

The First **Conference** on
ZalaZONE Related R&I Activities of
Budapest University of
Technology and Economics 2022

31 March 2022



**The First Conference on ZalaZONE Related R&I Activities of
Budapest University of Technology and Economics 2022**

31 March 2022, Budapest, Hungary

**The First Conference on ZalaZONE Related R&I Activities of
Budapest University of Technology and Economics 2022**

31 March 2022, Budapest, Hungary

Mandatory reviewers:

Szilárd Aradi PhD

Krisztián Bán PhD

Ádám Bárdos PhD

Tamás Bécsi PhD

Gábor Bohács PhD

Henrietta Lengyel PhD

Tamás Lovas PhD

Szabolcs Rózsa PhD

András Rövid PhD

Árpád Somogyi PhD

Tamás Tettamanti PhD

Viktor Tihanyi PhD

Árpád Török PhD

Balázs Varga PhD

Zsolt János Viharos PhD

The papers published in this proceeding reflect the views only of the authors. The publisher cannot be held responsible for the validity or use of the information therein contained.

Editors:

Dr. Krisztián Bán, Dr. József Hlinka, Dr. Viktor Tihanyi

Budapest University of Technology and Economics

Faculty of Transportation Engineering and Vehicle Engineering

Department of Automotive Technologies

ISBN 978-963-421-873-9

Cover designed by

Márta Iványosi - Szabó

© BME, Gépjárműtechnológia Tanszék

Budapest, 1111 Műegyetem rkp. 3., Tel.: 463-1615, Fax: 463-3978,

E-mail: gjt.titkarsag@kjk.bme.hu, Web: www.auto.bme.hu

Budapest, 2022.

Contents

<i>Mihály Csonthó, András Rövid</i> A Raw Fusion Based 3D Object Detector for Pedestrian and Vehicle Position Estimation	6
<i>Márton Cserni, András Rövid</i> Survey on Image Based Object Detectors	10
<i>Mátyás Szalai, Viktor Remeli, Viktor Tihanyi</i> Development of 3D Visualization for Displaying Real-time Sensor Information from Multiple Sources	15
<i>Szabolcs Nagy, András Rövid</i> 3D Object Detection in LIDAR Point Cloud Based on Background Subtraction	20
<i>Ádám Domina, Viktor Tihanyi</i> Model Predictive Controller for Path Following Applications.....	24
<i>Zsolt Vincze, András Rövid</i> On-Site Test Measurements at ZalaZONE Automotive Proving Ground for Aiding Various Research Projects	29
<i>Viktor Remeli, Ádám Boronyák, Viktor Tihanyi</i> Teleoperation with a Real-time Digital Twin	34
<i>Tamás Márton Kazár, Zsombor Pethő, Gábor Vida, Árpád Török</i> Simulation of Road Traffic Accidents Related to ADAS Systems in PreScan	39
<i>Dániel Panker, Tamás Tettamanti</i> Automotive Proving Ground HD Map Models	44
<i>Áron Horváth, Tamás Tettamanti</i> Test Automation – From Virtual Scenario Creation to Real-world Testing.....	48
<i>Szilárd Hunor Tóth, Ádám Bárdos, Zsolt János Viharos</i> Initiation and Stabilization of Drifting Motion of a Self-driving Vehicle with a Reinforcement Learning Agent	53
<i>Árpád Fehér, Gábor Kardos, Ádám Szabó, Szilárd Aradi</i> Development of a GNSS Based High Accuracy Measurement System to Support Vehicle Dynamics Testing	58

<i>Tamás Tettamanti, István Varga</i>	
Design of a Novel Road Traffic Control System for ZalaZONE Proving Ground	63
<i>Árpád Fehér, Szilárd Aradi, Tamás Bécsi</i>	
Double Lane Change Path Planning Using Reinforcement Learning with Field Tests..	67
<i>Tamás Lovas, Dániel Baranyai, Árpád Somogyi</i>	
Point Cloud Based Road Surface Modelling and Assessment	71
<i>Szabolcs Rózsa, Ágnes Ács, Bence Turák</i>	
Establishment of a Local GNSS Correction Service for the Localization of Autonomous Vehicles	75
<i>Dániel Pásztor, Péter Ekler, János Levendovszky</i>	
Routing Algorithms for Wireless Sensor Networks in Smart Cities	81
<i>Dávid Sik, Péter Ekler, János Levendovszky</i>	
Predictive Maintenance in Distributed Environments	85
<i>Dávid Sziroczák, Dániel Rohács</i>	
Conflict Management Algorithms Development Using the Automated Framework for Autonomous Vehicles.....	89
<i>László Kocsány, Tibor Gergely Markovits</i>	
Control Design of an Autonomous Moving Platform for Test Tracks	94
<i>Zsolt Farkas</i>	
Testing and Validation of the FRC08 Formula Student Race Car at the ZalaZONE Automotive Test Track	100
<i>Gábor Bohács, András Máté Horváth</i>	
Collaboration Possibilities for Autonomous Industrial Transport Vehicles.....	104

List of Authors

Aradi, Sz., 58, 67
Ács, Á., 75
Baranyai, D., 71
Bárdos, Á., 53
Bécsi, T., 67
Bohács, G., 104
Boronyák, Á., 34
Cserni, M., 10
Csonthó, M., 6
Domina, Á., 24
Ekler, P., 81, 85
Farkas, Zs., 100
Fehér, Á., 58, 67
Horváth, A. M., 104
Horváth, Á., 48
Kardos, G., 58
Kazár, T. M., 39
Kocsány, L., 94
Levendovszky, J., 81, 85
Lovas, T., 71
Markovits, T. G., 94
Nagy, Sz., 20
Panker, D., 44
Pásztor, D., 81
Pethő, Zs., 39
Remeli, V., 15, 34
Rohács, D., 89
Rózsa, Sz., 75
Rövid, A., 6, 10, 20, 29
Sik, D., 85
Somogyi, Á., 71
Szabó, Á., 58
Szalai, M., 15
Szirocák, D., 89
Tettamanti, T., 44, 48, 63
Tihanyi, V., 15, 24, 34
Tóth, Sz. H., 53
Török, Á., 39
Turák, B., 75
Varga, I., 63
Vida, G., 39
Viharos, Zs. J., 53
Vincze, Zs., 29

A Raw Fusion Based 3D Object Detector for Pedestrian and Vehicle Position Estimation

Mihály Csonthó^{1a}, András Rövid¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: csontho.mihaly@kjk.bme.hu*

Abstract

Robust sensing of the environment is an essential and safety-critical part of self-driving vehicles. Most of the algorithms used for this problem employ sensor fusion to increase the reliability of sensing. The most common is the object-level fusion, where separate sensor detections are combined to create an object list. Less common are low-level fusion algorithms that compile their detection list from fused low-level input data. The algorithm presented here uses low-level (raw) fusion and can remarkably improve the detection accuracy and reliability compared to single-camera systems. The proposed detector is aimed for pedestrian detection but is also capable of detecting the position of vehicles or other specific objects even in cases when the number of lidar points representing the object is low.

Keywords: *autonomous driving, environment perception, neural networks, object detection, sensor fusion*

1 Introduction

One of the critical challenges for the uptake of self-driving vehicles is the development of reliable environmental sensing methods. Mono-camera based object detectors can only determine the three-dimensional position using homography or by relying on machine learning based solutions, which is not as accurate as stereo camera or lidar-based systems. Another important aspect to consider here is the speed of detectors, where first of all, the camera-based systems might be emphasized. Sensor fusion based detectors can be an intermediate solution, combining the advantages of both sensors. In object-level sensor fusion, a so-called fused list is generated from a list of objects detected separately in the camera image and the point cloud. Another form of sensor fusion is the low-level fusion, where the different types of data are jointly processed at a lower level of abstraction.

The algorithm DD3D [1], uses a mono camera and a network to predict depth information for the image sequence. MonoRCNN [2] propose a geometry-based distance decomposition that directly predicts 3D bounding boxes from RGB images. CaDDN [3] learns to generate BEV representations from images by projecting image features into 3D space. Then, the 3D object is detected using this representation by an efficient detection network.

US researchers [4] have shown that a modified version of YOLO, written in Darknet, can perform real-time pedestrian detection using MobileNet on the Jetson TX2 platform. Researchers at Chengdu University proposed a pedestrian detection method based on the improved YOLOv3 algorithm [5]. The software system was implemented and verified based on YOLOv3. The experimental results show that the accuracy of the algorithm is improved in pedestrian recognition datasets such as the INRIA pedestrian dataset, and further research directions for pedestrian recognition technology are discussed. Researchers from National Chung Cheng University [6] proposed a model which has introduced a segmentation function that can divide non-overlapping pedestrians in a single image into two sub-images. Using network architecture, they performed multi-resolution adaptive fusion on the output of all images and sub-images to generate the final detection result. The study performed a comprehensive evaluation of several challenging pedestrian detection datasets and finally demonstrated the effectiveness of the proposed model.

Some networks use only the point cloud from a lidar sensor to detect the objects. Building on the PointNet [7] design developed by Qi et al., VoxelNet [8] was one of the first methods to perform authentic end-to-end learning in this area. VoxelNet creates voxels and a so called PointNet is applied to each voxel, followed by a 3D convolutional middle to consolidate the vertical axis, after which a 2D convolutional detection architecture is

applied. While the performance of VoxelNet is robust, inference time is too slow for real-time deployment. Recently, SECOND [9] improved the inference speed of VoxelNet, but 3D convolutions remain a bottleneck.

The bottleneck was solved by PointPillars [10], which is still one of the most computationally efficient architecture (according to the KITTI benchmark site [11]) designed for 3D object detection tasks in lidar point clouds. We can use sensor fusion to overcome the weaknesses of sensors. A proposed method called PointPainting [12] uses low-level fusion, where the lidar points are augmented by semantic information being extracted from images in the form of pixel categories resulting from semantic segmentation of the image.

2 Proposed detector

The built detector uses raw sensor fusion using camera and lidar data. The detector is primarily camera-based and augmented with point cloud data from the lidar to determine an accurate spatial position. The detector is capable of pedestrian and vehicle detection but is not suitable for orientation estimation alone. Accurate camera-lidar calibration is required by the method to work properly.

2.1 Pedestrian detection

The network is suitable for pedestrian detection, as orientation is not essential for pedestrians. The detector can estimate the position based on a few lidar points. Obviously, the larger is the number of points, the higher is the spatial accuracy of detections. The network is based on the YOLO [13] algorithm, trained on the MS COCO dataset. It is capable of detecting 80 different objects, of which 2 objects are currently used.

The detector was tested on a custom dataset even under low light conditions. We have tested the system with several different types of cameras. The detector uses unique logic to associate the points of the lidar point cloud with the detected object in the image, in order to filter out false lidar points. The system uses a 2D box of given size being encapsulated in the 2D bounding box of the object to promote the estimation of the 3D position of the object. This distance helps to identify valid points inside the defined frustum.

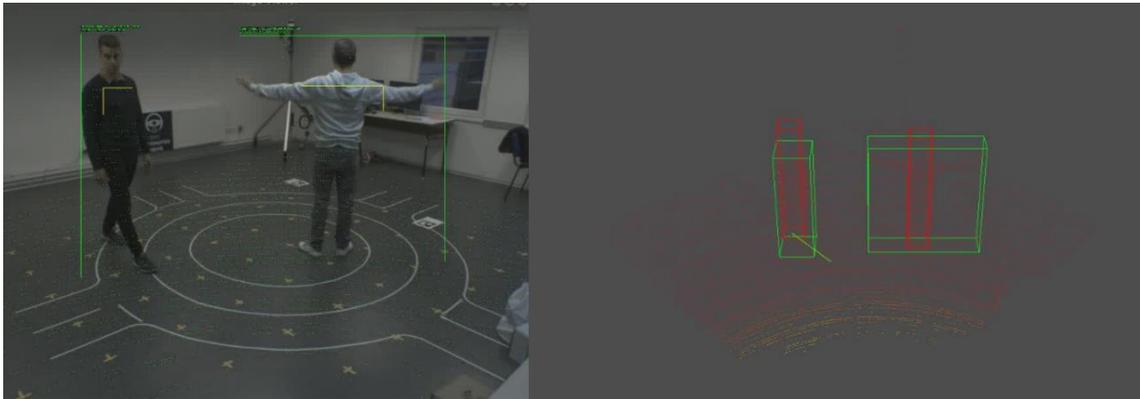


Fig. 1 Predictions of the raw fusion based pedestrian detector

Fig. 1 shows green rectangles, which are the 3D bounding boxes of detected objects of size being in accordance with the pose of the object (in this case human). The red rectangle shows a tracked position represented by a fixed size 3D box.

2.2 Vehicle detection

In addition to detecting pedestrians, the detector can also determine the position of vehicles or other type of objects, however the parameters the parameters of the algorithm must be tuned accordingly. Although the orientation is essential for vehicles, the proposed detector is aimed for position estimation only. Vehicle detections can be seen in Fig. 2.

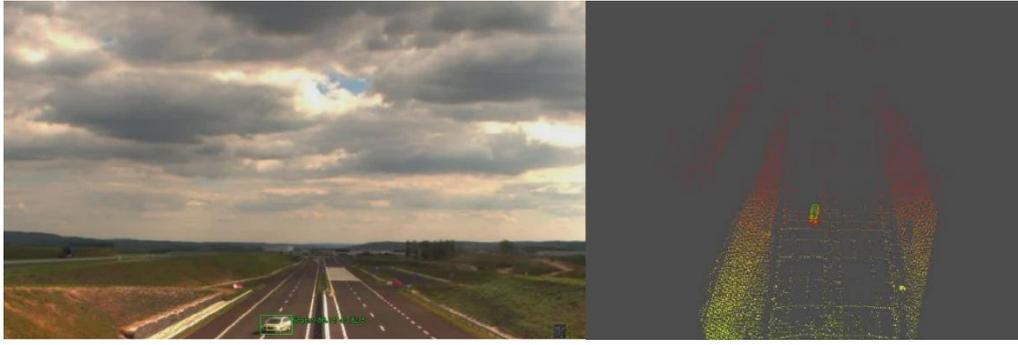


Fig. 2 Predictions of the raw fusion based vehicle detector

The method has been tested on the motorway module of the proving ground. Lidar and camera sensors were mounted on the overpass barrier and calibrated for this measurement. The range of detection depends on the resolution of the camera, the lens as well as on the lidar sensor. If the object recognition algorithm detects a vehicle in the image, theoretically a single 3D point is sufficient to estimate its 3D position. In practice, the maximum range the detector was efficient at was about ~150 meters (by considering the given sensor setup).

3 Calibration and hardware setup

Accurate knowledge of the intrinsics and extrinsics of the camera (wrt. lidar) is essential for the proposed detector. The calibration technique proposed in [14] was used to estimate the intrinsics of the camera, which requires the camera to observe a planar calibration pattern at a few poses.

In addition to the camera calibration, the extrinsics wrt. lidar are estimated. The method in [15] is a fully automatic and convenient extrinsic calibration method for a 3D lidar and a panoramic camera by using a simple printed chessboard as calibration pattern. The method extracts the 3D corners of the chessboard from the lidar point cloud. It formulates a full-scale model of the chessboard and fits it to the segmented 3D points of the chessboard. Finally, the corners of the fitted model are considered as 3D corners of the chessboard. Given the 3D corners and the corresponding projections in the image plane the estimation of extrinsics is converted to a 3D-2D matching problem. The such calculated parameters are then used as initial values and are refined by the Levenberg-Marquardt optimization.

The method was tested with two cameras (Hikvision, PointGrey) and several lidar sensors. The pedestrian detector was tested both indoors and outdoors under different lighting conditions. The system was able to detect pedestrians even with the night vision mode of the Hikvision camera. The system could also be used with a thermal imaging camera in the future, using an image processing algorithm trained on thermal camera specific data.

4 Conclusion

The proposed method is aimed especially for pedestrian detection; however, it might also be useful for estimating the position of other types of objects. The detector's framerate depends mainly on the speed of the image processing algorithm, which in this case exceeds the 20 FPS of lidars. The average speed of the inference function is 30 ms, so the algorithm can run at about 30 FPS.



Fig. 3 Joint usage of the raw fusion algorithm and the orientation estimator network.

The detector can easily be extended with an orientation estimation neural network. We have performed such experiment, as well at the so-called SmartCity module of the ZalaZONE proving ground. The obtained poses of vehicles are shown in Fig. 3. The orientation estimator is based on the YOLO network. Using this combined approach, a single lidar point is sufficient together with the camera image to estimate the position and orientation of the vehicle.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Park, D., Ambrus, R., Guizilini, V., Li, J., and Gaidon, A. (2021). Is Pseudo-Lidar needed for Monocular 3D Object detection?, ArXiv E-Prints, arXiv:2108.06417
- [2] Shi, X., Ye, Q., Chen, X., Chen, C., Chen, Z., and Kim, T.-K. (2021). Geometry-based Distance Decomposition for Monocular 3D Object Detection., ArXiv E-Prints, arXiv:2104.03775
- [3] Reading, C., Harakeh, A., Chae, J., and Waslander, S. L. (2021). Categorical Depth Distribution Network for Monocular 3D Object Detection., ArXiv E-Prints, arXiv:2103.01100
- [4] Jin, Y., Wen, Y. and Liang, J., (2020). Embedded Real-Time Pedestrian Detection System Using YOLO Optimized by LNN., 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1-5, doi: 10.1109/ICECCE49384.2020.9179384.
- [5] Zhang, J., Chen, X., Li, Y., Chen, T. and Mou, L., (2021). Pedestrian detection algorithm based on improved Yolo v3., 2021 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), pp. 180-183, doi: 10.1109/ICPICS52425.2021.9524267.
- [6] Hsu, W. -Y. and Lin, W. -Y. (2021). Adaptive Fusion of Multi-Scale YOLO for Pedestrian Detection., IEEE Access, vol. 9, pp. 110063-110073, doi: 10.1109/ACCESS.2021.3102600.
- [7] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation., ArXiv E-Prints, arXiv:1612.00593
- [8] Zhou, Y., and Tuzel, O. (2017). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection., ArXiv E-Prints, arXiv:1711.06396
- [9] Yan, Y., Mao, Y., and Li, B. (2018). SECOND: Sparsely Embedded Convolutional Detection., Sensors. 18. 3337. 10.3390/s18103337.
- [10] Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., and Beijbom, O. (2019). PointPillars: Fast Encoders for Object Detection from Point Clouds., ArXiv E-Prints, arXiv:1812.05784
- [11] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite., IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.
- [12] Vora, S., Lang, A. H., Helou, B., and Beijbom, O. (2020). PointPainting: Sequential Fusion for 3D Object Detection., Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4604-4612
- [13] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection., ArXiv E-Prints, arXiv:2004.10934
- [14] Bradski, G., (2000). The OpenCV Library. Dr Dobbs J. Softw. Tools
- [15] Kim E.-S., and Park, S.-Y. (2019). Extrinsic calibration of a camera-LIDAR multi sensor system using a planar chessboard., The 11th International Conference on Ubiquitous and Future Networks (ICUFN), 89–91. doi: 10.1109/ICUFN.2019.8806057.

Survey on Image Based Object Detectors

Márton Cserni^{1a}, András Rövid¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: mcserni@edu.bme.hu*

Abstract

Sensor fusion-based detector utilizes camera sensors to solve the problem of recognizing objects and their classifications accurately. This has been proven to increase accuracy compared to single sensor detectors and can significantly help with the 3D tracking of vehicles in the sensor system's area of interest. Even at a distance, where no lidar points are available from the target, a high-resolution camera-based detector can easily detect and classify vehicles. There is a variety of real-time capable 2D object detector convolutional neural networks, some of which are open source. This survey compiles a list of these algorithms, comparing them by precision scores on well-known datasets, and based on experimental evaluation completed on camera images taken on the ZalaZONE test-track to evaluate the distances at which the detectors first perceived the test vehicles. Additionally, inference times are also compared.

Keywords: *2D object detection, autonomous driving, camera, ZalaZONE*

1 Introduction

1.1 Context

Autonomous driving and advanced driver assistant systems (ADAS) can significantly reduce the difficulty of the complicated task of driving, and help eliminate human error, ultimately leading to a significant reduction in road traffic accidents. One such ADAS example would be the Autonomous Emergency Braking System (AEBS). In the event, where a dangerous situation arises, for example a child runs out into the road in front of the vehicle equipped with AEBS, the system shall mitigate or avoid a collision by recognizing the dangerous situation, and applying the brakes autonomously. Recently, AEBS is tested by Euro NCAP as a standard part of the safety rating [1].

A key part of any such system is an informative, perceptive and robust environment perception system. One of the most difficult, and hotly researched task in automotive environmental perception is object detection. Object detection consists of the recognition, classification and localization of relevant objects in the relevant environment of the perception system. Within object detection, 2D and 3D object detection have to be differentiated. 2D object detection outputs a two dimensional bounding box (usually) in images, in the image coordinate system. The bounding box is usually a rectangle given by the top left, and bottom right pixel points, or given by the centre point, and width and height of the box in pixels. 3D object detection is focused on outputting the 3D bounding boxes of objects in the sensor's (eventually a global) coordinate system. The 3D bounding box is given by the centre point, and width, height and length values in metres, and the heading angles (roll, pitch and yaw) of the object.

Object detection is a fundamental problem in computer vision. Camera-based detectors excel in 2D object detection, but lag behind in 3D accuracy compared to other sensors. This paper will focus on the state of the art (SOTA) in image based 2D object detection.

2 Image based object detection

The task of this paper is to compile a list of the SOTA of image-based 2D object detector algorithms in early 2022. The main focus will be on object detection, but an instance segmentation algorithm is also worth mentioning.

Image based object detection is usually approached with deep learning. These, contrarily to classical methods can produce robust and accurate detections regardless of the scenario in the image. Within the category of deep

learning based methods, there exist single-stage and two stage methods. Two stage algorithms usually first create region proposals in the image, in which objects are likely, and input the regions into a second network, which regresses the exact bounding box, and performs classifications. The most notable two-stage algorithms are the R-CNN algorithm family. Two-stage methods usually are more accurate, but slower, making them hard to compute in real-time. Hence, they are not used in autonomous driving applications. Single stage algorithms (such as YOLO, SSD, and RetinaNet) on the other hand can run in real-time, but lack accuracy in certain conditions, such as groups of small objects or irregular objects.

This paper uses the Microsoft COCO (Common Objects in Context) dataset as a reference for evaluating state of the art object detectors [2].

2.1 Two-stage detectors

The most important two-stage detectors are the R-CNNs. R-CNN stands for Regions with Convolutional Neural Network features, as published in 2014 in [3]. The principle of most two-stage detectors, specifically R-CNN is shown in Fig. 1. The network extracts 2000 region proposals from the image (likely containing relevant objects), then for each proposal, a CNN computes image features, which are finally classified by an SVM.

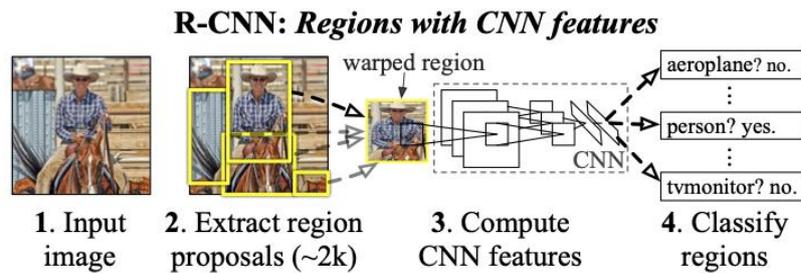


Fig. 1 R-CNN architecture [3]

The problem with R-CNN was that it was slow. Since 2014, the algorithm was improved in Fast-RCNN and Faster-RCNN in [4], and [5] making it much faster. Mask R-CNN was presented in 2017, implementing an instance segmentation (pixel level classification) solution alongside the object detector with minimal overhead [6]. The latest advancement is G-RCNN or Granulated R-CNN, which introduces spatio-temporal information incorporation into Fast and Faster R-CNN which improves detection speed and accuracy on videos, or image sequences [7].

2.2 Single-stage detectors

One-stage detectors focus on real-time applicability. The most popular of these deep learning models are the YOLO algorithms, the SSD, and RetinaNet. The original YOLO paper was published in [8], its working principle can be seen in Fig. 2. Yolo predicts bounding boxes by using a single neural network by dividing the input image into a grid. Each grid cell predicts a number of bounding boxes (anchors) alongside an objectness score (the probability of an object being present in the grid cell). The objects are filtered based on this objectness score, and the final bounding box is regressed as a distance from the anchor box with the highest objectness score.

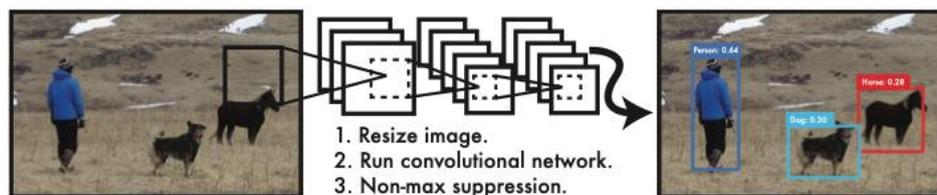


Fig. 2 Yolo principle [8]

Since 2016, YOLO has been updated and improved. The most significant results are Yolov3 [9], Yolov4 [10], and recently YOLOR [11]. Yolov3 improved the detection of small objects by introducing upscaling into the fully convolutional neural network model. Yolov4 presented an optimal architecture for speed and accuracy in object detection. YOLOR (You Only Learn One Representation) is a very recent discovery. The authors found a way to

incorporate explicit and implicit learning into the multi-task model training, achieving increased accuracy and significant detection speed increase (88% over the scaled Yolov4).

