHD navigation without artificial beacons," *Robotics and Autonomous Systems*, pp. 21–35, 2001.

[9] J. Marshall, T. Barfoot, and J. Larsson, "Autonomous underground tramming for center-articulated vehicles," *To Appear in Journal of Field Robotics*.

[10] A. Saffiotti, K. Konolige, and E. H. Ruspini, "A multivalued-logic approach to integrating planning and control," *Artificial Intelligence*, vol. 76, no. 1-2, pp. 481–526, 1995, online at http://www.aass.oru.se/˜asaffio/.

[11] A. Saffiotti, "Autonomous robot navigation: a fuzzy logic approach," Ph.D. dissertation, Université Libre de Bruxelles, 1998.

[12] P. Buschka, "An investigation of hybrid maps for robot localisation and navigation," Ph.D. dissertation, Örebro University, Örebro, Sweden, 2005.

[13] D. Driankov and A. Saffiotti, Eds., *Fuzzy Logic Techniques for Autonomous Vehicle Navigation*. Berlin, Germany: Springer-Verlag, 2001.

[14] D. Filliat and J. A. Meyer, "Map-based navigation in mobile robots. i. a review of localization strategies," *Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.