The current state of the art in object detection on the MS Coco dataset is the Swin Transformer V2 [12]. This model, alongside the M-RCNN is an object detector and an instance segmentator.

Yolov5 is also worth mentioning [13]. This is an open-source Pytorch implementation of object detector models (based heavily on Yolov4) as implemented by Ultralytics. A simple Pytorch Hub API call is enough to download a variety of models ranging from the Yolov5n to the Yolov5x, which are all pretrained on the MS Coco dataset. The models differ in size, thus inference speed and detection accuracy.

2.3 Metric and Comparison

The most important metrics for object detectors are the inference time, and the accuracy. Inference time depends heavily on hardware and implementation efficiency. As for the accuracy, the most widely accepted metric is the AP or average precision. The latest papers publicise the COCO mAP metric of their detectors. This is explained fully in [14] and on the COCO website [15]. This metric computes the average precision of detectors with variable Intersection over Union threshold value ranging from 0.5 to 0.95 with a step size of 0.05.

Table 1 contains the most relevant detectors and their mAP values. Inference speeds are hard to compare due to implementation differences, however there is more on this topic in the experimental evaluation section.

Table 1 Model Coco mAP values [16]

Name	Coco mAP
Swinv2-G	63.1
YOLOR-D6	57.3
Scaled Yolov4	55.8
Yolov5x (largest Yolov5 model)	55.4
Yolov5n (smallest Yolov5 model)	34.0
Mask-RCNN	46.1
Faster-RCNN	37.4

3 Experimental Evaluation

In this section an experimental evaluation is presented of the online available implementations of some of the state of the art 2D object detectors.

3.1 Scenario

The test scenario was recorded on the ZalaZONE test track, in Zalaegerszeg, Hungary. In the test scenario, four test vehicles, equipped with high precision GPS sensors were stationed approximately 20-50m apart, and asked to perform a Y turn, then drive away from the sensor. The camera used was a Flir PointGrey camera with 2MP image sensor and a 60° HFOV. The relatively low resolution of this camera affects network performance negatively in the case of targets at a longer distance. The detectors were evaluated based on the greatest distance they detected the retreating vehicles from, and their GPU usage and inference times were recorded. The evaluation was performed on a Nvidia RTX 2060 Super with a maximum of 175W power, 7982MiB memory and 7.2 Tflops of computing power.

3.2 Results

Firstly, the most easily accessible Yolov5 models were evaluated. These models are implemented in Pytorch, using the Tensor Cores of the GPU meaning this is a very efficient implementation. The results are surprising, as it turns out that the optimal model is the Yolov5l, which works better in this scenario compared to the larger Yolov5x. Next, an implementation of M-RCNN is evaluated as a comparison. Keep in mind, that M-RCNN precedes the Yolov5 with 4 years. The Gluon model-zoo implementation [17] uses MXNet, and the model inference also runs on the GPU. Table 2 compares the results of the Yolov5 models and the M-Rcnn. Stable distance refers to every frame being detected, and beyond that until furthest distance, the detector is noisy up to Furthest Distance, where the vehicle is lost.

Table 2 The results of the experimental evaluation

Name	Inference time [s]	GPU Power [W]	GPU Memory [MiB]	Stable Distance [m]	Furthest Detection [m]
Yolov5n	0.0084	46	1110	90	99
Yolov5s	0.0086	50	1144	118	124
Yolov5m	0.0129	55	1236	135	149
Yolov5l	0.0153	111	1436	164	223
Yolov5x	0.0234	130	1794	151	218
M-Rcnn	0.65	175	6065	100	120

4 Summary

In this paper, the state of the art of 2D object detection was briefly presented, along with an experimental evaluation conducted in a real-world scenario on ZalaZONE. It can be concluded, that Yolov5l is the presently most applicable implementation, however promising new results recently became available. Future research will include the evaluation of Yolor and Swinv2-G, as their code is also available online.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] European New Car Assessment Programme, "Euro NCAP Rating Review 2018," June 2020. [Online]. Available: <https://cdn.euroncap.com/media/58242/euro-ncap-rating-review-2018-version-12-with-appendices.pdf>. [Accessed 12 02 2022].
- [2] T.-Y. Lin and e. al, "'Microsoft coco: Common objects in context.," European conference on computer vision., 2014.
- [3] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580--587, 2014.
- [4] R. Girshick, "Fast R-CNN," Proceedings of the IEEE international conference on computer vision, pp. 1440-1448, 2015.
- [5] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, no. 28, 2015.
- [6] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," Proceedings of the IEEE international conference on computer vision, pp. 2961-2969, 2017.
- [7] A. Pramanik, S. K. Pal and P. Mitra, "Advances in neural information processing systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 1, pp. 171-181, 2021.
- [8] J. Redmond, "You only look once: Unified, real-time object detection," IEEE conference on computer vision and pattern recognition, pp. 779-788, 2016.
- [9] J. Redmond and A. Fahradi, "Yolov3: An incremental improvement.," arXiv preprint , no. arXiv:1804.02767, 2018.
- [10] A. Bochkovskiy, W. Chien-Yao and M. L. Hong-Yuan, "Yolov4: Optimal speed and accuracy of object detection.," arXiv preprint, no. arXiv:2004.10934, 2020.
- [11] C.-Y. Wang, I.-H. Yeh and L. M. Hong-Yuan, "You only learn one representation: Unified network for multiple tasks.," arXiv preprint , no. arXiv:2105.04206, 2021.
- [12] Z. Liu and e. al, "Swin Transformer V2: Scaling Up Capacity and Resolution," arXiv preprint, no. arXiv:2111.09883., 2021.
- [13] Ultralytics, "github.com," 2022. [Online]. Available: <https://github.com/ultralytics/yolov5>. [Accessed 13 02 2022].

- [14] J. Hui, "medium.com," 2018. [Online]. Available: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. [Accessed 13 02 2022].
- [15] "Coco Detection Leaderboard," [Online]. Available: <https://cocodataset.org/#detection-leaderboard>. [Accessed 13 03 2022].
- [16] "Papers With Code," [Online]. Available: <https://paperswithcode.com/sota/object-detection-on-coco>. [Accessed 13 02 2022].
- [17] Gluon, "Model zoo," [Online]. Available: https://cv.gluon.ai/model_zoo/segmentation.html. [Accessed 13 02 2022].

Development of 3D Visualization for Displaying Real-time Sensor Information from Multiple Sources

Mátyás Szalai^{1a}, Viktor Remeli¹, Viktor Tihanyi¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: szalai.matyas@kjk.bme.hu*

Abstract

For modern scientific results, it is increasingly challenging to present them in a form that the general public can understand. This is particularly true in the field of collaborative intelligent transportation services and connected vehicles, where many different platforms and sensors are operating simultaneously. It is important for users to build up the necessary trust in the new technology, which is why the presentation of the emergence and interaction of collective, platform- and sensor-level environment models is an important area. In this paper, a visualization environment that can present the different locally detected and centrally fused objects in real time are presented through an example of multiple infrastructure sensors. By statically and dynamically representing the digital twin, we can create a world where human users can see their environment through the eyes of individual vehicles, infrastructure stations, or the whole system. This, combined with Mixed Reality technology, can produce a result that feels real to the human eye and facilitates intuitive understanding of and trust towards the system.

Keywords: *Real-time, Sensor Detections, Sensoris, Unity3D, Visualization*

1 Introduction

Developments surrounding road vehicles could bring big changes to the transportation systems. In addition to the driving assistance systems that are now in almost all new cars, there is increasing talk of self-driving vehicles as a soon-to-be available option. Self-driving cars are surrounded by many questions from the general public. There are still several directions in which different industrial players are trying to reach the final goal, with different solutions for sensing, data processing, decision making and final implementation. Legal issues remain to be worked out, who is responsible for decision in self-driving vehicles, especially in case of accidents. Taken together, these issues continue to create a lack of trust in the average user [1], which is particularly true for the area of vehicle sensing. Knowing human vision, it is difficult to imagine how a vehicle sees, which is why it is important not only to be able to use machine vision, but also to present its operation in an appropriate form.

In the case of self-driving vehicles, people always talk about the vehicle's own perception system, but in intelligent transportation systems, the infrastructure itself can also play an active role. Multiple sensors mounted on roadside pillars provide greater coverage. By collecting their data on a single central server, a more complex digital twin is created [2]. Vehicles plugged into this digital twin can see much more information, with only a proper communication system (V2X). Referring to human vision, it is difficult for us to imagine seeing something from more than one direction, and the benefits of such a system are therefore harder to bring to the public's attention.

In this paper, we present a visualization system that can display sensor data from independent multiple sources, all positioned appropriately on static HD maps, creating an online digital twin. The platform created ensures that different systems can be managed in a single environment. Through a central system and Sensoris-based communication, the visualization module operates independently of the sensing system. Thus, we have come up with a solution that allows people to see the detections of the sensor systems on a large display or through AR headsets, projecting them into a digital twin of the real environment. This system can be used to illustrate the greater reliability of multi-sensor systems, the greater avoidability of outliers, and thus the robustness of the sensing. In the following sections, the interface between the data collecting central server and the Unity 3D

visualization tool is presented, and the logic of the visualization and the content of the incoming data also will be explained. Finally, the complete visualization system presented in operation, through a real-time application.

2 Communication Interface for reading and sorting Sensoris data

To create a solution that can be used universally, it is necessary that the information have to be available in some standardized form. For this purpose, we use the SENSORIS standard [3], which provides an appropriate descriptive language for objects already detected by sensors and for the senders of detections. Messages from different sources are available on a central server, which can be accessed by any actor through a communication client module. In our case, the visualization only reads from the server, but additional information also can be sent back via the client module. The following subsections describe the C# interface that communicates with the client module, and how Sensoris data is transformed into a format that can be handled by Unity.

2.1 Communication Interface

Unity provides the possibility to use so called Managed plug-ins [4], which require compiling dll files. By choosing this solution, we can create a reusable C# interface that can also manage the packages used by Sensoris. The Google Protobuf and Grpc packages are required to manage the Sensoris messages. These can be added to the solution in the form of so-called NuGets. While for Visual Studio the NuGet manager is a built-in module, for Unity we can add the NuGet manager using a custom-package called NuGetForUnity [5].

Through the interface built against netstandard 2.0, we can retrieve the real-time messages of the last fusion step. A fusion step contains all the incoming messages that have been used to determine the output of the fusion. In this case, only the most recent fusion step can be received. This guarantees that the visualization is running in real-time, the only delay is the delay of the entire system.

The `IAsyncStreamReader` class of `Grpc.Core` [6] is used to access the stream. The method call has two parameters, the name of the monitored topic and a `bool` parameter to allow or disallow receiving older messages. During real-time playback, the party using the interface receives a list of Sensoris datamessages. This list reflects the latest possible state after each call. However, this also means that, depending on the frequencies of the receiving and transmitting, it is possible that a step may be read more than once, or steps may be missed between two calls. To deal with this, the frequency of visualization recommended to be higher than the frequency of sensor fusion. The messages received through the interface therefore contain a list of elements of type `Sensoris DataMessage`. To manage their contents, additional operations are required, now within Unity.

2.2 Classification of Sensoris data into Unity classes

To simplify the handling of the information, the datamessages for a given fusion step are mapped one by one, according to a unique class for parsing Sensoris messages. This `SensorisParser` class is responsible for determining the sender of a given datamessage, and for sorting the detections associated by their type. The `Sensoris` specification contains several static and dynamic detection categories, the system can currently handle pedestrian and vehicle types. Objects are created for both the sender and the detection, which are populated with different parameters based on the `Sensoris` description. In the case of the sender, the `Origin` class contains the sender's ID, position, and rotation. In the case of detections, a `Detection`-type object is created, which contains the detection's own identifier, its position and rotation, as well as its 3D size, the probability of its existence and the identifier of its sender.

For the determination of the positions and rotations, it is important to find some reference point between the real and virtual worlds to create a digital twin. In our case this is a pre-selected null point, which we use as the origin of the whole system. The local position, interpreted in Unity's own coordinate system, is given by the UTM position relative to this null point. In addition, it was important to convert the incoming data expressed in millimeters to meter format and to handle the differences between the coordinate systems. Unity's own coordinate system differs from the coordinate system used in the automotive industry, while Unity's coordinate system is left-handed, and Y-Up type, the coordinate system of the automotive industry is right-handed, and Z-Up type [7]. This means that in addition to the offset and reverse rotation directions, the y and z axes are also swapped. As in the case of positions, in all other cases the conversion of the data to metric must be done so that the data has the correct type for Unity.

3 The logic of the visualization

The representation of the objects detected by the sensors should be interpreted as a function of time, where a packet of messages received from the server represents a specific fusion time snapshot. The received message packets, which are sorted into `DataMessage` type lists, can display different states. Since message reads are assigned to fusion steps, the last message packet is always associated with the fusion, and only one packet is guaranteed to be read. For the other sensor stations, the structure of the `Sensoris` topic, called “fusion-utm” guarantee that maximum one packet belongs to each sender. Depending on the frequency of the fusion and each sender, the `Sensoris` topic will always store only the most recent message for each sender, and if the frequency of a sender is lower, there may be no message part for that sender. The possible contents of messages as a function of time are illustrated in Fig. 1.

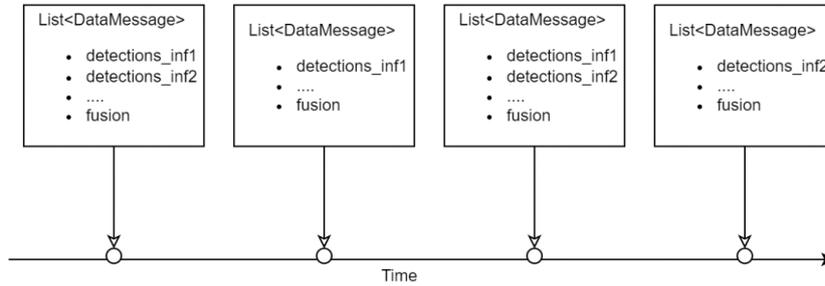


Fig. 1 Content of the DataMessage Lists over time

The visualization is always triggered by new incoming information, the visualized detections are moved when new information about their state is received. Each list is processed `DataMessage` by `DataMessage`, and based on their content, new detections may appear, old ones may disappear, or existing ones may be moved to another position. Fig. 2 illustrates how each data packet contains detections and the corresponding sender, called `Origin`.

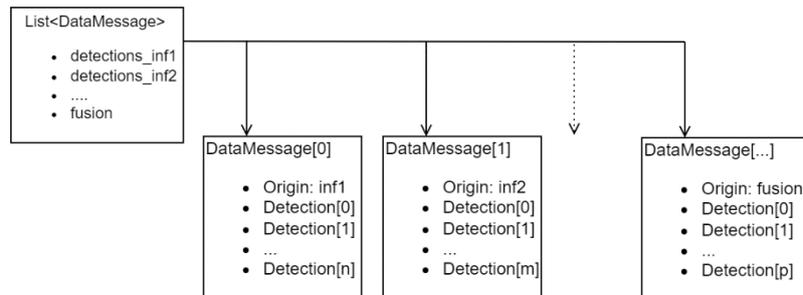


Fig. 2 DataMessage list's elements and their content by sender

A `DataMessage` can be divided into two main parts in terms of detections. The message always contains information about its sender and a list of objects detected by the sender, with their parameters. They are processed as described in chapter 2.2. Once the data is available in the Unity-defined classes, the next step is its visualization. The first visualization step is a simple task since each object must be represented in the virtual space according to its content. The next visualization step must act based on the objects that already exist. To ensure that only detected objects that exist at a given moment are displayed, all previous objects belonging to the sender of a given `DataMessage` are firstly deleted from the `Scene`, and then new detections are displayed according to the new message. This step also solves the problem if a particular sender can serve data at a lower frequency than the fusion. In this case, the detections associated with that sender will remain on the screen until a new message is received from that source. However, this solution raises the question of what happens to the detections if there are no more messages from that source, i.e., the sender has left the system for some reason. This has been solved by automatically deleting all detections 500 milliseconds after they have been displayed if they have not been deleted by another call.

An important area of visualization is to create virtual reality as a digital twin of reality. HD maps are becoming more and more familiar [8] and can easily be used to create a static digital twin of the real environment. These maps include their position in the Earth coordinate system, which we use in UTM format. The various sensor

stations also know their own positions, and in turn the positions of all the objects they detect. The resulting UTM positions can then be used to place the detections exactly where the real objects are in the real world. However, we cannot store the whole world in Unity, so only a small part of it is implemented. On this map, we choose an origin, whose exact position are known, and then all the other objects are displayed relative to it. Unity provides a tool, using local position and rotation parameters, that always returns the position of an object relative to its parent. Therefore, both senders and detections are defined as children under the HD map, and their position is given by the local parameters.

To summarize, the visualization receives incoming data packets bound to fusion steps and then handles them separately per source. After deleting the previous messages for a given source, new detections are generated according to the new content. Meanwhile, the source, i.e., the Origin object, is also represented in its own position, represented by a sphere in the case of a sensor station, and by a 3D model of the source in the case of a vehicle.

4 Real-time application of the visualization system

The implementation of a system's logic is complete when it works in reality. The system developed has been demonstrated on several occasions. In this chapter, this system will be presented.

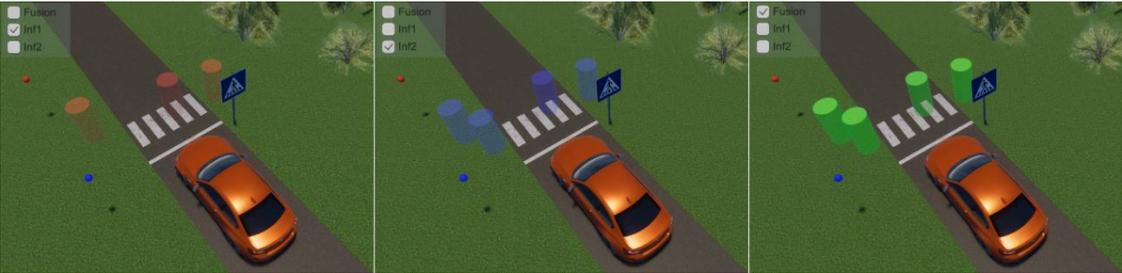


Fig. 3 Visualized detections from the two stations and the fusion

In the tested system, two real sensor stations sent their detections to the central system. These detections, as well as those generated by the fusion, are displayed in the visualization. In addition, a simulated vehicle was added to the scenario, which stopped when pedestrians entered the crosswalk. In Fig. 3, the different sensor stations are marked in different colors. Infrastructure sensor station 1 is marked in red, infrastructure sensor station 2 in blue, while detections from fusion are marked in green. It can also be seen that small spheres in the images indicate the exact location of the stations. It can also be observed that in the image for station 1 only 3 detection cylinders are visible, which is due to the occlusion. However, with the two stations, this object is also visible, it is included in the fusion detections that are considered to be most robust and reliable.



Fig. 4 The demonstrated system and the AR headset with the mixed reality views

The system presented was an indoor scenario with a single road and crossing. However, this small system is sufficient to demonstrate how the system works. Using Augmented Reality technology, the viewer entering the scenario can become a part of the virtual world. Using the Varjo XR-1 headset, the visitor can see everything just

as in reality, combined with virtually generated and projected objects to create mixed reality. Fig. 4 shows a part of the demonstrated system, with one sensor station mounted on the left, a visualization of the virtual world on the large displays, and a tester with AR glasses. The images on the right try to show mixed reality, the image above shows a detection and a virtual vehicle during testing in a laboratory room, and the image below shows the virtual vehicle driving onto a virtual road, exiting the reality.

5 Conclusion

In summary, with the developed solution, it is possible to visualize the already processed data of sensors. The advantage of the system is that it can be easily used in conjunction with augmented reality technology and does not depend on any licensed software or drivers. The client module makes the visualization completely independent of the sensors. The solution used allows detections from multiple sources to be managed in a single environment. Just as the detection of pedestrians has been demonstrated, the visualization of detected vehicles is also implemented, with multiple tasks from the 3D size and orientation point of view. However, the biggest advantage of the system is that it helps the general public understand how collaborative environment perception systems

work, thanks to augmented reality. A possible future development is to extend the list of visualizable types of detections. An important task is to optimize the system and ensure that it can be used in a more compact way. We would like to develop a system that can be deployed easily and quickly at all the ZalaZONE proving ground sites [9], without the need for engineering work to start a demonstration.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] C. Olaverri-Monreal, "Promoting trust in self-driving vehicles," *Nat. Electron.*, vol. 3, no. 6, pp. 292–294, Jun. 2020, doi: 10.1038/s41928-020-0434-8.
- [2] Zs. Szalay, V. Tihanyi, A. Rövid, V. Remeli, Zs. Vincze, M. Csonthó, Zs. Pethó, M. Szalai, B. Varga, A. Khalil, "Towards Cooperative Perception Services for ITS: Digital Twin in the Automotive Edge Cloud," *Energies*, vol. 14, no. 18, p. 5930, 2021, doi: 10.3390/en14185930.
- [3] "SENSORIS - SENSOR INTERFACE SPECIFICATION." <https://sensoris.org> (accessed Oct. 15, 2021).
- [4] "Managed plug-ins," Unity Documentation. <https://docs.unity3d.com/Manual/UsingDLL.html> (accessed Sep. 20, 2021).
- [5] "GlitchEnzo/NuGetForUnity," GitHub. <https://github.com/GlitchEnzo/NuGetForUnity> (accessed Aug. 23, 2021).
- [6] "Grpc.Core," NuGet.org. <https://www.nuget.org/packages/Grpc.Core> (accessed Sep. 16, 2021).
- [7] "A Guide to Unity's Coordinate System (With Practical Examples)," *techarthub*. <https://www.techarthub.com/a-guide-to-unitys-coordinate-system-with-practical-examples/> (accessed Jan. 28, 2022).
- [8] L. Bao, Q. Wang, and Y. Jiang, "Review of Digital twin for intelligent transportation system," in 2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT), Lanzhou, China, Oct. 2021, pp. 309–315. doi: 10.1109/ICEERT53919.2021.00064.
- [9] Zs. Szalay, Z. Hamar, and P. Simon, "A Multi-layer Autonomous Vehicle and Simulation Validation Ecosystem Axis: ZalaZONE," in *Intelligent Autonomous Systems 15*, vol. 867, M. Strand, R. Dillmann, E. Menegatti, and S. Ghidoni, Eds. Cham: Springer International Publishing, 2019, pp. 954–963. doi: 10.1007/978-3-030-01370-7_74.

3D Object Detection in LIDAR Point Cloud Based on Background Subtraction

Szabolcs Nagy^{1a}, András Rövid¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: nagy.szabolcs@kjk.bme.hu*

Abstract

Autonomous vehicles have a key role in transportation systems of the future, but there are still many difficulties to overcome. Nowadays one of the most critical problems in autonomous driving is the precise and robust detection of traffic participants. This paper presents a LIDAR-based 3D object detection method. The algorithm uses HD Map to subtract the static background points from the LIDAR point cloud. The remaining points are grouped by clustering, then 3D boxes are fitted to the clusters. The object detection method presented in this paper was tested on real sensor data collected by a solid-state LIDAR on the highway module of the ZalaZONE proving ground. The results showed that the developed algorithm performs as intended in a highway scenario, detecting vehicles even more than 100 meters away from the sensor by a framerate of ~20FPS.

Keywords: *background subtraction, HD map, LIDAR point cloud, object detection*

1 Introduction

One of the greatest challenges for self-driving vehicles is environmental perception, because autonomous vehicles rely on information about their surroundings acquired from perception systems. Thus, it is crucial to detect the presence of traffic participants like vehicles, pedestrians, and other elements. LIDAR-based perception systems have been introduced due to the need for reliable and accurate measurement capabilities [1]. The LIDAR is an active sensor which operates by emitting laser and measuring the time for the reflected light to return. The range is estimated based on the elapsed time between the transmitted and received signals, resulting in a 3D point cloud representing the surrounding environment. Many high-level perception systems use LIDARs to complement the lack of depth information in 2D image data captured by a digital camera [2].

HD maps are highly accurate maps used for autonomous driving purposes, at centimeter-level. High-definition maps provide information about the surrounding environment, including map elements as road shape, markings, traffic signs while maintaining high accuracy, thus high-definition maps are becoming a key technology for autonomous driving systems, although they are typically employed for motion planning applications [3]. However, there are already object detection solutions [4] that extract geometric and semantic features from the HD maps to improve performance and robustness.

Lidar-based environment perception algorithms have gone through increasing development in recent years. Based on the approach, the most object detection algorithms can be divided into two main groups: classic machine vision-based methods such as [5] using occupancy grid for object segmentation and techniques based on machine learning as PointPillars [6] adopting 2D convolutional layers to learn point cloud features in order to generate 3D bounding boxes for different object classes.

The aim of this paper is to present a traditional object detection method that only uses 3D point cloud from LIDAR as sensor data and benefits from the information contained in a high-definition map. The algorithm uses HD Map to subtract the background points from the LIDAR point cloud, then the remaining points are projected to the road surface. The projected point cloud is grouped into clusters, then to the clustered points 3D boxes are fitted by constructing convex hulls and minimum-area rectangle.

The method has been tested on real sensor data collected by a solid-state LIDAR installed as infrastructure sensor on the highway module of the ZalaZONE proving ground, which HD map is also available. Another

important consideration regarding the performance of the proposed object detector was its applicability in real time applications.

2 Methodology

2.1 Calibration

The measurements were taken at the highway platform of the ZalaZONE proving ground, where the LIDAR was installed as infrastructure sensor in the middle of a bridge over the highway, facing the lanes as shown by Fig. 1.



Fig. 1 High-definition map of the highway platform (left). The LIDAR was installed in the middle facing the lanes

Since only one side of the highway was used for the measurements, the not relevant parts of the HD map were removed manually, which resulted a 3D point cloud representing only the lanes being used. In order to reduce the complexity of the approach, the remaining point cloud was uniformly subsampled, followed by a transformation to a reference coordinate system which in our case was the UTM (Universal Transverse Mercator) frame. For estimating the rotation and translation parameters between the two coordinate systems the iterative closest point (ICP) algorithm [7] was applied.

2.2 Segmentation

Since the highway stretch used for the measurements can be approximated well as a plane, thus the equation of the plane is then calculated by using the least-squares method to solve the over-determined linear matrix equation. The segmentation task began with the removal of the LIDAR points that are on either side of the road. To achieve that, the point cloud was aligned with the XY plane of the UTM coordinate system. By projecting the transformed points to the XY-plane, we reduced the problem to 2D.

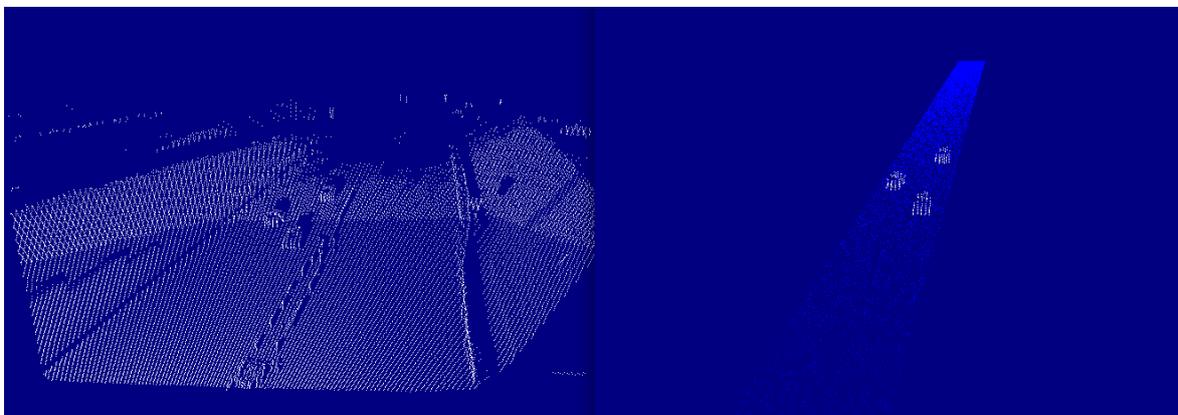


Fig. 2 Original LIDAR point cloud (left) and the segmented point cloud with the plane from the HD map approximating the highway stretch (right)

The remaining points were located either below or above the road surface. In the following step the perpendicular distances - of all kept 3D points - from the approximated plane were calculated. Then, based on the

calculated distances, the points closer than a lower threshold and farther than an upper threshold were omitted. As Fig. 2. shows, the remaining point cloud contains only points that are reflected from objects on the highway stretch.

2.3 Fitting bounding boxes

To fit 3D bounding boxes to the objects, the point cloud is first clustered. Before the clustering the 3D point cloud was arranged into a k-d tree representation, then by using the Euclidean Cluster Extraction method of the Point Cloud Library [8] the clusters were determined, each representing an object. By setting the minimum cluster size to 3 points, the outliers were filtered out. The next step was to fit 3D bounding boxes to the clusters of points. For each cluster, a 3D box is fitted based on the minimum area rectangle (MAR), containing points that were projected to the plane representing the highway section. This was achieved by determining the convex hull of the projected points, thus one of the edges of the MAR will coincide with the corresponding edge of the convex hull. Then for each edge of the convex hull, the corresponding minimum area rectangle was determined. Among all the possible MARs, the rectangle with the smallest area is selected as the bounding box of the 2D point cloud as shown by Fig. 3. For each cluster, the point having the largest distance to the plane of the highway stretch will be considered as the height of the 3D bounding box.

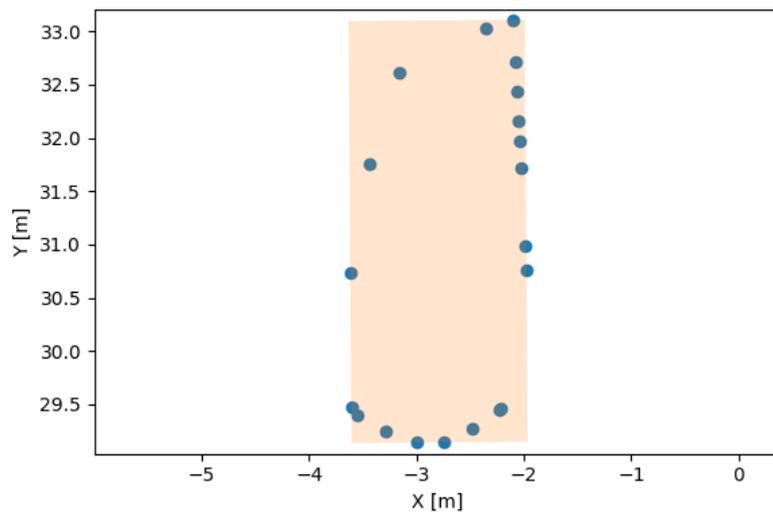


Fig. 3 The minimum area rectangle fitted to the convex hull of a projected point cloud

At longer ranges, where only few points are reflected from the objects, the shape and size of the bounding box fitted to the detected object may vary significantly. In order to reduce this effect, points from previous k frames (corresponding to the same object) have been aggregated.

3 Results

The 3D object detection algorithm presented in this paper was tested on real sensor data collected by a solid-state LIDAR on the highway module of the ZalaZONE proving ground. Several scenarios were examined with multiple vehicles driving at different speeds. As shown by Fig. 4, the algorithm can detect the moving vehicles as intended, while also maintaining the framerate of ~ 20 FPS. The first detections occurred at the distance of around 150 meters, but the size and orientation of the bounding box fitted to the detected object depends strongly on the number of points used, thus the presented method performs better in close range. Setting k to a small value (e.g. $k=3$) improved the bounding box fitting remarkably, on the other hand when taking too many frames into account the heading estimation degrades when the object changes direction. As future work, tracker will be incorporated into the algorithm in order to get better estimations on the orientation and position of the 3D bounding box. In addition, the number of points the vehicle is represented with, maybe used to emphasize the importance of clusters.



Fig. 4 The 3D bounding boxes in the LIDAR point cloud. Vehicles were moving on a sine-trajectory

4 Conclusion

The presented object detection algorithm can create 3D bounding boxes in LIDAR point cloud data by relying on high-definition maps as backbone for background subtraction. Although most object detection algorithms operate on a deep learning basis, this approach only uses classical methods. DL based solutions generally depend on supervised learning requiring a huge amount of labeled data. The trained network will perform as expected only when the setup of sensors doesn't change (height, angle). For example, LIDARs generate different patterns of objects from different positions, thus by changing the mounting height of the sensor may strongly affect the detection performance. This traditional solution can be applied for any LIDAR sensor setup, especially in a highway environment.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Y. Li and J. Ibanez-Guzman (2020). Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems, in *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50-61, doi: 10.1109/MSP.2020.2973615.
- [2] Tihanyi V, et al. (2021). Motorway Measurement Campaign to Support R&D Activities in the Field of Automated Driving Technologies. *Sensors*. 21(6):2169. <https://doi.org/10.3390/s21062169>
- [3] Seif, Heiko & Hu, Xiaolong. (2016). Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry. *Engineering*. 2. 159-162. 10.1016/J.ENG.2016.02.010.
- [4] Yang, B., Liang, M., & Urtasun, R. (2018). HDNET: Exploiting HD Maps for 3D Object Detection. *ArXiv*, abs/2012.11704.
- [5] Himmelsbach, M & Müller, A & Luettel, Thorsten & Wuensche, Hans-Joachim. (2008). LIDAR-based 3D object perception.
- [6] Lang, Alex & Vora, Sourabh & Caesar, Holger & Zhou, Lubing & Yang, Jiong & Beijbom, Oscar. (2019). PointPillars: Fast Encoders for Object Detection From Point Clouds. 12689-12697. 10.1109/CVPR.2019.01298.
- [7] Bouaziz, S., Tagliasacchi, A., Pauly, M., (2013). Sparse iterative closest point. In *Computer graphics forum*, vol. 32, no. 5, pp. 113-123. Oxford, UK: Blackwell Publishing Ltd.
- [8] R. B. Rusu and S. Cousins (2011). 3D is here: Point Cloud Library (PCL), 2011 IEEE International Conference on Robotics and Automation, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.

Model Predictive Controller for Path Following Applications

Ádám Domina^{1a}, Viktor Tihanyi¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: domina.adam@kjk.bme.hu*

Abstract

In this article, a model predictive controller (MPC) for automated vehicle path following is presented. The MPC calculates the optimal steering command based on the prediction of the future states of the vehicle and the reference given for the controller. The aim is to minimize the difference between the predicted and the reference states, which is ensured by the optimal control input. The MPC control technique applies a vehicle model for state prediction. In this paper, a bicycle model is used for state prediction, which model is updated at every time step formalizing an LPV-MPC structure. The controller aims to minimize the lateral and angular deviation of the vehicle from a fixed path. A reference path is built, on which the controller is tested, showing a good performance.

Keywords: *automated vehicle, MPC, vehicle model*

1 Introduction

The advanced driver-assistance systems, collision avoidance systems, and the different automated vehicle functions are getting more popular in recent years. The improvement of automated vehicles allows the potential to reduce pollutant emissions, the number of accidents, improve road safety and ensure safer transport. In the hierarchical software structure of an automated vehicle, the motion control layer is responsible for the lateral and longitudinal motion control of the vehicle. In this article, the authors mainly focus on lateral control, which is realized by the steering system in an automated vehicle. The longitudinal dynamics is controlled separately by a PI cruise controller.

In this paper, a MPC is applied for following a path from the numerous existing path following solutions presented in [1], [2], and [3]. The authors decided to apply the MPC technique because this controller includes both features required for accurate path tracking, are 1) the dynamics of the vehicle is applied for the calculation of the necessary steering angle as a control input and 2) the controller has information about the reference path ahead of the vehicle. Furthermore, MPC can handle constraints well, in this case, the constraints are applied as limits of the actuator intervention, as maximal steering angle values.

The controller is tested on a predefined path, using different vehicle speeds, the tests proved the practicality of the proposed controller, the performance is good, the vehicle can follow the path with small lateral and angle errors. The organization of this paper is as follows. Section II describes the applied vehicle models, Section III presents the MPC structure, Section IV considers the results, and finally, the results and concluding remarks are summarized in Section V.

2 Vehicle modeling

Two different vehicle models were applied in this article one for testing the performance of the controller and another for state prediction by the MPC.

2.1 Vehicle model for testing the controller

A three-state dynamic bicycle model [4] is used for testing the controller in a simulation environment. The model considers solely the planar dynamics of the vehicle, the pitching and the rolling dynamics are neglected, because these do not have a significant effect on our investigation. The states of the vehicle model are the yaw-

rate of the vehicle, the longitudinal velocity, and the sideslip angle at the center of gravity (C.G.). The derivative of the states is calculated by (1), (2), and (3).

$$\dot{r} = \frac{aF_{yF} - bF_{yR}}{I_z} \quad (1)$$

$$\dot{V}_x = \frac{F_{xR} - F_{yF} \sin(\delta)}{m} + rV_x \beta \quad (2)$$

$$\dot{\beta} = \frac{F_{yF} + F_{yR}}{mV_x} - r \quad (3)$$

where r is the yaw-rate, a and b are the distance from C.G. to front and rear axle, respectively, I_z is the moment of inertia around axis z , F_{yF} and F_{yR} are the lateral forces at the front and at the rear wheels, respectively, F_{xR} is the traction force at the rear wheel, δ is the steering angle, m is the mass of the vehicle, V_x and V_y are the longitudinal and the lateral velocity of the vehicle in the ego frame, respectively, and β is the sideslip angle of the vehicle. The bicycle model and the notations are shown in Fig. 1, where α_F and α_R are the sideslip angles at the front and the rear axles, respectively.

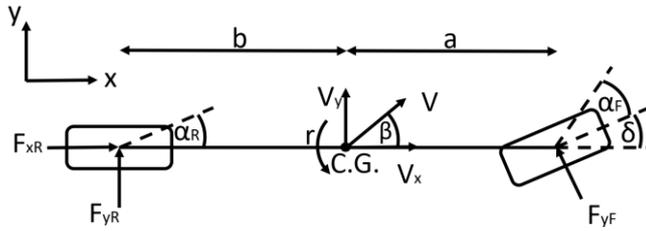


Fig. 1 Three-state bicycle model

The lateral dynamics of the tires is modeled by a Brush tire model (4), (5) which is enhanced with a ζ derating factor (6) at the rear wheels for considering the reduction of the maximal transferable lateral force. At the front wheel, the value of ζ is fixed as $\zeta=1$ since there is solely lateral force applied, assuming a rear-wheel-drive vehicle.

$$F_y = \begin{cases} -C_\alpha \tan(\alpha) + \frac{C_\alpha^2}{3\zeta\mu F_z} |\tan(\alpha)| \tan(\alpha) - \frac{C_\alpha^3}{27\zeta^2\mu^2 F_z^2} \tan^3(\alpha), & |\alpha| \leq \alpha_{sl} \\ -\zeta\mu F_z \text{sgn}(\alpha), & |\alpha| > \alpha_{sl} \end{cases} \quad (4)$$

$$\alpha_{sl} = \frac{3\zeta\mu F_z}{C_\alpha} \quad (5)$$

$$\zeta = \sqrt{\frac{(\mu_R F_{zR})^2 - F_x^2}{\mu_R F_{zR}}} \quad (6)$$

where C_α is the cornering stiffness of the tires, α is the sideslip angle, μ is the friction coefficient, F_z is the normal load on the wheels. The sideslip angles are calculated by (7) and (8).

$$\alpha_F = \arctan \frac{V_y + ar}{V_x} - \delta \approx \arctan \left(\beta + \frac{a}{V_x} r \right) - \delta \quad (7)$$

$$\alpha_R = \arctan \frac{V_y - br}{V_x} \approx \arctan \left(\beta - \frac{b}{V_x} r \right) \quad (8)$$

2.2 Vehicle model for state prediction

While the MPC is a discrete-time controller structure, a continuous-time state-space representation of the vehicle is formalized and transformed to a discrete-time model. The state-space representation of the vehicle is calculated at every time step based on the current state of the vehicle. A lateral dynamic bicycle model (9) is applied for state prediction which is presented in [5], the state vector is $x = [e_{lat} \ \dot{e}_{lat} \ e_{ang} \ \dot{e}_{ang}]^T$, where e_{lat} is the lateral error, e_{ang} is the angle error, the derivative of the vehicle state vector is calculated as $\dot{x} = Ax + Bu$.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f} + 2C_{\alpha r}}{m} & -\frac{2C_{\alpha f}l_f + 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_zV_x} & \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2 - 2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \quad (9)$$

The vehicle model applies a linear tire model, where the lateral force is linearly proportional with the sideslip angle for the entire sideslip domain, which approximation is considered correct if the tires operate on the linear region of the tire characteristics which is shown in Fig. 2.

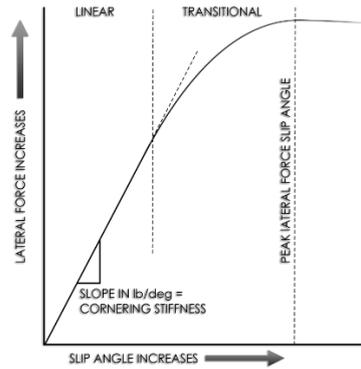


Fig. 2 Characteristics of a tire

The lateral and angular errors are interpreted at the closest point of the reference path.

3 MPC structure

The controller aims to drive the vehicle along the path while the lateral and angular errors converge to zero and stay at this value. The basic advantage of MPC is the continuous online optimization based on a built-in system model. The controller computes the series of the optimal control inputs in every time step as a row vector for a fixed horizon which is the control horizon (N_c) and applies only the first element of the vector on the controlled system. The control objective is defined by a cost function that needs to be minimized by the optimization. In this case, the cost function is (10)

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (10)$$

where R_s vector contains the reference, which is a zero vector in this case, since the reference lateral and angle errors is zero, Y is the system output vector and U is the control input. The reference is defined as (11), where $r(k_i)$ contains the zero values as references.

$$R_s^T = \overbrace{[1 \ 1 \ \dots \ 1]}^{N_p} r(k_i) \quad (11)$$

$$Y = Fx(k_i) + \phi \Delta U \quad (12)$$

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; \quad \phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \quad (13)$$

where F and ϕ are used for predicting the states for the prediction horizon. In this paper, the control horizon and the prediction horizon are treated as equal values as $N_p = N_c$. The value of J is minimal when the first derivative of J is equal to zero. The optimal solution for the control signal ΔU is

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} \phi^T (R_s - Fx(k_i)) \quad (14)$$

In (14) R is reflected in the size of ΔU when the objective function J is made to be as small as possible. R is a diagonal matrix

$$\bar{R} = r_\omega I_{N_c \times N_c} \quad (r_\omega \geq 0) \quad (15)$$

where r_ω is used as a tuning parameter. In the case that $r_\omega=0$ represents the situation where we would not want to pay any attention to how large the ΔU might be and the goal would be solely to make the errors as small as possible. In the case when r_ω is large the cost function describes a situation where would carefully consider how large the ΔU might be and reduce the error.

4 Results

A reference path is built for testing the controller, the tests are conducted at different vehicle velocities. The path is shown in Fig. 3, while the results are presented in Fig. 4 and Fig. 5 for the 40 km/h and the 50 km/h cases, respectively.

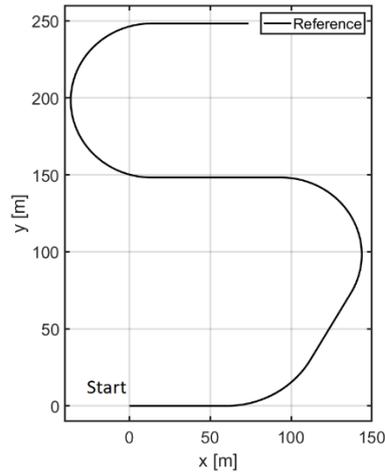


Fig. 3 The reference path

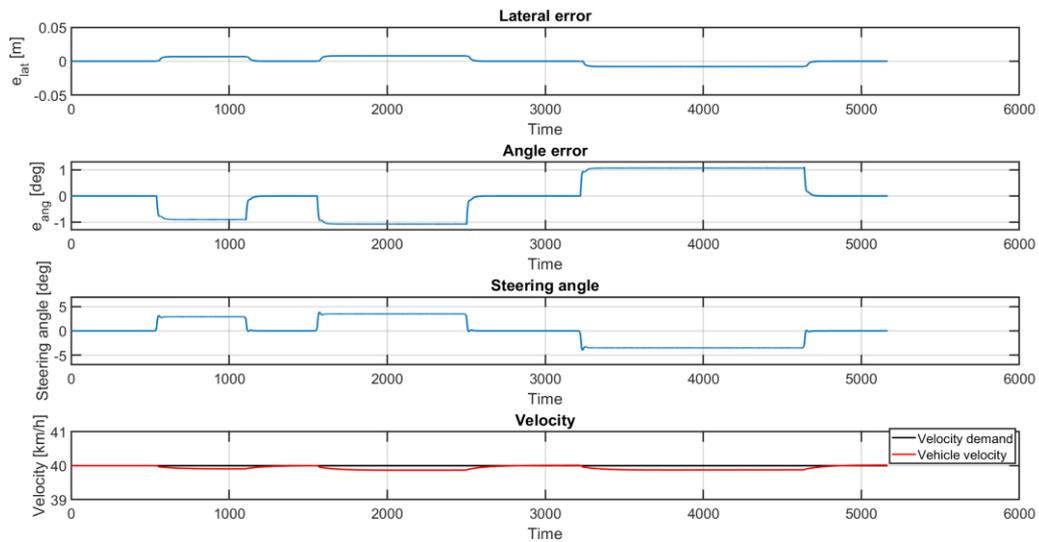


Fig. 4 The simulation results at 40 km/h

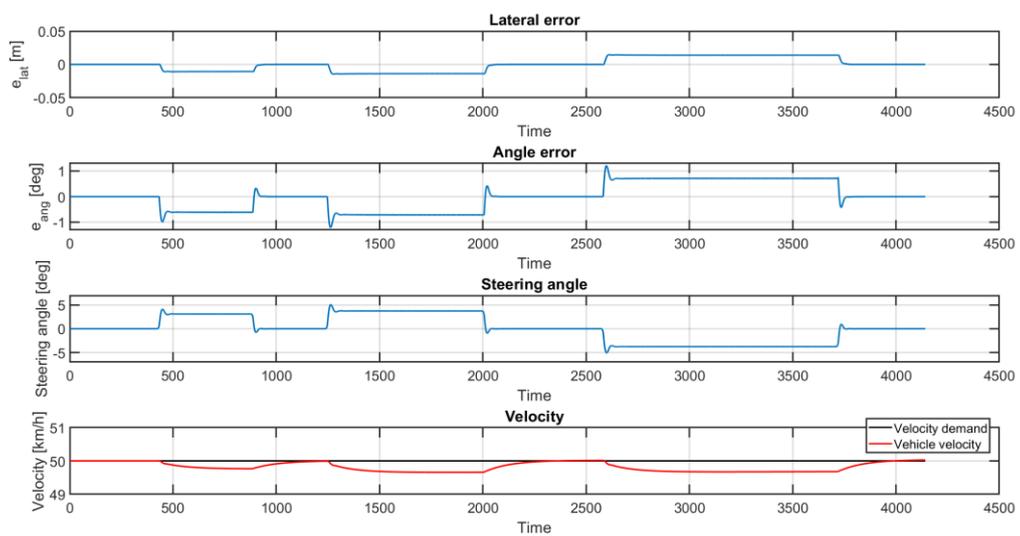


Fig. 5 The simulation results at 50 km/h

As shown by the steering angle in Fig. 5, the controller tends to become unstable at 50 km/h, due to the higher sideslip angles reached at the corners. As stated before, the state prediction is based on a linearized tire model, which is modeling the tire behavior correctly solely on the linear region. Beyond the linear region, the state prediction becomes incorrect which is responsible for the limited application of the controller. However, at moderate vehicle velocity, the controller shows an accurate and stable path following performance.

5 Conclusion

In this paper, a MPC for automated vehicle path following application is proposed. The controller proved to be stable until 50 km/h vehicle velocity, where the predicted and the real states are started to deviate from each other due to the nonlinearities of the tires. Thus, the proposed controller is applicable until the tires are operating on the linear region.

Possible further research direction is the inclusion of the tire nonlinearities into the state prediction, e.g. apply a LTV-MPC structure, or nonlinear MPC methods, the disadvantage of which is the higher computational requirement.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Z. Lu, B. Shyrokau, B. Boulkroune, S. van Aalst and R. Happee, "Performance benchmark of state-of-the-art lateral path-following controllers," 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC), 2018, pp. 541-546, doi: 10.1109/AMC.2019.8371151.
- [2] Viana, I. B., Kanchwala, H., Ahiska, K., and Aouf, N. (February 4, 2021). "A Comparison of Trajectory Planning and Control Frameworks for Cooperative Autonomous Driving." ASME. J. Dyn. Sys., Meas., Control. July 2021; 143(7): 071002. <https://doi.org/10.1115/1.4049554>
- [3] Í. B. Viana, H. Kanchwala and N. Aouf, "Cooperative Trajectory Planning for Autonomous Driving Using Nonlinear Model Predictive Control," 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), 2019, pp. 1-6, doi: 10.1109/ICCVE45908.2019.8965227
- [4] R. Hindiyeh, "Dynamics and control of drifting in automobiles," PhD dissertation, Stanford University, 2013.
- [5] R. Rajamani, "Vehicle dynamics and control," in Mechanical Engineering Series, 2nd ed. Springer, Boston, MA, 2012.

On-Site Test Measurements at ZalaZONE Automotive Proving Ground for Aiding Various Research Projects

Zsolt Vincze^{1a}, András Rövid¹

¹ Department of Automotive Technologies, Faculty of Transportation Engineering
and Vehicle Engineering, Budapest University of Technology and Economics

^a Corresponding author: vincze.zsolt@kjk.bme.hu

Abstract

The ZalaZONE Automotive Proving Ground can provide multiple controlled and safe test environments which becomes essential when various measurements and field tests are required for different vehicle-related scientific research projects. The Motorway and Smart City platforms are detailed models of a real highway section and a densely populated city area. These test sites were used to set up multiple field tests for various Intelligent Transportation Systems related research projects we are working on. Various measurements were performed by deploying multiple sensors of different types along the Motorway module as well as in the SmartCity zone in form of sensor stations. The Smart City platform was used to perform pre-accident specific scenarios and collect corresponding data in order to support the development of methods which are aiming for the recognition of traffic situations where the occurrence of an accident is highly probable.

Keywords: ground truth powered measurements, multi-sensor measurement, multi-station system

1 Introduction

To test already developed methods or start the implementation of new algorithms, sometimes test data becomes indispensable. Collection of such data often proves to be difficult, especially when it is related to specific traffic situations or diverse traffic environments. Both cases involve the use of vehicles, which need to be operated in a safe, controlled environment to avoid damage in life or property. Although, the use of a public road section for testing and data collection might be an alternative [1], this solution would require significant preliminary organization work. Furthermore, tests performed on public roads cannot easily be repeated if such demands arise.

To overcome this barrier, numerous automotive proving grounds are built, several of them can offer high detail models of different traffic environments, as well. At these locations the tests are easy to set up and perform, all the participating drivers can be skilled official test drivers, who can safely perform the different traffic scenarios. The participating cars can be altered or equipped with additional test equipment according to current demands.

In Hungary, the ZalaZONE Automotive Proving Ground provides this environment for various tests and related measurements. It has seven main modules, some of them are optimized for technical vehicle tests, like the Dynamic Platform or the Braking Platform. There are also modules enabling to perform testing in diverse environments such as highway, rural road or densely populated areas. During the planning and building, the proving ground was carefully optimized to support wide range of autonomous car tests [2, 3].

2 Measurements at ZalaZONE

Many projects hosted by BME Automated Drive Lab are linked to a cooperative perception system [4], which is currently under development. The research project reached the state of small-scale testing at designated sites appropriate for sensor deployment. However, before the deployment of sensors a lot of questions must be answered regarding the sensor types and sensor setup. Preliminary simulations were performed to find the theoretically optimal configuration for infrastructure sensors. Beside the simulations, real life tests were also needed to identify further potential problems. For the evaluation of our sensor system, the Motorway platform of ZalaZONE was selected. The platform is an actual highway section, with an overpass. This bridge can be seen in Fig. 1 which

provided the possibility to set up the experimental sensor system above the road surface in any chosen point in lateral direction.



Fig. 1 Motorway platform with the overpass element (left) and Smart City module (right) of ZalaZONE Vehicle Proving Ground

One of the planned features of the cooperative perception system is the ability to extract specific scenarios. After the development of this feature, the system will be able to recognize traffic situations where an accident is imminent with high probability and trigger recording. Such recorded data might be useful for automotive companies and accident reconstruction. In order to develop this feature, data covering various scenarios is required. For this purpose, highly dynamic tests were planned and performed to collect the necessary data. The Smart City platform was used to perform these tests. This module represents a densely populated city area, with crossroads, and pedestrian crosswalks. An appropriate crosswalk region has been selected as the location of preplanned highly dynamic scenarios.

Both the Motorway and Smart City platforms were scanned by a high precision digital mapping system, thus high-density point clouds are also available. These models have precise position information wrt. a global coordinate system for every point in the corresponding point clouds.

2.1 Multi-station data recordings

On the Motorway platform, multi-station test measurements were conducted. For the tests, two recording stations were deployed. Each station collected camera images and LiDAR point clouds with its own sensor set. The stations were synchronized in time - by using NTP (Network Time Protocol) [5] - to the clock of the GPS system with Cohda devices. This allowed proper comparison of the two recordings. The first station (Station A) was deployed onto the railing of the overpass element. The sensors were located at approximately 6m height above the ground. In lateral direction, the sensors were in the middle of the left half (incoming traffic direction) of the Motorway section. The second station (Station B) was placed at the left side of the left half of the road. The sensors were near the side railing of the road. The setup is shown by Fig. 2. Station A was equipped with a 4D traffic radar, a camera and a LiDAR unit. The camera was a 2MP unit with 60° FOV (Field Of View) lens. The image rate was 30 frames per second, global shutter mode was used to eliminate the distortion of fast-moving objects in the image. The LiDAR units at Station A were a close-range unit, with 64 laser beams, all directed horizontally in range between 0° and -45°, and a long-range type with 128 laser beams. Station B consisted of a 2MP camera, and a long-range type LiDAR. The data collection at each station was performed by a nearby PC, with RTMaps data logging framework. This framework saves all data with timestamps and can play back the recordings synchronized in time.

Calibration points were selected and measured with a GNSS device in order to estimate the extrinsics of sensors wrt. UTM frame. The marked calibration points have been acquired by cameras as well as by LiDARs of both stations. This information allowed to project the point clouds from each local coordinate system to a global one, merging the separate point clouds into a single cloud, resulting higher point density. For extrinsics estimation the Levenberg Marquardt optimization algorithm [6] and the Iterative Closest Points method [7] were used.

The test vehicles were equipped with high precision GNSS devices to log their precise position and orientation. This information was considered as ground truth. Two of the test vehicles were instructed to drive besides each other and the third vehicle had to follow them in the inner lane. This scenario was performed with different speed levels: 50km/h and 100 km/h.



Fig. 2 Station B (left) and Station A (right)

2.2 Multi-sensor perception and LiDAR performance tests

Multiple types of LiDAR sensors are available on the market, but to choose the appropriate type to be used in a given traffic environment additional information was necessary. Simulations were carried out to evaluate the performance of 360° horizontal FOV mechanical rotating units and solid-state technology-based types, with smaller FOV but increased resolution. According to the simulation results, when the sensor set is deployed in a road crossing, the 360° types are far more cost effective, and the available resolution is sufficient for the required detection range. In a highway environment, a solid-state LiDAR can have advantage over the rotating one. Since the region of interest can be defined by two frustums, the points falling outside this region (provided by 360° FOV LiDARs) are neglected. However, the reduced FOV of the solid-state variant comes with an increased resolution, which means, that the objects are represented by more points in a point cloud at a given distance compared to the rotating variant. This phenomenon extends the detection range, as well. Before the final decision regarding the sensors used in the highway environment, the simulation results were verified by various field tests with actual sensors. The field tests were conducted on the Motorway element at ZalaZONE proving ground.

During the tests, we tested a complex perception architecture, consisting of numerous sensor types. Also, different LiDAR devices were evaluated. The perception equipment included a traffic radar, a 2MP camera with 60° FOV lens, a far infra-red thermal imaging camera, and the evaluated LiDAR units. The reference types were a 64-channel variant for close-range, and a 128-channel variant for long-range. The evaluated units were a close-range type with 32 channels and a 128-channel long-range variant. On the second test session, the perception system was a simplified version of the first setup, i.e. beside the 2 MP camera, and the reference LiDAR, a solid-state type lidar was used as the evaluated item. In both cases, the data recorders were synchronized to the GPS time with Cohda devices. Fig 3. shows the two sensor setups.



Fig. 3 Perception sensor system at the first test session (left) and at the second test session (right)

2.3 High dynamic traffic situation tests

The development of the scenario extraction feature of the cooperative perception system started with the data collection phase. An appropriate cross walking has been selected, and a mobile sensor setup was deployed. The setup consisted of a 64 channel mid-range LiDAR, a 2MP industrial camera and an IP camera. The recorded scenarios were traffic situations, where a vehicle with high speed is approaching a turning vehicle or a pedestrian who walks across the road and must brake hard to avoid the accident. These scenarios were hazardous for human safety and for the test cars, thus professional test drivers had to perform the scenarios. As pedestrian a controllable dummy was used. Fig. 4 shows the testing scene.



Fig. 4 The sensor station, the dummy, and the calibration point markers

3 Results and Conclusion

The recorded data during the above-mentioned test measurements is slightly more than 565GB-s of point clouds, camera images, radar object lists and GNSS information from the test vehicles and the calibration points. The recorded data is useful for testing and validating functions of the cooperative perception system and to give support for various environment perception related activities as for instance the automatic label injection, object detection in point clouds with classical methods, low level sensor fusion-based 3D detector development, etc. For example, the GNSS positions of the test vehicles can be visualized as Fig. 5 shows. The dense areas of the test vehicles are scanned by the nearby Station B. The point cloud from Station A shows the front sides of the vehicles, which are occluded from the other LiDAR sensor, and the second measurement station itself.

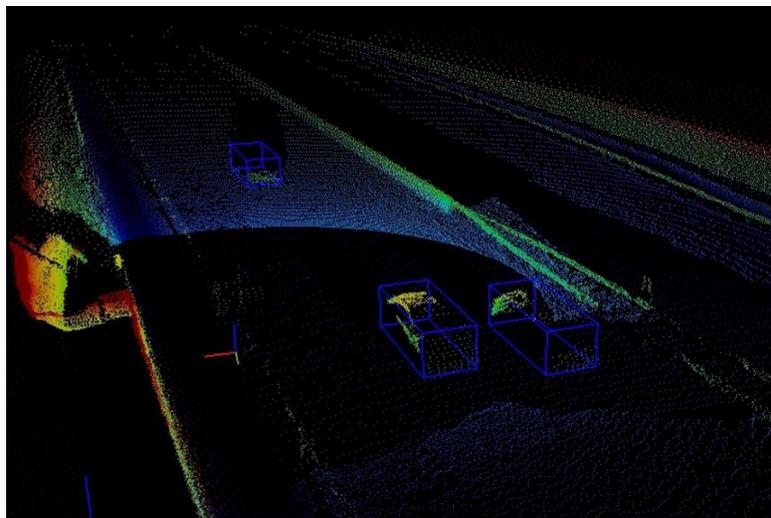


Fig. 5 Visualized GNSS ground truth in the merged point clouds of LiDAR sensors

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Tihanyi, V. et. al. (2021) Motorway Measurement Campaign to Support R&D Activities in the Field of Automated Driving Technologies, *Sensors* 21, no. 6: 2169. <https://doi.org/10.3390/s21062169>.
- [2] Szalay, Z., Tettamanti, T., Esztergár-Kiss, D., Varga, I., Bartolini, C. (2018) Development of a Test Track for Driverless Cars: Vehicle Design, Track Configuration, and Liability Considerations, *Periodica Polytechnica Transportation Engineering*, 46(1), pp. 29–35. <https://doi.org/10.3311/PPtr.10753>.
- [3] Szalay, Z., Nyerges, Ádám, Hamar, Z., Hesz, M. (2017) Technical Specification Methodology for an Automotive Proving Ground Dedicated to Connected and Automated Vehicles, *Periodica Polytechnica Transportation Engineering*, 45(3), pp. 168–174. <https://doi.org/10.3311/PPtr.10708>.
- [4] Tihanyi, V., Rövid, A., Remeli, V., Vincze, Z., Csonthó, M., Pethő, Z., Szalai, M., Varga, B., Khalil, A., Szalay, Z. (2021) Towards Cooperative Perception Services for ITS: Digital Twin in the Automotive Edge Cloud, *Energies*, 14, 5930. <https://doi.org/10.3390/en14185930>.
- [5] Mills, D. L. (1985). Network time protocol (NTP).
- [6] Ranganathan, A. (2004) The levenberg-marquardt algorithm, *Tutorial on LM algorithm* 11, no. 1: 101-110.
- [7] Bouaziz, S., Tagliasacchi, A., Pauly, M., (2013) Sparse iterative closest point. In *Computer graphics forum*, vol. 32, no. 5, pp. 113-123. Oxford, UK: Blackwell Publishing Ltd,.

Teleoperation with a Real-time Digital Twin

Viktor Remeli^{1a}, m Boronyk¹, Viktor Tihanyi¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: remeli.viktor@kjk.bme.hu*

Abstract

One of the state-of-the-art trends in intelligent transportation systems research is the development of real-time digital twin technologies and their applications for increased safety and autonomy on the roads. The areas of deployment also include test tracks, logistic yards, factories, hospitals, airports, railways, and any such constrained and managed environment where transportation automation may be of benefit. While applying the same principles as for open road networks, managed environments allow the efficient deployment of optimally distributed sensory infrastructure while also observing certain simplifying assumptions (constraints on the expected scenarios), both factors leading to earlier adoption and ROI. In the case of fleets or swarms of automated vehicles, certain cost savings can be enacted by recovering the environmental model exclusively from infrastructure sensors. However, such a setup is never exempt from the necessity of occasional manual intervention performed by a teleoperator. In our current paper we explore system and HMI designs that would facilitate safe and efficient on-demand teleoperation functionality in such a system.

Keywords: *digital twin, ITS, real-time, teleoperation*

1 Real Time Digital Twin

There is an increasing demand for functionalities and services that can be derived from the real time digital twin of highly automated environments. A highly automated environment consists of a distributed system of heterogeneous platforms that must work together towards common and individual goals. Smart roads and intelligent and connected vehicles of the future comprise such an environment. Logistic, agricultural, and military uses also come to mind as related areas with some overlapping requirements. Advances in highly automated manufacturing colloquially known as industry 4.0 [1] point in a similar direction: an Internet of Things (IoT) network of subsystems must establish efficient cooperation even though aspects of sensing, decision making, and control are highly distributed.

While the employed solutions and technologies differ depending on the application, real time digital twin systems can be most beneficial when the task to be solved involves a distributed environment perception problem. The objective of the digital twin is to maintain a real time, consolidated and consistent digital replica of the environment [2]. This joint model should be more reliable and complete than any model the individual perception systems could provide by themselves. A clear case is provided by objects that cannot be detected from a given platform due to cover or the lack of appropriate sensors. Another, slightly related case is the need for coordination of platforms in certain tasks, or their remote management.

The digital twin can be established in the cloud or on the platforms themselves. A logically centralized digital twin can be physically centralized, or physically distributed. We can also imagine cases where several competing or complementary digital twins might be employed, as in the case of peer-to-peer communicating platforms. In this paper we will explore the possibilities relating to the cloud-based, logically centralized digital twin. The exact deployment of the data processing steps (platform-side, edge-side and central) will be of less consequence to our discussion. For more details on the implementation of intelligent transportation and automotive digital twin systems of the mentioned type, refer to previous work, notably including [2] and [3].

The specific problem we want to address within the context of the digital twin is the task of manual teleoperation of the participating platforms. This kind of task is expected to come up in nearly any industrial use-case or scenario, as the business workflow must be ensured even in case of environmental disturbances or platform malfunctions.

A cost-effective method of mitigating a large proportion of such situations is remote human teleoperation of the otherwise autonomous system. Another field of interest is telemanipulated robotics that focuses on remotely solving dexterous tasks with humanoid machines and robotic arms [4], operating in harsh environments like elevated radiation risk [5], or solving complex tasks that require specific and remote expertise like surgeries, see [6].

In this paper we explore how such tasks can be approached when a real-time digital twin of the operation environment is readily available.

2 Teleoperation workflow

Teleoperation can proceed according to different workflows and can require different approaches strongly dependent on the constraints of the communication medium – latency and bandwidth [7]. Below we describe some high-level ideas for a general approach that is suitable for most use cases involving a number of teleoperable platforms and a – usually much smaller – number of operators.

The teleoperation task in general begins with a *triggering event*, which first stops certain aspects of the ongoing automation, second it generates a *notification* towards the teleoperating personnel. The operator then decides whether to solve the situation using manual *remote-control* subsystems or to further escalate it. After the situation has been solved, the control must be *handed back* to the automation system. Thus, the simplified sequence of phases in a typical teleoperation workflow are:

- ***Handover phase***
- ***Teleoperation phase***
- ***Handback phase***

2.1 Handover phase

The handover phase consists of three discernible (but not necessarily sequential) steps for the platform:

- it begins with the *detection* of the trigger event,
- it continues with an *emergency maneuver* intended to stop affected aspects of the automation,
- and it finishes by issuing an *event update* towards some (centralized) event broker.

The platform steps are then followed by steps taken by the notification subsystem and the operator:

- incoming events generate necessary *notifications* towards affected parties (operators),
- operators confirm *receipt and acknowledgement* of notification,
- operator initiates coordinated *handover* of control to the operator.

The digital twin can be especially useful in detecting the trigger event, in coordinating the emergency maneuver, and in serving as the event, notification and handover broker. We will explore these possibilities step by step.

2.1.1 Trigger event detection, maneuver, and update

Detection of the triggering event is more likely and more accurate if a platform can rely on sensors of other platforms as well. This can also be cheaper or even unavoidable if outside (off-platform) sensors are in some way more suited for the task, e.g. a loitering drone (single or swarm) over the field of operations. In this case, the central system performs the fusion of the incoming data streams to produce a unified environment model. Real-time requirements, especially regarding raw data processing and communication channels, are crucial in this regard. Such requirements will finally decide whether processing is done on-platform and whether raw data is communicated. It is often the case that only processed metadata is sent to the cloud. Therefore, central fusion usually occurs at object-level. For example, the current spatial state of intelligent platforms and other important participants or obstacles is estimated in real time based on incoming detection or track streams. The detection of a trigger event pertaining to one or more platforms must then occur also in real time, and preferably on the integrated environment model that is considered most robust.

In fact, trigger events can be detected in three main ways: (a) a platform locally detects a trigger event pertaining to itself, (b) a platform locally detects a trigger event pertaining to other platforms (as well), and finally, (c) the central system detects an event that affects one or more platforms. In case (a), the emergency maneuver should be initiated immediately, and the event update sent to the central broker. In case (b), the event update must be sent to the cloud: the affected platforms are immediately notified by the central system to perform the proper maneuvers.

This can also be achieved in a peer-to-peer (P2P) fashion. The triggered platforms must take care to deduplicate events which might point to the same situation only detected from different broadcasting sensory platforms. When case (b) is implemented with central processing, however, the deduplication is a form of data fusion that happens centrally. Case (c) assumes that fusion of perception sources happens on a lower level, prior to the event detection.

In all cases (b) and (c), and sometimes even (a), event data is backpropagated to the appropriate platforms without delay allowing them to perform the emergency maneuver in time. The exact maneuver decisions and controls are calculated either on the platform itself, or in the cloud. How this is handled depends not only on the compute capacity of the setup, but also on whether the set of environment data necessary for maneuvering is more readily available in the cloud (via collective perception) or on the platform (via unique sensors that are not connected to the central system). The best of both worlds is achieved when the platform receives a fused collective perception data stream from other sensing platforms and installations, while also having access to its own sensors.

2.1.2 Notification and handover

Every time a trigger event occurs for a platform, an update is dispatched to the central broker and awaits further processing to notify operators. A subscribe-publish scheme can be implemented to allow different stakeholders to be notified of different events via different channels, and even using differing schedules (e.g. some may need immediate SMS notification while some may just want a weekly summary in their email). For audit purposes, it might be important to develop ways to confirm the date and time of notification receipt and acknowledgement. Since the requirements for the notification system are fairly general, certain software tools like [8] may already support much of the required functionality out of the box.

The central system can also act as an arbiter for the handover, especially when multiple operators might want to take control of the same system at the same time. The management of the teleoperation state of the platforms can be handled centrally.

2.2 Teleoperation phase

Using the digital twin approach to teleoperation, the advantages over classical manual operation go beyond the reduction of the need for personnel to be present physically (passively on alert or actively operating) at various locations, sometimes in rapid succession. Notably, extended perception, virtual reality and central management aspects contribute to unique and crucial capabilities that cannot be delivered by traditional setups.

2.2.1 Extended perception

The digital twin allows for extended environment perception using multiple sensors and sensors of different platforms. This can have several advantages, starting with the obvious capacity to teleoperate based on raw data from many disparate sources. Relevant camera images (or other sensor data) can even be injected dynamically into the teleoperation user interface according to the state (e.g. position and trajectory) of the controlled platform and its relation to all the surrounding intelligent platforms and installations.

We note that the conventional kind of teleoperation that relies on a single sensor data stream (video camera without a digital twin) usually reduces the situational awareness of the operator, as it usually lacks both spatial and modal aspects of the data – e.g. a fixed field of view and no audio. Extended perception allows more precise and safer remote manual control not only due to better situational awareness of the teleoperator, which is the major factor, but also as it enhances the possibility for further electronic assistance mechanisms like collision avoidance (emergency braking), and other ADAS (automated driving assistance) and related functions.

2.2.2 Visualization

Since some (or all) data may also be available in processed form, the concept of extended perception can be visualized in an augmented reality continuum ranging from a collage of video streams (entirely real) up to a 3D simulation (entirely virtual – so called synthetic vision systems [9]), including all stages in between (e.g. point clouds and bounding boxes projected on images and other attention assistance and information conveying methods much like heads-up displays on modern aircraft). Certainly, in visualizing such complex environments, augmented reality (AR) glasses can be of great help, for instance see [10]. The ergonomic design of such interfaces is of notable importance, as they must support the recommendation, choice, and visual integration of the most relevant and useful information streams. A typical problem of data integration and joint visualization may arise around

time synchronization of different data streams due to a range of factors including differing latencies experienced on various channels.

2.2.3 Central management

Intelligent coordination of automatically and remotely controlled operations quickly becomes a requirement in any system that has a large number of platforms and/or teleoperators. Besides unified handling of authentication, authorization and other security related issues, resource allocation and deconfliction are also among the obvious benefits of central management, another one is auditing. A further benefit can be that the heterogeneity of the platforms can be encapsulated and made transparent to the user, who only must know how to access the central teleoperation interface.

It can be argued that the necessity for centralization could become even more pronounced if the controlled system is a complex machine that requires the simultaneous coordination of several operators (e.g. certain military platforms), or conversely, if several simple machines have to coordinate in a common environment. The same is true for coordinating manual control in semi-automatic platforms and in environments with platforms of various levels of automation.

2.3 Handback phase

The central system setup can aid the operator in finding the proper time, location, mode, etc. for the re-automation or handback of a given aspect of control, with particular regard to other platforms' state and the mission objectives. The central system can also oversee the timed handback at a later or more opportune moment without the need for the operator's intervention or presence. The mission of the re-automated platform can be a default mission, one defined by the operator via the central system, or one chosen by an algorithm that decides using the central environment and system state model.

3 Conclusion

We have examined the theoretical possibilities of teleoperation in highly automated environments with a centralized real-time digital twin. We conclude that depending on the problem type and complexity, there can be significant potential benefits to be explored and exploited using the digital twin setup. The benefits derive from two main properties: the *common environment model* allows better event detection, maneuvering and teleoperation via extended perception, while the *central state management* of the system helps with resource allocation, notification and auditing.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Lasi, H., Fettke, P., Kemper, H. G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & information systems engineering*, 6(4), 239-242.
- [2] Tihanyi, V., Rövid, A., Remeli, V., Vincze, Z., Csonthó, M., Pethő, Z., ... & Szalay, Z. (2021). Towards Cooperative Perception Services for ITS: Digital Twin in the Automotive Edge Cloud. *Energies*, 14(18), 5930.
- [3] Krämmer, A., Schöllner, C., Gulati, D., & Knoll, A. (2019). Providentia – a large scale sensing system for the assistance of autonomous vehicles. In *Robotics: Science and Systems (RSS), Workshop on Scene and Situation Understanding for Autonomous Driving*.
- [4] Jang I., Carrasco J., Weightman A., Lennox B. (2019) Intuitive Bare-Hand Teleoperation of a Robotic Manipulator Using Virtual Reality and Leap Motion. In: Althoefer K., Konstantinova J., Zhang K. (eds) *Towards Autonomous Robotic Systems. TAROS 2019. Lecture Notes in Computer Science*, vol 11650. Springer, Cham. https://doi.org/10.1007/978-3-030-25332-5_25

- [5] Allspaw, J., Roche, J., Yannuzzi, M., Yanco, H. A., & Lemiesz, N. (2018, July). Remotely Teleoperating a Humanoid Robot to Perform Fine Motor Tasks with Virtual Reality-18446. WM Symposia, Inc., PO Box 27646, 85285-7646 Tempe, AZ (United States).
- [6] Meng, C., Wang, T., Chou, W., Luan, S., Zhang, Y., & Tian, Z. (2004, April). Remote surgery case: robot-assisted teleneurosurgery. In IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 (Vol. 1, pp. 819-823). IEEE.
- [7] Lichiardopol, S. (2007). A survey on teleoperation. Technische Universitat Eindhoven, DCT report, 20, 40-60.
- [8] Amazon Simple Notification Service <https://aws.amazon.com/sns>
- [9] Alexander, A. L., Wickens, C. D., & Hardy, T. J. (2005). Synthetic vision systems: The effects of guidance symbology, display size, and field of view. *Human Factors*, 47(4), 693-707.
- [10] Szalai, M., Remeli, V., Tihanyi, V. (2022). Development of 3D Visualization for Displaying Real-time Sensor Information from Multiple Sources. First Conference of ZalaZONE Related R&I Activities of Budapest University of Technology and Economics 2022

Simulation of Road Traffic Accidents Related to ADAS Systems in PreScan

Tamás Márton Kazár^{1a}, Zsombor Pethő¹, Gábor Vida¹, Árpád Török¹

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

^a *Corresponding author: kazar.tamas@kjk.bme.hu*

Abstract

This paper describes a simulation of a specific accident reconstructed in PreScan software. Through this, we show that the PreScan software can be efficiently applied in accident reconstruction, particularly the operation of ADAS systems can be explored and analyzed. Beyond this, we also demonstrated during our research that the investigated simulation environment can be used efficiently to evaluate the reliability and safety of electronic control units developed to control highly automated vehicle functions. Consequently, we also proved the applicability of concept related to the hybrid multi-agent simulation environment (especially considering the ZalaZONE ecosystem) including differently controlled components (e.g. fully autonomous, human-driven or highly automated vehicles). In the simulation, we examine the Adaptive Cruise Control (ACC), Lane Keeping Assist (LKA), and Automatic Emergency Braking System (AEBS) in the accident. Evaluating the results, we found that the lane-keeping system is very sensitive to the quality of pavement signals. In the selected case, poor-quality signs likely played a major role. On the other hand, it was clear that the properly working safety systems would have been effective in reducing collisions speed, and a longer tracking distance would have had a positive effect. In the case of the adequate operating FCW, the driver would have had sufficient time to intervene and stop the vehicle. In the final section, we analyzed how PreScan can be used to connect an external device to the simulation environment.

Keywords: *Adaptive Cruise Control (ACC), Automatic Emergency Braking System (AEBS), Lane Keeping Assist (LKA), PreScan*

1 Introduction

Regarding a crash sequence shown in Fig. 1, Advanced Driver Assistance System (ADAS) has two significant roles in a vehicle. Under normal driving conditions, it acts as a convenience feature by enabling automated driving and reducing the workload of the human driver. When the danger of a potential crash appears, the safety side of ADAS becomes dominant. By warning and assisting the driver and/or taking preventative actions automatically, they are effective at avoiding or at least mitigating the effects of a collision. Most modern vehicles already have some driver assistance systems installed. Beginning from 2022 it has become mandatory to include LKA and AEBS in all new models of passenger cars sold in the EU. However, these systems are prone to malfunction and still require supervision from the human driver. To further improve these systems, it is important to analyze crashes where ADAS failed to intervene.

2 Related works

PreScan is also a widely used co-simulation framework used primarily to develop automotive functions, as its sensor simulation capabilities make it well-suited for Hardware-in-the-Loop (HiL) development and validation tasks.

Miao, Q et al. [2] used PreScan to develop control algorithms for different ADAS and evaluate and optimize the performance of such systems using multiple traffic scenarios [6] based on real-life data. They found PreScan to be an effective, low-cost, and simple environment for developing safety systems [8], [9], [10].

Son, T. D. et al. [3] proposed a co-simulation framework for developing and testing ADAS and Autonomous Vehicle (AV) systems. They used Siemens Simcenter Amesim to develop the control algorithm, which provides a detailed vehicle dynamic model for the simulations [7], [11]. The Simulink-based Amesim simulation is interfaced

with PreScan, which is responsible for simulating the traffic scenario and generating sensor data using detailed sensor models to test and validate the control algorithms. The co-simulation enables the testing and validation of the systems to be more effective, saving time and cost by reducing the number of real prototype tests.

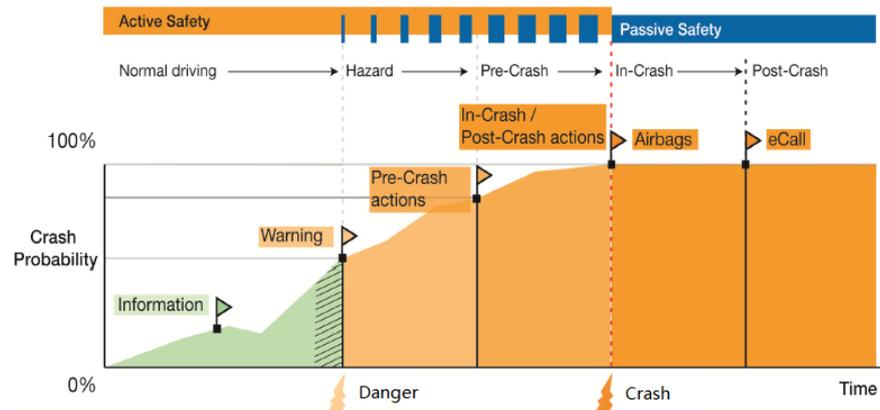


Fig. 1 Crash sequence [1]

Another group of researchers created a test system for the real-time evaluation of Lane Departure Warning System (LDWS). This contained a virtual reality system running PreScan to simulate different scenarios, a real-time platform to run a real-time vehicle model, and a real-time controller for the LDWS hardware. The LDWS was aimed at a monitor showing PreScan’s generated environment and communicated with the real-time controller via CAN-bus. They tested the system with multiple LDWS hardware from different manufacturers and found that PreScan’s versatility in the road, traffic, and environmental conditions makes it ideal for this kind of testing [4].

Similar studies describe the need for creating a complete test program for intelligent vehicle systems to standardize and speed up the validation and approval of such systems. To achieve this, they propose a multi-stage method consisting of simulation, hardware-in-the-loop testing, and test track testing. They summarize the scenario-building process of PreScan and describe the benefits of using it for the simulation stage. It enables testing the systems in a wide range of scenarios in a relatively short time which can also be automated. When a system passes these tests, the most important and critical scenarios are further tested in the consequent stages [5].

The implemented solution demonstrated that the applied simulation framework could be used efficiently to evaluate the reliability and safety of electronic control units developed to control highly automated vehicle applications. In accordance with this, we also proved the correctness of our concept, stating that it is possible to develop a hybrid multi-agent simulation environment including differently controlled components (e.g., fully autonomous, human-driven, or highly automated vehicles). The framework based on the new system concept can be deployed efficiently in cooperation with a proving ground (e.g., ZalaZONE [12]) environment applying digital twin technology, providing a feasible and reliable balance between the costs of test and validation processes and the high number of test scenarios required.

3 Methodology

PreScan was used to recreate a crash scenario based on real accident data where the failure of ADAS was found as a contributing factor. The chosen crash involving a Tesla Model X operating in Autopilot mode happened on March 23, 2018, near Mountain View, CA. According to the crash report issued by the NTSB (National Transportation Safety Board), Tesla was cruising on a multi-lane highway in the HOV (High-occupancy vehicle) lane. Approaching a highway interchange, a left exit HOV lane opens, which gets separated from the other lanes by a concrete barrier. Leading up to the barrier, the Tesla entered the gore area between the two HOV lanes and crashed into a nonoperational crash attenuator placed at the beginning of the concrete barrier. Data gathered from the Event Data Recorder (EDR) showed that Tesla’s assistance systems were activated leading up to the crash but didn’t recognize the danger, nor did the driver make any preventive action.

Tesla’s Autopilot systems consist of multiple ADAS components. In the PreScan simulation, LKA, ACC and AEB systems' functioning were analyzed. The applied LKA model uses a camera sensor to identify and follow lane markings. The ACC and AEB both use a long-range (150 m beam range, 9° horizontal beam angle) and a

short-range radar (60 m beam range, 80° horizontal beam angle), modeled with PreScan’s Technology Independent Sensors (TIS).

During the investigation, the NTSB found that the lane line on the left side of the gore area was more prominent and visible than on the right. It was proposed that this difference between the lane lines might cause the LKA system to follow the left line, explaining why the vehicle entered the gore area. PreScan’s line parameters “Fade” and “Hole” was used to determine how bad line quality affects the performance of the LKA system.

In an accident investigation, it is also essential to evaluate whether or how the accident could have been prevented. There was no identified reason why the ACC and AEB systems failed to recognize the obstacle. To assess these systems’ impact on collision speed – should they function correctly – headway times of 0.9 seconds (minimum setting on Tesla) and 2.0 seconds (maximum setting) were assumed. It was also evaluated whether the driver would be able to stop the vehicle before the collision, the AEBS should issue an FCW (Forward Collision Warning) alert. For this simulation, a reaction time of 0.7 seconds was assumed for the driver after the FCW alert.

The last section explores the concept of using PreScan as an environment for testing and evaluating ADAS or Automated Driving Systems (ADS) running on external hardware. We created an interface for connecting to the PreScan simulation and running the algorithms in real-time for Processor-in-the-Loop (PiL) or HiL tests. Thanks to the Simulink-based PreScan simulation, various networking blocks are available to establish communication with an external device connected to the same local network. We used a Raspberry Pi 4 Model B to test the connection, running a UDP server for communication and a simple control algorithm. Once the PreScan simulation is started the Simulink model is configured to connect to the server on the Raspberry as a client and send the required signals for the control algorithm. The Raspberry then feeds back the control signal to Simulink, which is propagated to the vehicle dynamic model. The proposed architecture is shown in Fig. 2.

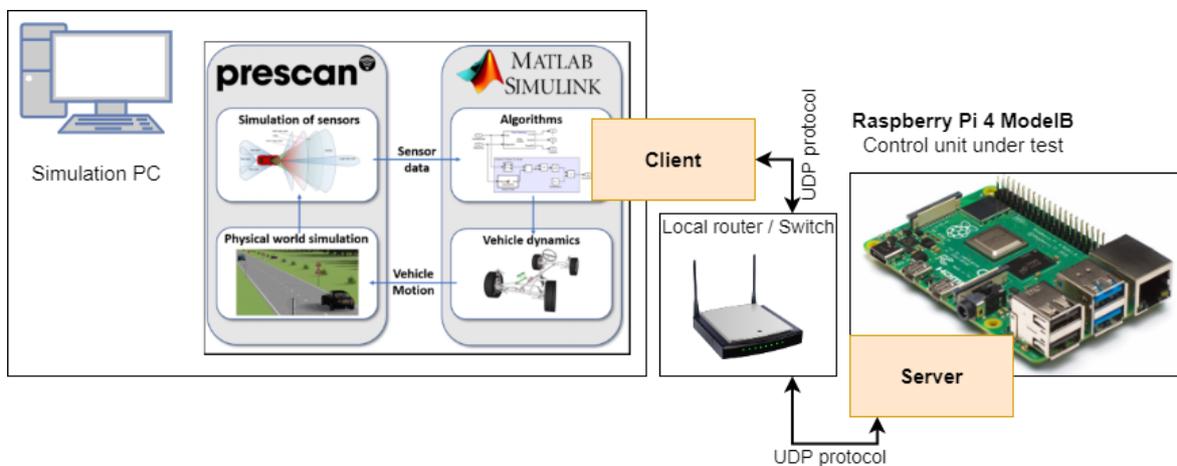


Fig. 2 Proposed system architecture for communicating with external hardware

The control algorithm receives throttle and brake signals along with simulation time. It is configured to echo back the received signals until a particular simulation time, after which full braking is initiated regardless of the incoming signals. The test can be interpreted as a hardware failure or even a cyberattack scenario where a malicious third-party intercepts the original signal and injects its own. By comparing the sent and received signal in Simulink, the latency introduced by the communication was measured to test whether this method is able to provide real-time performance.

4 Results and evaluation

By setting the right channeling line’s both “Hole” and “Fade” parameters from the ideal 0% to 20% the LKA system’s performance starting to degrade. At the start of the gore, where the two lines separate LKA starts to follow the left line by making a left steering input. However shortly after it is still able to correct itself by picking up the line on the right side of the gore and returning to the lane’s centerline. This steering maneuver is shown on Fig. 3. By raising both parameters to 40% the system completely ignores the right line and continues to direct the vehicle into the gore area. This is likely caused by the edge detection algorithm failing to identify the lower contrast and fragmented right line as the lane boundary.

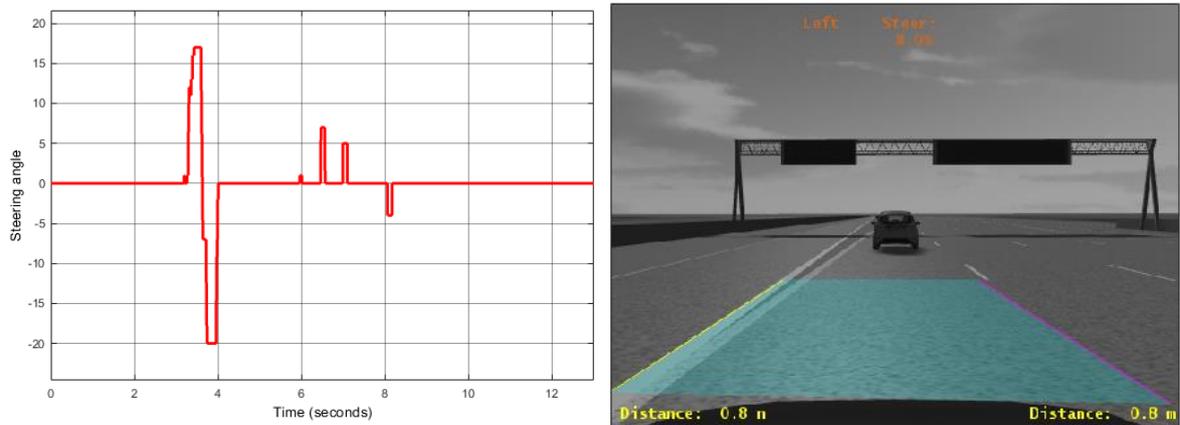


Fig. 3 LKA system's steering input and identified lane boundaries

In the accident the driver set the ACC systems demanded headway time (HWT) to the minimum 0.9 seconds. The simulations show that a properly functioning ACC system with this setting is still able to identify the crash attenuator as obstacle 2.5 seconds before impact but only starts braking 1.2 seconds before impact. Even with the short time available and limited braking capabilities of the ACC, it still manages to reduce the collision speed from 114 km/h to 99 km/h. With the maximum headway time setting of 2.0 seconds the braking starts immediately after detection which is 2.2 seconds before impact. This further reduces collision speed to 84 km/h.

Enabling the AEBS in the simulations resulted in improved braking. The detection times are identical, but unlike ACC, the AEBS can utilize the full braking potential of the vehicle. With HWT = 0.9 seconds the collision speed is 69 km/h and with HWT = 2.0 seconds the collision speed is 65 km/h.

Assuming the AEBS issues an FCW alert to the driver at the time of detection and the driver reacts to this with a reaction time of 0.7 seconds the simulation shows that he would be able to stop the car before impact even with the minimum headway time setting.

Regarding the Raspberry test system mentioned earlier a latency of 20 milliseconds was measured. This corresponds to exactly 2 time-steps since the simulation was running at 100 Hz. We found this to be acceptable. The benefit of this system is that it is based on UDP protocol which means it is easily scalable, multiple external devices can be connected. Since we are not using a real-time rig, the components can be easily swapped to support rapid prototyping for algorithm development.

5 Conclusion

PreScan simulation software – while built for development purposes – can be efficiently applied to analyze the performance of ADAS during an accident and determine what factors most likely contributed to the system's failure. The simulation can be further improved by using system models identical to those that ran on the crashed vehicle, but this is only possible with cooperation from the vehicle's manufacturers because these are protected under intellectual property laws.

We developed a method for testing an external ECU hardware model – that is, running a control program in real-time – to control a vehicle in a PreScan simulation. This work was the first step toward the long-term goal of developing an overall testing environment focusing on automotive electronic control units focusing on the security of the system and its modules, including algorithms, software frameworks, hardware and interfaces.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Xiang, You & Sun, Lei & Sui, Li & Kang, Jun & Jiang, Yong. (2016). Interactive Safety Analysis Framework of Autonomous Intelligent Vehicles. MATEC Web of Conferences. 44. 01029.

- [2] Miao, Q., Tang, X., Wang, D., Tideman, M., & Li, J. (2012, July). The Application of PRESCAN in the Concept Development of Active Safety System. In 2012 Third International Conference on Digital Manufacturing & Automation (pp. 884-886). IEEE.
- [3] Son, T. D., Bhave, A., & Van der Auweraer, H. (2019, March). Simulation-based testing framework for autonomous driving development. In 2019 IEEE International Conference on Mechatronics (ICM) (Vol. 1, pp. 576-583). IEEE.
- [4] Zhang, Q., Chen, D., Li, Y., & Li, K. (2015). Research on performance test method of lane departure warning system with PreScan. In Proceedings of SAE-China Congress 2014: Selected Papers (pp. 445-453). Springer, Berlin, Heidelberg.
- [5] Hendriks, F., Tideman, M., Pelders, R., Bours, R., & Liu, X. (2010, July). Development tools for active safety systems: Prescan and VeHIL. In Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety (pp. 54-58). IEEE.
- [6] Ortega, J., Lengyel, H., & Szalay, Z. (2020). Overtaking maneuver scenario building for autonomous vehicles with PreScan software. *Transportation Engineering*, 2, 100029.
- [7] Abdunazarov, J., Mikusova, M., & Kyamakya, K. (2021). The system dynamic and COMPRAM methodologies for modelling, simulation and forecasting of road safety of Uzbekistan. *Journal of KONBiN*, 51(3), 49-63.
- [8] Sebron, W., Tschürtz, H., & Krebs, P. (2019, September). Extending the shell model via cause/consequence analysis of component failures. In European Conference on Software Process Improvement (pp. 70-82). Springer, Cham.
- [9] Cokorilo, O. (2020). Urban air mobility: safety challenges. *Transportation research procedia*, 45, 21-29.
- [10] Praviionis, T., Eidukynas, V., & Sokolovskij, E. (2020). An analysis of the reliability of a bus safety structure on carrying out the numerical and experimental tests. *Sensors*, 20(24), 7092.
- [11] Maretić, B., & Abramović, B. (2020). Integrated Passenger Transport System in Rural Areas—A Literature Review. *Promet-Traffic&Transportation*, 32(6), 863-873.
- [12] Szalay, Z., Hamar, Z., & Simon, P. (2018, June). A multi-layer autonomous vehicle and simulation validation ecosystem axis: Zalazone. In International Conference on Intelligent Autonomous Systems (pp. 954-963). Springer, Cham.

Automotive Proving Ground HD Map Models

Dániel Panker¹, Tamás Tettamanti^{2a}

¹ Automotive Proving Ground Zala Ltd.

² Department of Control for Transportation and Vehicle Systems,
Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics

^a Corresponding author: tettamanti.tamas@kjk.bme.hu

Abstract

Self-driving vehicles also need maps - just like us drivers - for route plans, but traditional maps are often inaccurate and contain significantly less information than so-called High Definition Maps (HD). HD maps are usually based on laser measurements, thus the accuracy of each road element can be around 2 centimeters. Our goal is to properly support research in this direction as well, therefore we created an HD map of the selected modules of the ZalaZONE proving ground. The selected components are the High-Speed Handling course, Dynamic Platform, ADAS surface and Motorway section. These materials are accessible for download including the high-resolution and high-precision Lidar point cloud which the maps are based on, recorded with the Lecia Pegasus 2 system. We have implemented our HD maps in the increasingly widespread OpenDRIVE format in the automotive industry.

Keywords: HD map, OpenDRIVE, ZalaZONE

1 Introduction

The ZalaZONE proving ground is located in Zalaegerszeg, Hungary. The traditional proving ground features focusing on driving and driving stability are implemented together with the research and development infrastructure elements for future vehicle validation. The track also allows validation tests for electric vehicles and autonomous vehicles as well. [1] The construction work started in 2017 and will be finished in 2022 (Fig 1). ZalaZONE enables to carry out control scenarios, test automotive and communication technologies [2].



Fig 1 Visualization of ZalaZONE Proving ground

ZalaZONE Proving Ground enables complex test scenarios for both classical and for automated/autonomous vehicles. The main elements of the ZalaZONE test track are listed below:

- Dynamic platform: The dynamic element is a circular surface with a diameter of 300 m, provided with multi-layer asphalt pavement.

- Handling course: It is a 2 km long test track element where it is possible to examine vehicle behavior, vehicle handling, and test different technical settings under different conditions.
- ADAS surface: Created for EuroNCAP test with a standard junction on middle of the module.
- Noise-measurement surface
- Smart-city: Created urban environment, for testing autonomous vehicles in complex urban situations. It includes different lane types, junctions, topography, materials, traffic situations.
- Motorway: Typical Hungarian motorway layout, two plus one stop lane with entry and exit ramps.
- Rural road: Various country road like layout with different topography.

ZalaZONE in cooperation with University of Budapest decided to create virtual copy of proving ground elements. It is a great opportunity for industrial companies and researchers to test developments virtually on ZalaZONE. It is also reducing cost of development and helps test preparation. The main steps of the modeling process is shown in Fig. 2.

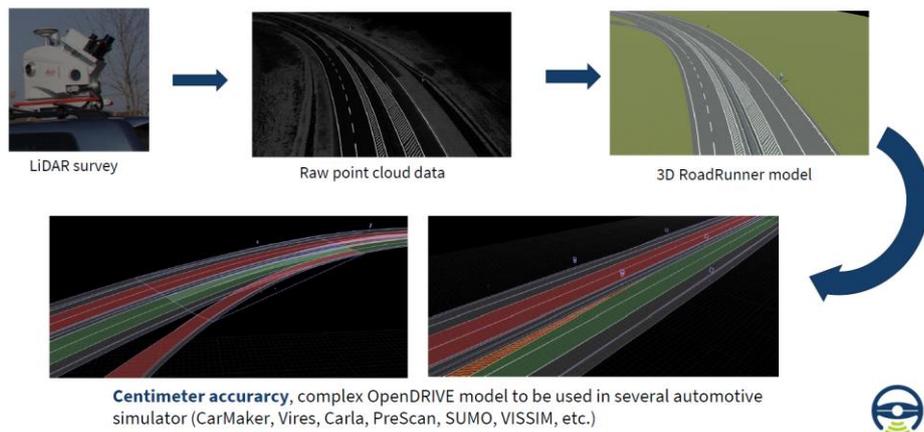


Fig 2 Steps of the creating ZalaZONE models

2 LiDAR survey

The first step in creating HD maps is to scan the completed track elements. It is important that the given module has all the final elements (e.g. paintings, road signs, guardrails) so we can use scanning as an accurate reference. Data collection was performed with a Leica Pegasus 2 mobile mapping tool. The essence of the method is to mount the measuring device on the top of a vehicle and travel along the route to be recorded at a relatively slow speed (15-30 km / h) depending on the track element and the desired accuracy. During the traversed section, the lidar of the measuring device continuously scans the asphalt or any other object it detects, resulting in a point cloud file 2-3 centimeters accuracy. The final result shown in Fig 3.

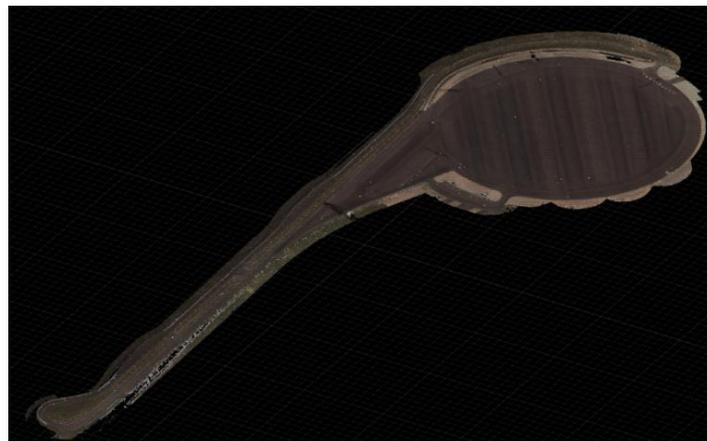


Fig. 3 Lidar measured point cloud file of the Dynamic platform in the modeling software

This point cloud file can be used as a 3D reference for modeling.

3 Virtual model creation

Simulation models are created with MathWorks RoadRunner software. The great advantage of the software - compared to its competitors - is that in addition to 2D images, we can also use 3D files as a reference (eg: aerial image, elevation data). With this procedure we can make accurate models in the absolute coordinate system (lat, long) centimeter. After importing the georeferenced data, need to build the model step by step, starting with defining the main geometries. This is followed by the adjustment of the lanes, and the paintings, stencils, street signs and guardrails. As a final step, the altitude data are adjusted to the actual test track, so that we can accurately match the geometries in both the lateral and longitudinal directions. The resulting model can be exported to different file types and transferred to automotive simulation tools (e.g. IPG Carmaker, VTD Vires, dSpace ASM)



Fig. 4 Road model of ZalaZONE ADAS surface in RoadRunner software

All models introduced above are freely available at the website of BME Automated Drive Lab: <https://www.automateddrive.bme.hu/downloads>

4 Description of static and dynamic object - ASAM Standards

In recent years, with the development of the automotive industry, there has been a growing need for a uniform description of the road network. The ASAM OpenDRIVE format provides a common base for describing road networks using the xodr file extension. The data that is stored in an ASAM OpenDRIVE file describes the geometry of roads, lanes and objects, such as roadmarks on the road and signals. The main purpose of ASAM OpenDRIVE is to provide a road network description that can be imported into simulations to develop and validate ADAS functions. With the ASAM OpenDRIVE, these road network descriptions can be exchanged between different simulation toolchains. Providing a standardized format for road descriptions reduce the cost of creating and converting files for their development and testing purposes [3].

4.1 Relation to other standards

The ASAM OpenDRIVE description format contains all static objects on the road. To create the complete environment, additional description formats for static 3D objects, like trees and buildings, are needed. Road surface profiles are included from the ASAM OpenCRG file format. The dynamic content of driving simulations can be described with ASAM OpenSCENARIO. The three standards complement each other and cover the static and dynamic content of in-the-loop vehicle simulation applications [3].

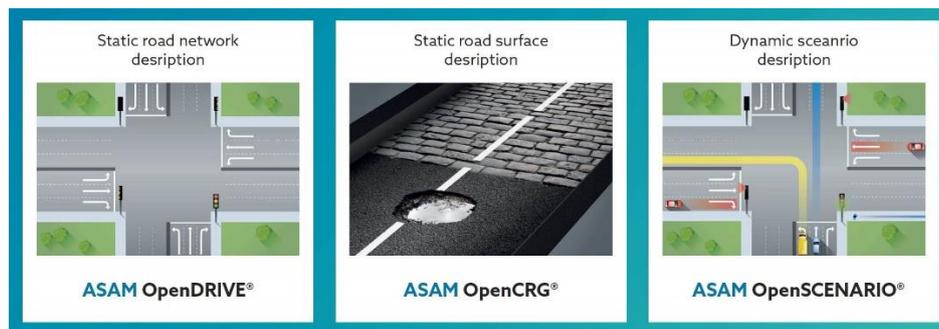


Fig. 5 ASAM Standards for road and scenario description [3]

5 State-of-the-art simulation tools at ZalaZONE

Thanks to the OpenDRIVE format, we have the option of importing test track modules into various simulation tools, but there are limitations how they can handle OpenDRIVE files. The complexity of roads, geometries and the junctions can be influencing factors. In general, software vendors are constantly develop and improve OpenDRIVE importing ability.

In ZalaZONE, currently, three state-of-the-art vehicle simulation toolchains are in daily use. In the next section, three short summaries are provided, which are daily used at ZalaZONE [4].

5.1 Siemens Prescan 2019

Prescan is a vehicle dynamic and ADAS development tool. It is mainly used for test and simulate sensors, because of the huge variety of sensor models in it's library. It is also integrated into MATLAB/Simulink. Prescan also works well with other simulation tools like IPG Carmaker though so-called Co-simulation [4]. OpenDRIVE import is available, but can not modify the roads and segments.

5.2 IPG Carmaker 9.0

IPG Carmaker is a simulation tool from IPG Automotive GmbH. Main advantage of this tool is vehicle dynamic models. Carmaker is also integrated into Matlab/Simulink, so testers can trigger event or evaluate measurements in widely popular Matlab environment [4]. Import available with OpenDRIVE 1.4 and 1.5 version. Complex junctions, road superelevation and roads with a lot curves can cause faulty reading.

5.3 Virtual Test Drive Vires 2021

One of the most known simulation tool. Vires has ability to perform HiL, SciL, MiL tests. The advantage of this toolbox is ability to properly use of OpenDrive, OpenCRG and OpenSCENARIO files. Vires was also one of the establishers of OpenX project (OpenDRIVE, OpenCRG, OpenSCENARIO) in the early 2000 years [4]. This tool reads OpenDRIVE really well, based on the fact they are among the developers.

6 Conclusion

We will continue to build ZalaZONE's virtual models as construction is completed. Based on the feedback, this is also useful for our industry and research partners. This process will become more efficient, and with the development of OpenDRIVE and related simulation tools, more and more people will be able to use the models. Our goal is to create a complete ZalaZONE model that includes all track elements and can be integrated into leading automotive software with high level of accuracy.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Zs. Szalay, T. Tettamanti, et al., Development of a Test Track for Driverless Cars: Vehicle Design, Track Configuration, and Liability Considerations, *Periodica Polytechnica Transportation Engineering*, 46 (1) (2018), pp. 29-35. doi: <https://doi.org/10.3311/PPtr.10753>
- [2] Zs. Szalay, Á. Nyerges, Z. Hamar, M. Hesz, Technical Specification Methodology for an Automotive Proving Ground Dedicated to Connected and Automated Vehicles, *Periodica Polytechnica Transportation Engineering*, 45 (3) (2017), pp. 168-174. doi: <https://doi.org/10.3311/PPtr.10708>
- [3] Road and scenario description - <https://www.asam.net>
- [4] Gangel, K., Hamar, Z., Hány, A., Horváth, Áron, Jandó, G., Könyves, B., Panker, D., Pintér, K., Pataki, M., Szalai, M., Szalay, Z., Tettamanti, T., Tihanyi, V., Tóth, B., Varga, B., & Viharos, Z. J. (2021). Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. *Acta Technica Jaurinensis*, 14(4), 488–507. <https://doi.org/10.14513/actatechjaur.00606>

Test Automation – From Virtual Scenario Creation to Real-world Testing

Áron Horváth¹, Tamás Tettamanti^{2a}

¹ Automotive Proving Ground Zala Ltd.

² Department of Control for Transportation and Vehicle Systems
Faculty of Transportation Engineering and Vehicle Engineering
Budapest University of Technology and Economics

^a Corresponding author: tettamanti.tamas@kjk.bme.hu

Abstract

As advanced driver assistance systems (ADAS) are continuously upgrading, the testing standards, descriptions are evolving as well. The border disappears more and more between simulation and real-world testing. For more accurate repeatable scenarios we need a new method, which helps to make faster the process of scenario creation. The result for this would be the connection of simulated scenarios and real test equipment. Our result includes several components such as HD maps, industrial simulation software, OpenSCENARIO compatibility, MATLAB programming, AB Dynamics software and test equipment. With HD maps and OpenSCENARIO compatible simulation software we can create GPS accurate scenarios. Then, the own developed MATLAB program can convert the OpenSCENARIO file into a new format, which is readable by AB Dynamics software. After this process the last step is the real-world testing with AB Dynamics test equipment.

Keywords: AB Dynamics software and test equipment, HD maps, MATLAB, OpenSCENARIO, Simulation software

1 Introduction

The continuous development in the production of self-driving vehicles and ADAS driving assistance systems is also creating an increasing demand in the field of vehicle testing. The newly built ZalaZONE Automotive Test Track and the Budapest University of Technology, working closely with it, provide an excellent opportunity for this [1].



Fig. 1 ZalaZONE Proving Ground

Not only the self-driving and ADAS functions but also the general vehicle dynamics tests are possible on the test track. These measurements require a lot of preparation, even several hours of parameterization. Additional unforeseen problems may occur on the test surface. The development is intended to make this test preparation, repetitions, and quick

modifications smoother. The way to do this is to simulate test cases and transfer them to a real test environment using AB Dynamics test tools. A conversion program has been created for this purpose, linking the simulated and real environments [1].

2 ZalaZONE Proving Ground HD Maps

The implementation of the scenarios in a simulation way is based on cm-accurate orbital models and HD maps. It is a basic requirement that the position of the simulated vehicle is as close as possible to the position of the real vehicle. To achieve this, the modules of the real test track were measured up by laser scanning. From this, a point cloud was created, on which a 3D model could already be developed, from which the OpenDRIVE format corresponding to the needs of the simulation software could be exported. This trajectory modeling activity was performed using RoadRunner software [2].

The models are freely available at the website of BME Automated Drive Lab:
<https://www.automateddrive.bme.hu/downloads>.

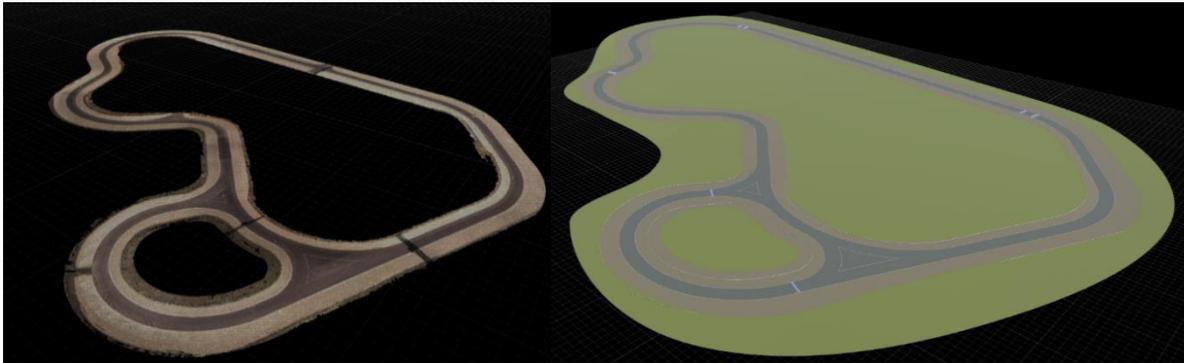


Fig. 2 ZalaZONE Handling Course Point Cloud (left), 3D Model (right)

3 Scenario creation in simulation software

The software used is VTD Vires, which not only handles OpenDRIVE files but also supports the creation of OpenSCENARIO files. Adapted to the developed conversion mode, the parameterization of a scenario is performed by the following steps:

1. Open the Scenario Editor, where you can load the previously created OpenDRIVE track model.
2. Then come the placement of the vehicles participating in the scenario.
3. Then the most important step is to add routes and assign them to a specific participant.
4. There are several options for adding a route:
 - a. You can also freehand add points that the software connects with straight lines, this is the polyline setting. The position of the recorded points can be changed later.
 - b. Another option is to use a spline.
 - c. For accurate route planning for NCAP tests, however, the clothoid setting is used. This creates the most accurate route and is also the most suitable for subsequent data processing and conversion.
5. You can then set how the vehicle relates to the route:
 - a. Drive through it in self-driving mode.
 - b. Walk the route by performing various actions. (This is our preferred setting.) Actions can be used to divide a route into sections according to specific accelerations and target speeds.
6. The desired test vehicle must be placed on the routes with actions (min. 4).

Finally, the finished Scenario must be exported in OpenSCENARIO format.

A top view of a Cut-out Scenario with 2 participants + 1 static target created in the scenario editor is shown in the following figure: [4]

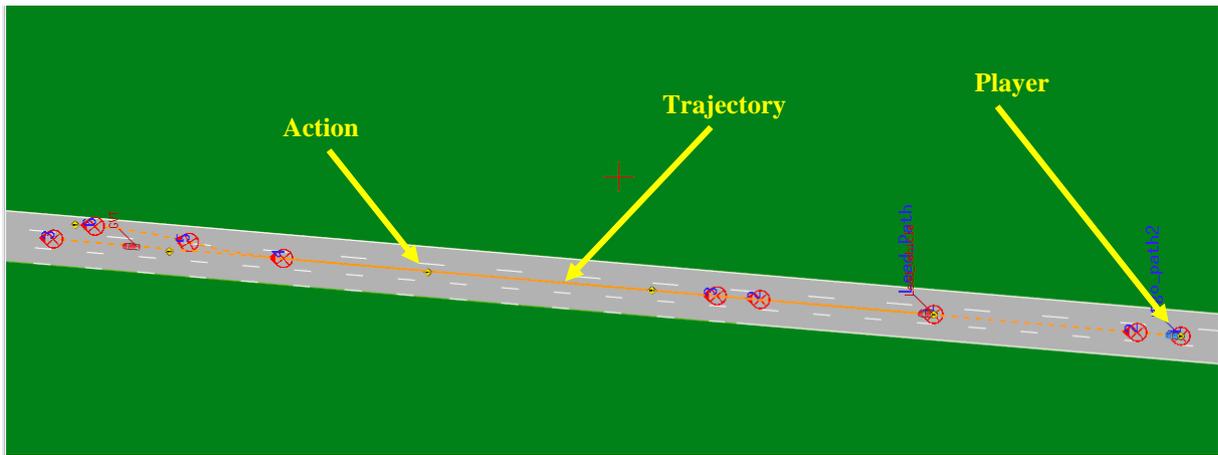


Fig. 3 View of OpenSCENARIO in editor

4 OpenSCENARIO Converter

The compiled and exported OpenSCENARIO file must be converted to readable file for ABDynamics tools. To accomplish this, we have developed a converter in MATLAB software environment that converts files into a structure that can be directly processed by AB Dynamics test tools software.

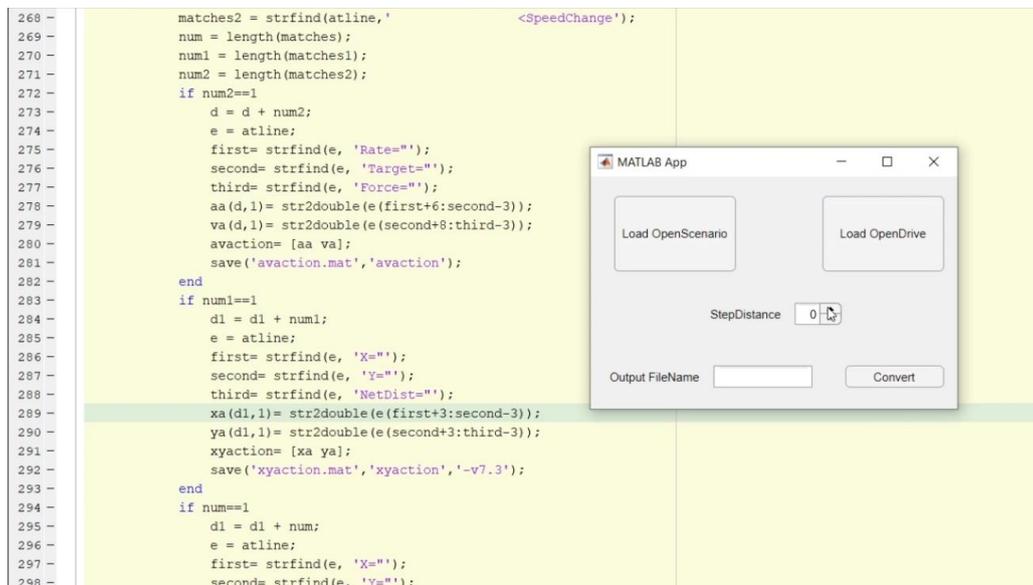


Fig. 4 GUI of OpenScenario Converter

The initial files for the conversion are OpenSCENARIO with XML extension and OpenDRIVE with XODR extension. Based on the XODR, it determines the center of the orbital relative coordinate system from Latitude, Longitude GPS data. The XY coordinate data collected from the XML is positioned accordingly. The converter retrieves the coordinates from the text data, then converts them into a usable number format and compiles them. It needs to decide whether it is one or more routes. Depending on this, one or the other way of the program continues to work. The two ways go through almost similar tasks, but for multiple routes, it runs until you have done the calculations for each route. In a multi-participant scenario, the starting position of each route must be shifted relative to the center of the common coordinate system. To make this happen: the starting point of the first participant gives the center of the coordinate system, and the others are positioned accordingly. An essential element of the converter is the use of the speed profile from OpenSCENARIO to create as accurate a run as possible in time and space compared to the predefined scenario. Speed and acceleration data are defined in maneuvers, along with their XY coordinates. The program also collects this data and compares it with the routes to determine which sections are subject to what acceleration and target speed. Based on the information already available, it calculates the main data for each route, which are Distance, Time, XY coordinates, and curvature. After the main data, the additional ABDynamics file information is also printed and then saved as a file with a PMC extension per path. This file format can be read by ABDynamics software.

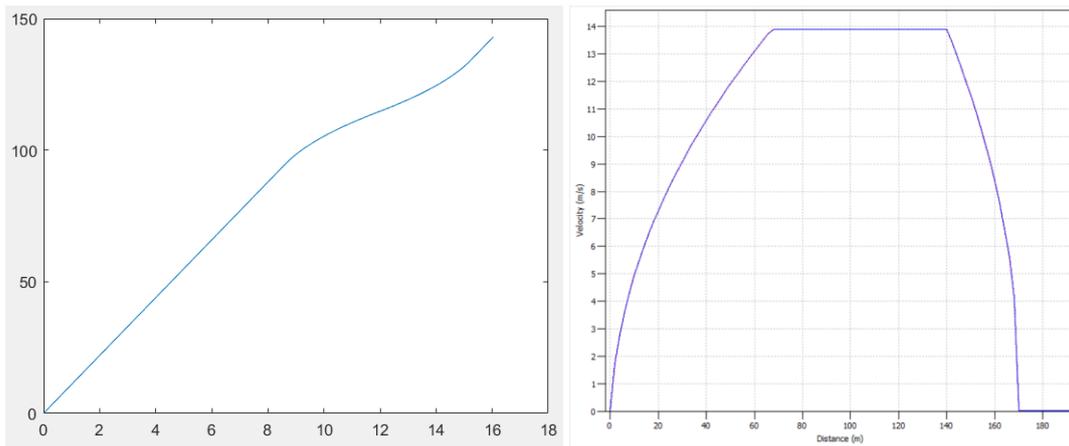


Fig. 5 Converted Clothoid Trajectory and Speed Profile in MATLAB

5 AB Dynamics tools preparation

The PMC files can even be edited by the AB Dynamics software. The speed profile can be checked and minor errors can be corrected manually here. The center of the coordinate system, which calculated during the conversion, can be specified together with the underlay, positioned according to GPS. All the routes belonging to the scenario are loadable into the common coordinate system. On these a preliminary offline Collision Check can be run, so as to see the time course of the participants on the track in a simple visualization. If this is the same as the simulation you created earlier, the user can go for a test performed with real test tools. This data can also be loaded for vehicles with steering and pedal robots and platforms carrying dummy puppets, so even a complex scenario can be implemented. The whole process of pre-building the scenario in a simulation and converting it with a few clicks using a converter greatly speeds up the compilation of real tests. The design phase can take place immediately in the simulation. It is not necessary to structure the route separately in each section in AB Dynamics software for each participant. There are no pre-measurements to set the coordinate system, all of which can be specified in advance and new files can be generated quickly even after any modifications.



Fig. 5 Compare of the trajectory in editor and in the AB Dynamics software

6 Conclusion and future work

According to the experiments of real tests, the development makes easier the whole testing process. The GPS position is accurate and the speed of the vehicles is appropriate, but not exactly the same as the simulated.

The working environment is demonstrated here: <https://www.youtube.com/watch?v=6X10POWy3uY>

As future work, we need to develop the converter's speed profile counter and we have a lot of other potential. We can try to cover the OpenSCENARIO's maneuvers, like lane change, stop, backward or the movements of the autonomous mode of the Vires Simulation. The final goal is the conversion of any kind of OpenSCENARIO file.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] <https://zalazone.hu/>
- [2] <https://www.automateddrive.bme.hu/downloads>.
- [3] K. Gangel, Z. Hamar, A. Hary, . Horvath, G. Jando, B. Konyves, D. Panker, K. Pinter, M. Pataki, M. Szalai, Zs. Szalay, T. Tettamanti, V. Tihanyi, B. Toth, B. Varga, Zs. J. Viharos (2021). Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires
- [4] Marius Dupuis (2017). VTD VIRES Virtual Test Drive
- [5] https://higherlogicdownload.s3.amazonaws.com/AUVSI/14c12c18-fde1-4c1d-8548-035ad166c766/UploadedImages/2017/PDFs/Proceedings/ESS/Wednesday%201630-1700_Marius%20Dupuis.pdf

Initiation and Stabilization of Drifting Motion of a Self-driving Vehicle with a Reinforcement Learning Agent

Szilárd Hunor Tóth^{1,2}, Ádám Bárdos^{1a}, Zsolt János Viharos^{2,3}

¹ *Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics*

² *Research Laboratory on Engineering and Management Intelligence, Intelligent Processes Research Group, Institute of Computer Science and Control*

³ *Department of Management and Business Law, Faculty of Economics, John von Neumann University*

^a *Corresponding author: bardos.adam@kjk.bme.hu*

Abstract

Performing special driving techniques like drifting can be challenging even for professional human drivers. However, such maneuvers can be essential for avoiding accidents in critical road scenarios like evasive maneuvers. This paper reports novel research results whose main goal is to develop a self-driving agent for drift motion control based on vehicle simulation in MATLAB/Simulink. The state representation of the vehicle includes the longitudinal and lateral velocities with the yaw rate. The agent action space consists of two actuators: the throttle position and the roadwheel angle. The goal of the agent is twofold: first, it needs to jump into a drifting state; second, it has to keep the vehicle in drift. The simulation results show that the proposed RL agent is capable of learning to approach a predetermined drift equilibrium from cornering and staying in this drift situation as well. For the training, the solution excluded using any prior data. It only works with information gained from the simulation model, which is a remarkable difference from the actual state-of-the-art RL-based solutions.

Keywords: *reinforcement learning, vehicle drifting, vehicle motion control*

1 Introduction

The research and development of the autonomous vehicles has increased in pace and precision in the recent years [1]. In the field of vehicle control, more and more challenges have arisen regarding lateral vehicle control, including motion control at handling limits.

Drifting is a kind of cornering motion, where the driver is constantly counter-steering to maintain a high side-slip angle, which most ordinary drivers are unable to control, and usually leads to the vehicle spinning uncontrollably. This maneuver is mostly seen in motorsports (for example in rally), but the application of this motion has notable potential on the public roads as well. According to the GES (General Estimates System) crash reports of 2013, a study [2] showed that the relative frequency of control loss-related accidents is around 8.32%. Although, there is a potential possibility to decrease this significant value by using self-driving vehicles which can maintain and adjust the motions of the vehicle when it becomes unstable. In addition, the increased value of automated vehicles in motorsports [3] and various potential entertainment purposes (like implementing a “drift button” for hobby drivers) also grants a motivation for research.

As for previous examples and results for the implementation of self-drifting, using linear control methods seem to work successfully for steady-state drift problems in simulation [4] [5] and also in real-life applications [6]. The simulation application of the MPC (Model Predictive Control) controller for drift stabilization tasks has also been successful so far [7]. In each of these cases, a car with a high rear traction force considered to be essential for achieving satisfying performance, based on real-life observations and measurements data.

The idea of using reinforcement learning (RL) methods to solve self-drifting problems promise better generalization abilities in continuously changing driving situations than the previously mentioned control methods. In paper [8] a model-based policy search algorithm was used to solve a steady-state problem with good results,

validated on a radio-controlled car. In the work of [9] and [10] the goal was to achieve high side slip angles at high speeds while following a pre-defined trajectory using actor-critic algorithms, the result showing good generalization abilities on previously unseen trajectories. Although, all of these research achievements incorporated some kind of prior knowledge when training the agent. These indicate the challenge of how to apply reinforcement learning without any preliminary setup or knowledge to control and initiate drifting. In addition, previous work [11] proves the potential of applying RL for automatization problems.

In this paper, novel results on reinforcement learning aided autonomous drift are introduced. The task is the initiation and stabilization of a steady-state drift without using any kind of prior knowledge for training the operating RL agent. A single-track dynamic vehicle model was implemented in MATLAB/Simulink, and a Soft Actor-Critic (SAC) [12] algorithm for training was designed. The target drift state was calculated by solving a system of equilibrium equations for the vehicle model. Also, a particular drift indicator was defined for identifying vehicle drifting.

The next section describes the drift state of vehicles established by equation-based solutions while describes the applied vehicle simulation model. The subsequent section presents the coupled reinforcement learning method. After the section about the experiments and results, conclusions and references close the paper.

2 Vehicle Modelling

The basic definition of drift with the state-of-the-art solutions of finding drift equilibriums based on theoretical models (equations), and the integrated vehicle model are described in the following paragraphs.

2.1 The Drifting Motion

To initiate a drift, the driver needs to apply high enough torque input for the rear wheels to increase the rear tire slip angle, so the rear end of the car can “drift” off the arc of the corner. This is easier to achieve using rear-wheel-drive vehicles because they can directly increase the longitudinal forces applied to the rear wheels and saturate them. To control the now unstable car, the driver needs to counter-steer to compensate for the high rear-slip angle, so the car can stay on the corner’s arc, otherwise the vehicle would spin out.

A method for identifying drifting operation points is called the equation-based (or model-based) solution which involve using the vehicle model to describe the drift equilibrium points. Previous work [13] show this can be done by solving a system of algebraic equilibrium equations, based on the Newtonian laws of motion. This calculation can be seen in detail in the aforementioned works.

2.2 Dynamic Vehicle Model

The model used for the simulations presented in this research is a single-track dynamic model with two different tire models for the front (steered) and the rear (driven) wheel. The advantage of this model is its simplicity: it ignores the roll dynamics and aerodynamics, which are far less important than the tire model for accurately representing the motions during drifting [14].

The most important equations for the model are the longitudinal (1), lateral (2), and yaw (3) motions of the vehicle’s body frame [14]:

$$\dot{v}_x = \frac{1}{m} F_x + r v_y \quad (1)$$

$$\dot{v}_y = \frac{1}{m} F_y - r v_x \quad (2)$$

$$\dot{r} = \frac{1}{I_z} M_z \quad (3)$$

These are the derivatives of the v_x the longitudinal velocity, the v_y lateral velocity and the r yaw rate. For the detailed explanation of the force components in the above equations, see [14].

The model of the front tires is an analytic hybrid model based on the brush tire model, which is purely a lateral slip model [15]. Because this model has a rear-wheel drive, longitudinal forces are only needed to consider in the case of the rear wheels, which would also generate longitudinal slip, so a combined slip tire model is needed. This means that consideration of wheel speed dynamics is also required if the brush tire model is being used. However, the longitudinal rear tire force (F_{x_r}) is handled here as an input, a simpler approach can be used based on the friction circle approximation as proposed in [15]. For the formulas, please see the referenced paper.

3 Drifting with Reinforcement Learning

The description of proposed soft actor-critic reinforcement learning algorithm and the structure of the proposed RL problem are described in this section.

3.1 Soft Actor-Critic Method

The learning process of the RL algorithms is based on the interaction between an agent and an environment. The agent takes an action based on the current state of the environment and receives a reward signal which informs the agent on the effectiveness of the selected action. The agent's mission is to maximize the expected value of the next reward signal with exploring which actions give the maximum cumulative reward in the various states [16].

Actor-Critic methods are state-of-the-art RL algorithms which usually use neural networks to operate within a continuous environment. The actor is responsible for deciding on the actions while the critic estimates the values of the states and produces a critique to update both networks, so they can maximize the cumulative expected reward. The Soft Actor-Critic (SAC) algorithm [12] operates with two separate critic networks to maximize the effectiveness of value approximation and uses a stochastic actor to ensure the exploration, which has an adaptive feature as well. These abilities made this method the most ideal solution above the other state-of-the-art algorithms to solve a complex continuous control problem like autonomous drifting.

3.2 RL Agent Architecture

To represent the control of drifting as a reinforcement learning problem, it's needed to define the environment's state and action spaces and the reward signal.

The velocities defined in (1), (2) and (3) are enough to obviously determine the state of the vehicle, so the continuous state space was chosen to be $\mathcal{S} = (v_x, v_y, r)$.

For the agent's actions, drift is achievable by just adjusting the longitudinal force applied on the rear wheel through the pedal input ped_{acc} and the front wheel angle δ by changing the steering wheel angle δ_{steer} . So, the continuous action space is $\mathcal{A} = (ped_{acc}, \delta_{steer})$. These values are defined to be in the intervals $ped_{acc} \in [0,1]$ and $\delta_{steer} \in [100^\circ, -200^\circ]$, meaning $ped_{acc} = 1$ is full throttle, and $\delta_{steer} > 0$ is the left-hand side domain of the steering wheel.

By defining $v_x = 10 \text{ m/s}$ and $\delta = -10^\circ$ for this problem (which means a given speed and a fixed steering wheel position), based on the solution method described in [13] the equilibrium point received for the target drift state is $S_{drift} = (10 \text{ m/s}, -3.48 \text{ m/s}, 0.8334 \text{ rad})$. The goal of the agent is to reach and maintain this state, which is a point in the state space. The defined reward function (4) is the negative of the relative Euclidian distance of the state vector from the target drift state, so

$$r(S_t) = -\frac{1}{3} \sum_{i=1}^3 \left(\frac{S_{t_i}}{S_{drift_i}} - 1 \right)^2. \quad (4)$$

4 Experiments and Results

To validate the implemented agent and the Simulink vehicle environment, the following preliminary task was defined: after starting the simulation from the target drift state, the agent's designated task is to stay in a very narrow proximity of the drift equilibrium, and with that, it's able to control the pre-initialized drift. After completing the above-mentioned test successfully, experiments were made to see if it can learn how to approach the target drift state then stay close to it.

To identify if the vehicle satisfies the required drift conditions, the following indicator function (5) was created:

$$Isdrift(S_t) = \begin{cases} 1, & \text{if } \frac{S_{t_i} - S_{drift_i}}{S_{drift_i}} < 0.1 \quad \forall i \in \{1,2,3\} \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

meaning it returns 1 if each state variable is in a close relative distance to the target variable at the same time, and 0 otherwise. The training was done by using episodes, which are start from an initial state and lets the agent operate until some simulation termination time T . In the case of the preliminary tests the initial state was set to be the target drift state ($S_0 = S_{drift}$), and for the following experiments the $S_0 = (9 \text{ m/s}, 0.825 \text{ m/s}, 0.8334 \text{ rad})$ starting point was defined (a medium-speed left cornering non-drift equilibrium).

For first, $T = 5\text{s}$ termination time was set which is a little more than the required amount of time for the vehicle to enter drifting from the initial state under a perfect action selection policy. The SAC agent was successfully

trained to accomplish reaching the drift state and to stay in it until the end of the episode, although when the episode length $T > 5$ was set, it was found the agent loses control over the drift after a while. The found solution for this problem was to use a consecutive training technique where the training session was repeated on the same agent with extending the duration of the episodes every time by a small step. After reaching the setup $T = 10s$ using this method, the agent improved significantly enough to hold the target drift for even long-drawn-out episodes ($T \geq 120s$).

The analysis and the identification of desired drifting was done by monitoring the *Isdrift* indicator (5). On Fig. 1, a scope of a 15-second-long simulation is shown with the vehicle's motion trajectory pictured on Fig. 2. It can be seen that the indicator only "lights up" after approx. 8 seconds of simulation, though drifting occurs much before around 2 seconds. While the role of the indicator was to capture when the vehicle is close to the target, these observations might suggest defining a new or an additional indicator in the future for more effective training and analysis.

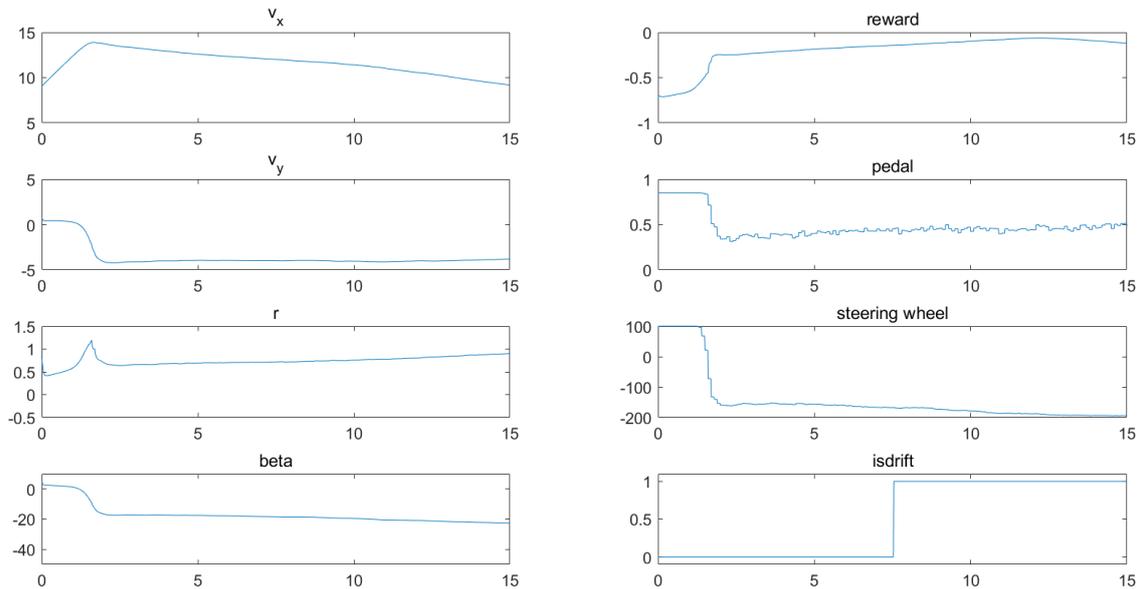


Fig. 1 Scope graph of a simulation showing 15 seconds of the agent's operation. In addition to looking at the indicator (bottom right), the state variables (left side) and the reward signal (top right) also shows the good performance of the agent, which has chosen actions similarly as a human driver would in practice.

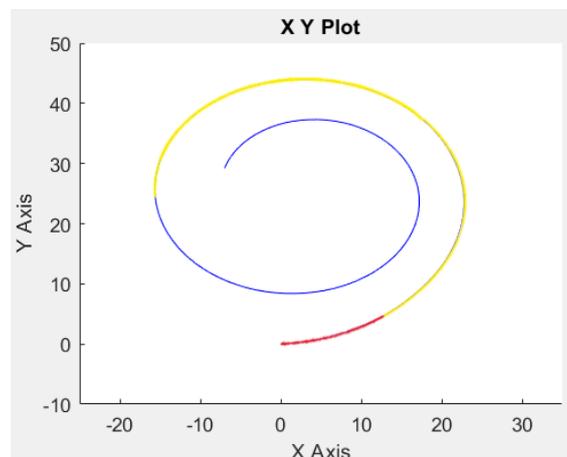


Fig. 2 Trajectory of the simulation shown on Fig. 1 The colors describe the stages of drifting: red means no drift, yellow shows drifting, which changes to blue when *Isdrift* = 1. It can be observed that the trajectory's arc bends differently in each of these sections.

5 Conclusion

The paper presented a reinforcement learning agent trained on a single-track vehicle model to approach and maintain a target drift state in a MATLAB/Simulink simulation environment. The results and the analysis of the

defined drift indicator function indicate that the agent is capable of learning to approach a target drift equilibrium point from various initial states. The next steps of this research include evaluating more RL algorithms on this problem (e.g., discrete Q-learning, DQN) to find the most beneficial solution in practice. In the future, the goal is to test this approach in real conditions on a commercial test car on the ZalaZONE proving ground preceded by a validation on the freely available ZalaZONE simulated environment [17]. Also, this research will set its focus on more complex problems and situations later, like adding noise to the environment (e.g., varying road surfaces) and performing more challenging maneuvers in form of trajectory following problems.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Di, X., & Shi, R. (2021). A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transportation research part C: emerging technologies*, 125, 103008.
- [2] Li, T., & Kockelman, K. M. (2016, January). Valuing the safety benefits of connected and automated vehicle technologies. In *Transportation Research Board 95th Annual Meeting (Vol. 1)*.
- [3] Yang, F. (2021, June). Research on the course, form and strategy of autonomous driving competition. In *Journal of Physics: Conference Series (Vol. 1948, No. 1, p. 012093)*. IOP Publishing.
- [4] Bárdos, Á., Domina, Á., Szalay, Z., Tihanyi, V., & Palkovics, L. (2019, November). MIMO controller design for stabilizing vehicle drifting. In *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)* (pp. 000187-000192). IEEE.
- [5] Velenis, E., Katzourakis, D., Frazzoli, E., Tsiotras, P., & Happee, R. (2011). Steady-state drifting stabilization of RWD vehicles. *Control Engineering Practice*, 19(11), 1363-1376.
- [6] Bárdos, Á., Domina, Á., Tihanyi, V., Szalay, Z., & Palkovics, L. (2020, October). Implementation and experimental evaluation of a MIMO drifting controller on a test vehicle. In *2020 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1472-1478). IEEE.
- [7] Czibere, S., Domina, Á., Bárdos, Á., & Szalay, Z. (2021). Model Predictive Controller Design for Vehicle Motion Control at Handling Limits in Multiple Equilibria on Varying Road Surfaces. *Energies*, 14(20), 6667.
- [8] Cutler, M., & How, J. P. (2016, May). Autonomous drifting using simulation-aided reinforcement learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5442-5448). IEEE.
- [9] Cai, P., Mei, X., Tai, L., Sun, Y., & Liu, M. (2020). High-speed autonomous drifting with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2), 1247-1254.
- [10] Orgován, L., Bécsi, T., & Aradi, S. (2021). Autonomous Drifting Using Reinforcement Learning. *Periodica Polytechnica Transportation Engineering*, 49(3), 292-300.
- [11] Viharos Z. J.; Jakab, R. B. (2021). Reinforcement Learning for Statistical Process Control in Manufacturing. *Measurement*, Vol. 182.
- [12] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... & Levine, S. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- [13] Hindiyeh, R. Y., & Gerdes, J. C. (2009, January). Equilibrium analysis of drifting vehicles for control design. In *Dynamic Systems and Control Conference (Vol. 48920, pp. 181-188)*.
- [14] Jazar, R. N. (2019). Pp. 115-161. *Advanced vehicle dynamics*. Springer.
- [15] Hindiyeh, R. Y. (2013). *Dynamics and control of drifting in automobiles (Doctoral dissertation, Stanford University)*. Pp. 20-28.
- [16] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [17] Gangel, K.; Hamar, Z.; Hány, A.; Horváth, Á.; Jandó, G.; Könyves, B.; Panker, D.; Pintér, K.; Pataki, M.; Szalai, M.; Szalay, Zs.; Tettamanti, T.; Tihanyi, V. R.; Tóth, B.; Varga, B.; Viharos, Zs. J. (2021). Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. *Acta Technica Jaurinensis*, 14(4), 488-507.

Development of a GNSS Based High Accuracy Measurement System to Support Vehicle Dynamics Testing

Árpád Fehér^{1a}, Gábor Kardos¹, Ádám Szabó¹, Szilárd Aradi¹

¹ Department of Control for Transportation and Vehicle Systems,
Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics

^a Corresponding author: feher.arpad@kjk.bme.hu

Abstract

The article presents the development of a low-budget positioning device that aims to provide an alternative in self-driving vehicle development research that could replace costly, commercially available devices. In addition to being financially advantageous, it has the added benefit of allowing students to be involved in development. The primary function of the device is the sensor fusion, which outputs position, velocity, and orientation estimation based on data provided by Real-Time Kinematic (RTK) Global Navigation Satellite System (GNSS) technology and an Inertial Measurement Unit (IMU). High-frequency estimates are generated by running an Extended Kalman Filter (EKF) on a microcontroller in an embedded environment. During the work, new challenges arose several times that required solutions. For example, delays due to the operation of GNSS receivers, which the estimation algorithm must compensate, and proper calibration of the sensors for the measurement vehicle. In addition to the software, the development of the tool includes the complete design, manufacture, and testing of the hardware, which allows testing the completed software units not only in a simulation but also in a real environment. During testing, the output of the developed device was compared several times with commercially available hardware for similar purposes.

Keywords: GNSS, IMU, Kalman filter, low-cost, positioning, RTK

1 Introduction

Nowadays, with the growing interest in autonomous vehicles and the ever-increasing number of new developments, there has been a need for high-precision measurement of the dynamic condition of cars. This may include monitoring the current position, accelerations, velocities. Many products on the market already meet these requirements, but their usage can be an exclusionary reason in many cases due to high prices. There are solutions for applying smartphones for similar topics [1], which can spare the custom hardware design and production cost. The disadvantage, which excludes this solution, is the inaccuracy of GNSS positioning.

In [2], the proposed algorithm is an Extended Kalman Filter with 15 states, which also includes errors regarding the inertial sensor and GNSS data to maximize the prediction accuracy. The publication also mentions the use of a vehicle wheel odometer which can be a viable option for our case in the future.

In [3], the authors approached the problem in a different way. There are tightly coupled techniques to integrate GNSS and INS sensors to provide more accurate information. In this case, the algorithm is based on the raw GNSS measurements, for example, pseudorange and Doppler observables. This approach offers better accuracy in poor signal and limited coverage scenarios.

Even if the GNSS data are integrated with other information like INS or odometry, the measurements can be inaccurate in urban canyons surrounded by high buildings. In [4] a unique method is proposed, which uses a fish-eye lens camera to observe the visibility of the sky and the satellites. Based on this information, they can decide the weighting factor of the used sensors during prediction.

The purpose of this publication is to illustrate the hardware and software architecture of a device capable of producing high-precision, high-frequency position, velocity, and orientation data. The main goal is to use less financial material than its commercial counterparts with loosely-coupled GNSS and INS sensor integration.

2 RTK GNSS device

2.1 Hardware

The hardware comprises several different modules to implement the desired functions. It includes two u-blox GNSS receivers, an IMU sensor, a microcontroller, LEDs, and various communication interfaces like CAN and USB ports. For said hardware elements to work together as a unit, designing a printed circuit that implements the expected circuit connections between the components is necessary.

The central microcontroller is a 32-bit ARM Cortex-M7 controller with a maximum clock speed of 300 MHz, providing sufficient resources to receive, process, and communicate sensor data. In addition to the essential functions, it has several interfaces, such as UART ports, SPI, I2C, or CAN communication capabilities typical of automotive controllers. This way, sensor data can be read out at high frequencies and transmit raw and calculated data to other external devices simultaneously.

Two u-blox GNSS receivers on the hardware provide accurate position, speed, and orientation information. They support RTK technology, meaning they can produce data with up to 2 centimeters accuracy. It requires correction data provided by an external base station. A single receiver would be sufficient to determine position and speed, but a second device is necessary to determine high-precision orientation even in a stationary position. In this case, the heading is calculated from the relative position of the two antennas. The GNSS device of choice supports this feature, so it does not need to be implemented separately on the microcontroller.

For the device to estimate the required states of a vehicle, a Bosch BMI085 IMU sensor has also been placed on the circuit. It is a 6-degree-of-freedom sensor that includes a 3-axis gyroscope and a 3-axis accelerometer. The sensor can provide data with a maximum frequency of up to 2000 Hz, depending on the built-in filter configuration. The microcontroller communicates with the sensor via an SPI interface.

A fundamental requirement for a device intended for an in-vehicle environment is to communicate with other ECUs via the CAN network so that they can properly access the data it provides. The hardware is able to send some basic information about its status and, of course, publish raw GNSS and IMU data, as well as Kalman filter estimates with a 100 Hz update.



Fig. 1 The GNSS module (left) and the GSM module (right) hardware.

2.2 Software and challenges

The primary function is to predict high-frequency speed, position, and orientation data. The sensor data processing required for this takes place on a microcontroller with ARM Cortex M7 architecture.

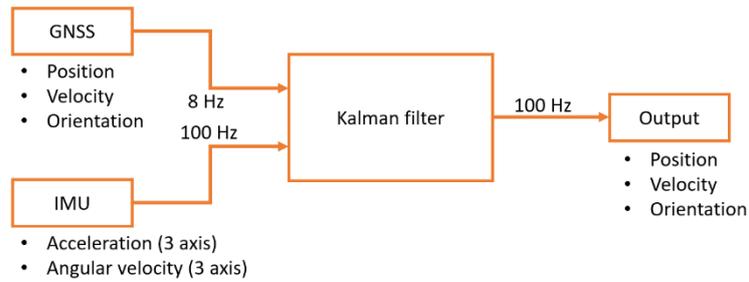


Fig. 2 Sketch of the prediction data flow

Fig. 2 also shows the calculation is done by an EKF [5], which can determine the device's exact position, speed, and orientation with knowledge of the high-precision 8 Hz GNSS and 100 Hz IMU data.

The state-space consists of 10 parameters (1). The first four are for the orientation, represented in quaternion form. The next three are the latitude, longitude, and height for the position, and the last three are the velocity in NED coordinate system. In the future, it can be expanded by new ones if needed.

$$[q_0 \ q_1 \ q_2 \ q_3 \ posN \ posE \ posD \ velN \ velE \ velD] \quad (1)$$

There have been several challenges during the development that significantly affect the accuracy of the estimates. One such factor is the fundamental property of GNSS receivers. Calculation of the centimeter-accurate information with RTK correction takes time, ranging from 50 to 100 milliseconds based on tests. Due to this delay, the position delay can be multiples of 10 centimeters in a high-speed scenario. Based on the outputs of the GNSS receivers, this delay is measurable, so by modifying the Kalman filter accordingly, this can be compensated. Fig. 3 shows the estimator already gives an accurate result.

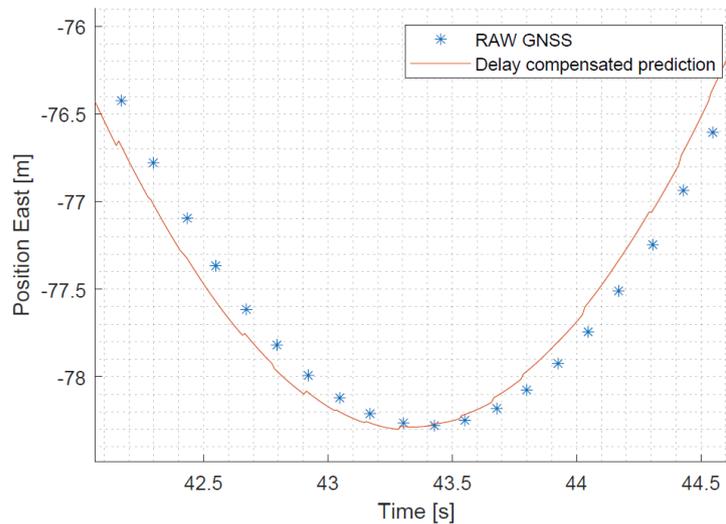


Fig. 3 Raw 8 Hz GNSS position (with blue) and delay compensated predicted position based on GNSS and IMU (with orange)

The prediction accuracy also depends on the calibration of the IMU, which has two types. The so-called static calibration is required to determine and compensate the accelerometer and gyroscope biases and scale errors. The other is the relative orientation of the IMU sensor frame to the vehicle chassis frame, requiring rotational sensor data transformation. Most of these parameters can be calculated by a PC application, which is also under development.

One basis for the operation of the EKF is to use lower frequency but more accurate GNSS data to clarify the error of the data estimated from the IMU. The orientation around the vertical axis is determined from the positions given by the two antennas of the GNSS modules, but the exact values of pitch and roll are unknown. The calculation of these can be examined with the help of the IMU gyroscope, but this estimate may become more and

more inaccurate over time due to the noise and inaccuracy of the sensor. The development to solve this issue is currently running and being tested.

3 Correction module

RTK GNSS correction requires correction data from an external base station with a well-known location. The information can be transmitted through a wireless internet connection or radio communication. In this case, the internet connection is the appropriate way because there are several online services available nowadays. Furthermore, the Faculty of Civil Engineering is operating and testing GNSS base stations at the Budapest University of technology and Economics and on the ZalaZONE Automotive Proving Ground.

The hardware for this task is a separate device from the GNSS module. A Quectel LTE modem is responsible for the mobile internet connection, which later can be replaced by a 5G compatible one. An 8-bit microcontroller performs the configuration of the modem and the data management. An RS232 port is available for transmitting the correction data to the GNSS module.

4 Results and conclusion

During development, it was possible to compare the results of current improvements with a commercially available device developed for a similar purpose (iMAR iNAT-M200). In these tests, for example, we validated the results of the already mentioned delay compensation. In Fig. 4, it can be seen that the self developed position estimation (orange) fits the iMAR position prediction output (purple). Based on the more significant delay of the raw position, we assume that the GNSS chipset used in the iMAR is older than the ZED-F9P. The output of the presented device has more noise and corrections when a new GNSS measurement is available, which is also perceivable on the plot. This originates from the IMU inaccuracy, which means that the next steps must be taken in this direction.

Furthermore, in collaboration with the Department of Automotive Technology, we had the opportunity to perform a successful autonomously controlled double lane change during high-speed tests based on the signals from our device. We collaborate with the Faculty of Civil Engineering during the work, responsible for testing and configuring the base stations.

In the current phase of the work, it can be said that the performance meets the expectations. In the following, the main goal is to improve the performance by decreasing the prediction error from several different sources and increasing the reliability based on test results and user feedback.

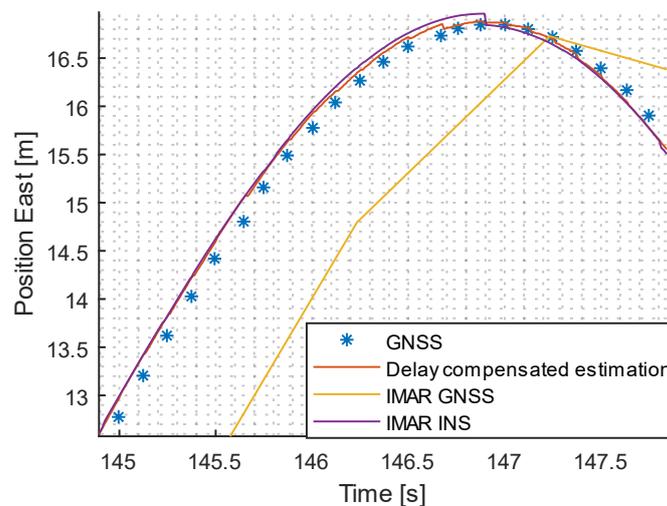


Fig. 4 Position delay comparison in the function of time.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Yan W, Zhang Q, Wang L et al. (2020): „A Modified Kalman Filter for Integrating the Different Rate Data of Gyros and Accelerometers Retrieved from Android Smartphones in the GNSS/IMU Coupled Navigation”. *Sensors*, Vol. 20, Page 5208, 20(18), 5208.
- [2] Lan, Haiyu & Elsheikh, Mohamed & Abdelfatah, Walid & Wahdan, Ahmed & El-Sheimy, Naser. (2019). Integrated RTK/INS Navigation for Precision Agriculture. 4076-4086. 10.33012/2019.17116.
- [3] Falco, G., Pini, M., & Marucco, G. (2017). Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios. *Sensors* 2017, Vol. 17, Page 255, 17(2), 255.
- [4] Weisong Wen, Xiwei Bai, Yin Chiu Kan, and Li-Ta Hsu (2019): „Tightly Coupled GNSS/INS Integration via FactorGraph and Aided by Fish-Eye Camera”, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, VOL. 68, NO. 11
- [5] R. Munguía (2014). „A GPS-aided inertial navigation system in direct configuration”. *Journal of Applied Research And Technology*, 12(4), 803–814.

Design of a Novel Road Traffic Control System for ZalaZONE Proving Ground

Tamás Tettamanti^{1a}, István Varga¹

¹ *Department of Control for Transportation and Vehicle Systems
Faculty of Transportation Engineering and Vehicle Engineering
Budapest University of Technology and Economics*

^a *Corresponding author: tettamanti.tamas@kjk.bme.hu*

Abstract

The testing of Connected and Automated Vehicles (CAVs) and that of the smart infrastructure (traffic control devices and vehicle sensors) in relation with CAVs will be supported by the development of a novel type of road traffic light management system at ZalaZONE Automotive Proving Ground. The system to be developed aims to allow a fully flexible traffic signal control during vehicle and system testing. The system shall provide a freely programmable open Application Programming Interface (API) towards the traffic light control units in contrast with traditional (rigid and closed) traffic control equipments. In this concept, each traffic light control unit will be made available via a remote control software running on a cloud system.

Keywords: road traffic control, traffic light, ZalaZONE

1 Introduction

The development is motivated by the favorable situation that a new, automotive proving ground was constructed in Hungary (Zalaegerszeg), called ZalaZONE (<https://zalazone.hu/>) [1]. This test track is specifically designed for Connected and Automated Vehicle (CAV) as well as for Cooperative, Connected and Automated Mobility (CCAM) testing and homologation processes in the near future. The more, the mission of ZalaZONE is not limited to pure commercial use. It also aims to actively support research and innovation activities in national and international cooperation both with academia and industrial partners.

In this research, a system is designed for a fully flexible road traffic control system at ZalaZONE Proving Ground where 7 signalized junctions are located at the Smart City Zone and 3 signalized intersections at the University Test Track. Additionally, a public road intersection in Zalaegerszeg city will be signalized in 2022, which shall be also controlled by the proposed traffic management system.

2 Traffic Light Control in the era of CAVs

Until now all realizations of traffic lights have been based on the fact that traffic signals are perceived by human drivers exclusively. Therefore, all relevant standards prescribe the technical requirements according to the capability of human perception, e.g. traffic lights' radiation angle [2] or the position and number of traffic signal heads at the road crossing.

With the presence of automated cars the time has arrived to fundamentally rethink the classical approach concerning the production and operation of traffic light controllers. The goal of this technological revolution is the cooperation between the traffic controller and the vehicles, i.e. V2X (Vehicle to Everything) communication technologies. This can be realized in one-way or two-way communication:

- Traffic light controller provides messages to road vehicles which process the received information for their own purposes.
- Road vehicles communicate information to traffic light controller.
- The communication is bidirectional between the traffic controller and the vehicle.

In relation with the wireless technology, the technical specification for Signal Phase and Time and Map Data (SPaT/MAP) [3] must be taken into consideration when talking about future traffic controller design. SPaT/MAP

offers a potential channel for detailed information exchange between traffic systems and road users. Based on SPaT data the vehicles (or drivers) can be informed about the current status and change of the traffic signal ahead as well as about the next signal stage change. It also provides information about approaching traffic to optimize the signal system. MAP data describes the physical geometry of one or more intersections. In connection with SPaT/MAP the ISO/TS 19091:2017 norm [4] is also important to mention as it defines the message, data structures, and data elements to support exchanges between the roadside equipment and vehicles.

3 Traffic Control System Design for Test Track

The aim of the planned road traffic control is to enable a flexible system such that traffic signal heads (vehicle, bicycle, pedestrian and auxiliary signals) can be controlled even separately and freely during vehicle tests. The control of all traffic light at the ZalaZONE test tracks shall be made available by means of a central control software running in a cloud system. The basic concept of the designed system is illustrated in Fig. 1. where each traffic lights as well as traffic signal heads can be arbitrary controlled. A more specified architecture is shown by Fig. 2.

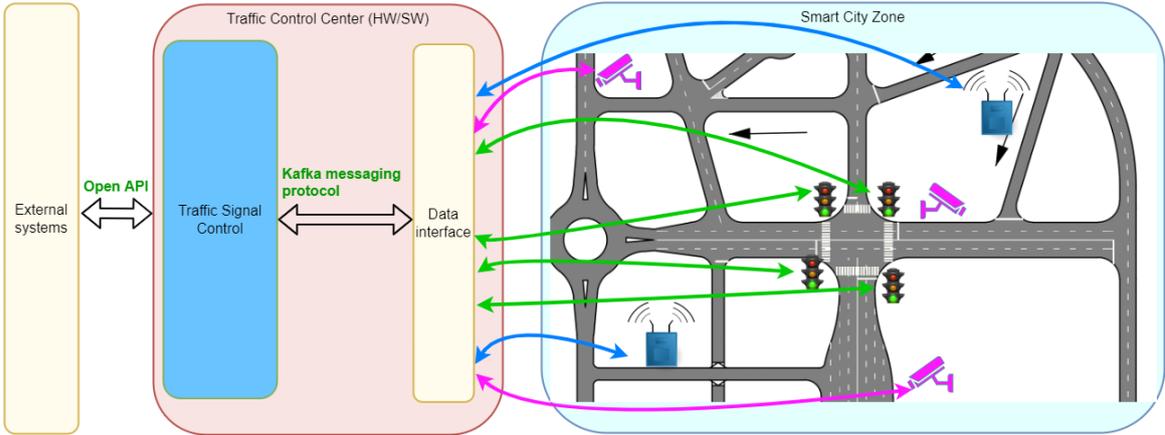


Fig. 1 Overview of the ZalaZONE traffic light control system

The central software controls the traffic light system via Kafka Messaging Protocol. Furthermore, the control center shall be made available to external systems via an open API too. This means that users might apply own (flexible) traffic control logic when testing, basically using “getter” and “setter” functions for traffic light operation.

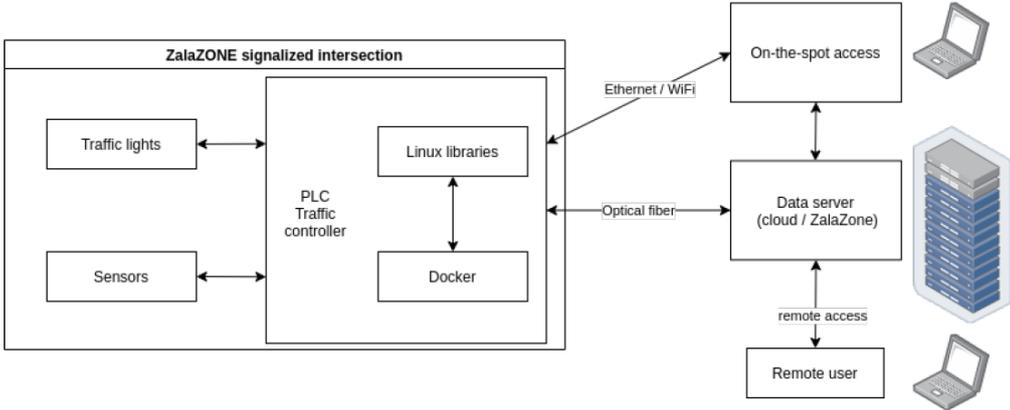


Fig. 2 System architecture of the traffic control

The main requirements for the control system is defined as follows. The traffic light control system shall be freely programmable. All safety systems common in road traffic management systems (and required by standard originally) shall be flexibly switched on or off (i.e. when deactivated, there is no intergreen time matrix, no green conflict monitoring). The central control software shall also ensure that the signal heads are accessible at all times for verification: the software shall continuously check that the predetermined signal phases (even if intentionally irregular for testing purposes) are displayed on the light points and that the LED bulbs are not broken down.

The control center has three control modes:

- Signal Control Script:
Operation according to a predefined sequence in a script file (a case of this is the conventional fixed-time program protected by intergreen time matrix).
- Signal Control API:
Control implemented by commands from an arbitrary program (e.g. Matlab or Python script) via open API.
- Signal Control GUI:
Control can be realized via a GUI. In practice, it means an arbitrary modification of the currently running program.

The hierarchy between the 3 control modes introduced above holds the following sequence. A Signal Control Script based control (1) can be overwritten by logic (2) via the Signal Control API or modified by control (3) via the GUI at any time. Additionally, a Signal Control API based control (2) can only be overwritten by an intervention through the Signal Control GUI (3).

In the designed system the following access levels are defined for users.

- Access level "admin" denotes access to every function and the development environment.
- Access level "tester" means access to every relevant function.
- Access level "researcher" is the access to every function.
- Access level "demo/viewer" means access to limited functions.

4 Software development via PLC devices

The central management software will be developed according to the predefined requirements: all light points shall be freely programmable in the system. This means that even all safety functions, which are used in traditional traffic light controllers according to the standards (e.g. intergreen time matrix checking), can be switched off while testing with adequate security measures. To realize this mission, proper controller devices are needed on the intersection spots. Therefore, at the signalized ZalaZONE intersections WAGO Programmable Logic Controllers [5] (PLCs, shown by Fig. 3) will be used for:

- local control of the traffic lights and
- allowing remote control from the central traffic management system.

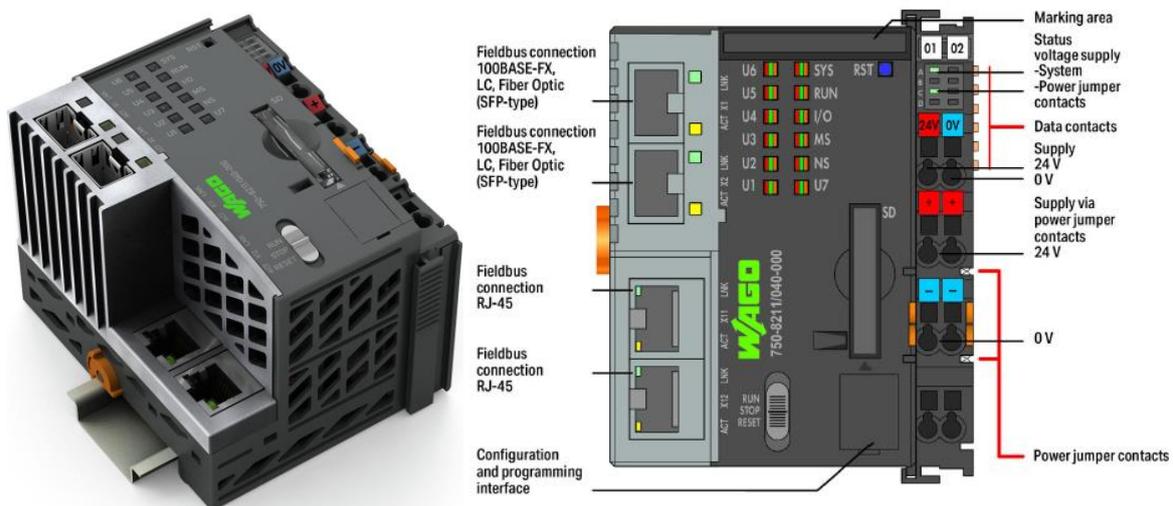


Fig. 3 Programmable Logic Controller (WAGO PFC200) for ZalaZONE signalized intersections (pictures from [5])

The WAGO PLCs work on Linux based operation system allowing a fully flexible programming. On the one hand, a basic local program works on this type of PLC. On the other hand, in the background Linux programs can be run. In this vein, the cloud based central traffic management software can arbitrary change the PLC live running programs via Apache Kafka Messaging Protocol.

A representative realization of the designed system will be the remote management option, i.e. the traffic lights will be capable to be monitored and controlled via the internet from anywhere.

5 Conclusion

A novel road traffic management system is planned aiming to allow a fully flexible traffic signal control during vehicle and system testing at ZalaZONE Proving Ground. The methodology of the proposed traffic management system can be adopted for any other test tracks or even for real-world traffic control system. The system to be developed will also support the future customers of ZalaZONE as in this way the tests can be monitored and saved real-time. In relation with this last option, the digital twin concept must be emphasized which is also a future plan of the designed system. This work will use two simulation software packages: PTV VISSIM (a common industry software) and SUMO (open source software mostly used by researchers). The digital twin will allow to real time mimic the test tracks' traffic light control in parallel in traffic simulation software environment. By the help of this the so called mixed reality simulation [6] and Vehicle-in-the-Loop [7] testing processes can be implemented in the future.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology. The research is also supported by the "MMA Michelberger Mesterdíj Pályázat".

References

- [1] Szalay, Zs. (2021). Next Generation X-in-the-Loop Validation Methodology for Automated Vehicle Systems. IEEE Access, 9, 35616-35632, doi: 10.1109/ACCESS.2021.3061732.
- [2] CEN (2015). Traffic control equipment. Signal heads. EN 12368:2015.
- [3] Amsterdam Group (2015). Signal Phase and Time and Map Data, version 1.1. Technical report, 2015.
- [4] CEN (2017). Intelligent transport systems - Cooperative ITS - Using V2I and I2V communications for applications related to signalized intersections. ISO/TS 19091:2017.
- [5] WAGO (2022). Data sheet, Item number: 750-8211/040-000, Controller PFC200; 2nd Generation; 2 x ETHERNET, 2 x 100Base-FX; Extreme, www.wago.com/750-8211/040-000
- [6] Szalai M., Varga B., Tettamanti T. and Tihanyi V. (2020), Mixed reality test environment for autonomous cars using Unity 3D and SUMO, In 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 73-78.
- [7] Horváth M.T., Lu Q., Tettamanti T., Török Á., Szalay Zs. (2019). Vehicle-In-The-Loop (VIL) and Scenario-In-The-Loop (SCIL) Automotive Simulation Concepts from the Perspectives of Traffic Simulation and Traffic Control, Transport and Telecommunication Journal. 2(20), 153-161.

Double Lane Change Path Planning Using Reinforcement Learning with Field Tests

Árpád Fehér¹, Szilárd Aradi^{1a}, Tamás Bécsi¹

¹ *Department of Control for Transportation and Vehicle Systems,
Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics*

^a *Corresponding author: aradi.szilard@kjk.bme.hu*

Abstract

Performing dynamic double lane-change maneuvers can be a challenge for highly automated vehicles. The algorithm must meet safety requirements while keeping the vehicle stable and controllable. The problem of path planning is numerically complex and must be run at a high refresh rate. The article presents a new approach to avoiding obstacles for autonomous vehicles. To solve this problem, a geometric path generation is provided by a single-step continuous Reinforcement Learning (RL) agent. At the same time, a model-predictive controller (MPC) handles the lateral control to perform the dual lane-change maneuver. The task of the learning agent in this architecture is optimization. It is trained for different scenarios to provide geometric route planning parameters at the output of a neural network. During training, the goodness of the generated track is evaluated using an MPC controller. A hardware architecture was developed to test the local planner on a test track. The real-time operation of the planner has been proven. Its performance has also been compared to human drivers.

Keywords: *Local path planning, Model predictive control, Reinforcement learning, Vehicle dynamics*

1 Introduction

With the beginning of the 2010s, machine learning, in-depth learning and artificial intelligence have undergone rapid development. In addition to classical control planning and decision algorithms, they are advancing in solving control tasks, especially for various vehicle control tasks. It seems that combining artificial intelligence-based developments with control techniques can be an effective method for autonomous vehicle controls.

All vehicles need to drive the most optimal route when performing a critical maneuver, such as a double lane change. Several optimization criteria can be specified to minimize jerk and lateral acceleration. The moose test defined by ISO 3888-2 is a good tool for testing the stability of a vehicle in a dynamic limit situation.

The best path and trajectory planning algorithms for fully autonomous vehicle functions were covered in the following review publications [1], [2], [3]. A path is a sequence of waypoints that the vehicle must follow, referred to as a trajectory with supplied velocity information.

Several solvers can handle the entire constrained optimization problem, albeit it presents the issue of real-time-ness [4]. One approach is to use deep learning to train a neural network for the solutions of many optimization results and use it as a practical solution or an initial guess for the solver [5], which can work for simple setups but is a difficult task to cover the entire state space in more complex scenarios.

Another method is using Reinforcement Learning (RL). The agent interacts with its environment based on trial-and-error and previous experiences and learns the best behavior using performance measurements called rewards [6]. These techniques often use end-to-end solutions, which means the agent responds to steering and acceleration demands. The trajectory planning is planted somewhere in the knowledge acquired via training. Many sensor models, such as grid-based [7], beam sensors [8], camera [9], or ground truth position information [10], can be utilized since the agent can cope with unstructured data. The other end of the RL-based research spectrum focuses on strategic decisions, defining high-level actions, and delegating task execution to an underlying controller, which is beyond the scope of this study. Only a few solutions exist in the literature where the path planning is done with RL [11], [12]. A survey on RL-based motion planning can be found in [13].

2 Methodology

Our current research aimed to create an experimental method to plan the optimal route for a double lane change in real-time, and some simplifications were introduced: constant asphalt-wheel friction coefficient, the vehicle has an ideal high-level sensor model, nonlinear single track vehicle model is used, and the maneuver is performed on a straight section of the road.

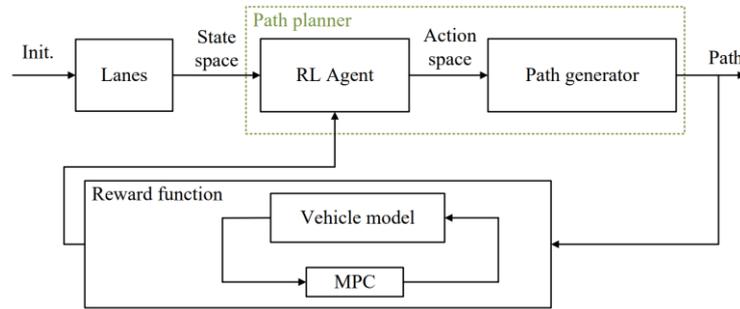


Fig. 1 RL agent training architecture with classical control loop

This task was solved differently from the classical optimisation based approaches. The reinforcement learning was combined with geometric path planning. The state-space contains the lane's width, length, and position that the vehicle must traverse without touching the lane boundaries. Arc length and curvature parameters of polynomial and straight sections constitute the action-space. The generation of points x, y of the track is performed by a path generator based on the action-space. The path consists of two straight and three polynomial sections. The coefficients of the polynomials are calculated by determining their curvature function. The function of the path yaw angle and x, y points are determined by numerical integration along the arc length, which gives a good approximation.

The state space is 11, and the action space is 10 continuous values. The task is to generate the optimal route for a given condition or lane option.

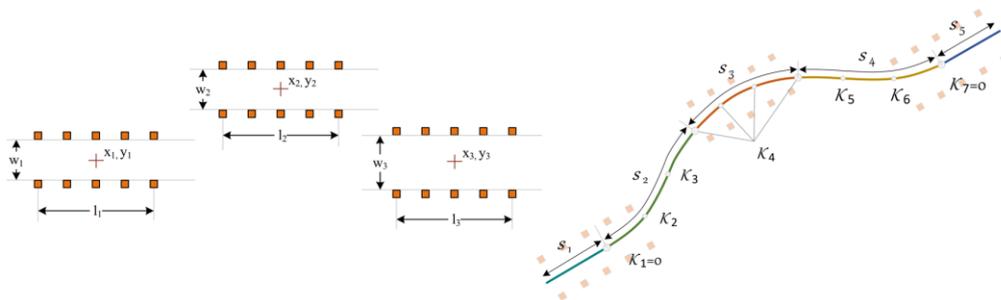


Fig. 2 State and action-space representation of the training environment

In reinforcement learning, an agent is taught, which requires a simulation environment. The training process needs many iterations. An iteration is called a step, and a series of steps is called an episode. One-step teaching was used wherein each step, the agent receives a different random but executable or near-executable state space and predicts an action space from which the path can be generated. The modeled vehicle is guided along the track by an MPC controller responsible for lateral control. Longitudinal control of the vehicle should only be performed until acceleration. The slip values, lateral acceleration, angle, and distance errors from the course are evaluated during the course. The value of the reward or the penalty can be determined if a cone has been touched or the track has been left. A TD3 (Twin-Delayed Deep Deterministic Policy Gradient) agent was implemented for training in a Python environment.

3 Hardware architecture for field test

The trained neural network of the agent consists of 3 hidden layers and a few hundred neurons, allowing fast and real-time prediction. The agent has been implemented on a Jetson AGX Xavier running Robot Operating System 2, which receives state space elements via CAN network and predicts action space.

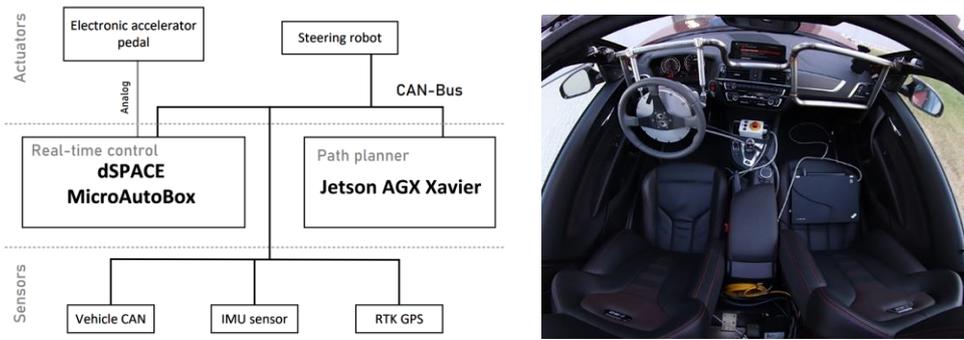


Fig. 3 Hardware architecture and experimental setup

A real-time dSPACE MicroAutoBox generates the track, and an MPC controller performs lateral control-based RTK GPS based on the action space. The real-time device also performs longitudinal control. A steering robot has been installed, and by removing the accelerator pedal, the engine control electronics receive the analog signal directly. The RTK GPS, IMU, and vehicle CAN data are logged for later evaluation.

A series-production BMW M2 car with a competition package was used for the tests, which after the modifications mentioned above, is suitable for throttle and steer-by-wire operation. The installation of the instruments is shown in the figure.

4 Results

After training the agent, the results were also validated with field tests. For real vehicle tests, a moose test as defined by ISO 3888-2 was performed at the ZalaZONE test track in Zalaegerszeg. The cones of the test were placed cm accurately with RTK GPS on the center of the 300 m diameter ultra-flat dynamic platform. While two supervising drivers were sitting in the vehicle, the system automatically accelerated to an initial speed until the ISO standard specified torque release point and then performed the double lane change.

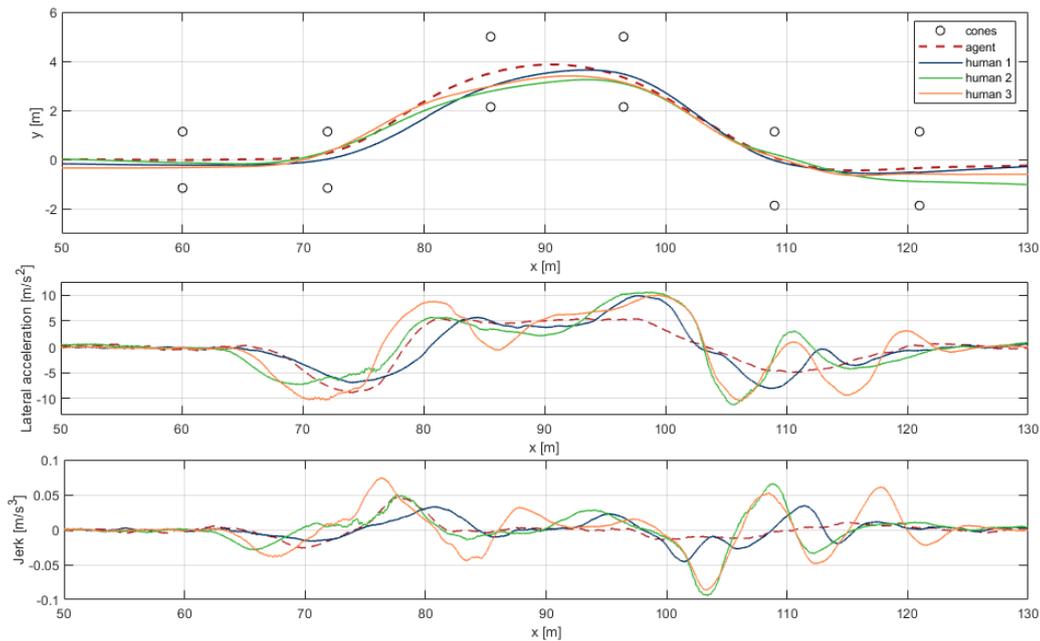


Fig. 4 Traversed routes, lateral acceleration, and jerk of moose test experiments

The agent successfully passed the test with 60 km/h maximum initial speed. The limitations of the lateral controller determine the maximum speed. In the case of a human driver, only 5 out of 14 ended without cone touch. There was already a cone touch when entering the first lane in many cases. Fig. 4 shows an attempt at nearly the same initial speed. The dashed line represents the agent's traversed path, side acceleration, and jerk. The blue successful and the green and orange represent unsuccessful attempts of three different human drivers. The figure shows that the agent performs the path with less lateral acceleration and jerk. A driver who succeeds often yanks

the steering wheel, while drivers who have performed with errors have made many corrections. The tests also showed that the first lane-change maneuver was the most difficult, as the vehicle's speed here was the highest, and the drivers are shown in the diagram also made a mistake here.

5 Conclusion

The article presents real-time geometric path planning by an agent which used reinforcement learning to perform a double lane maneuver. It is clear how effectively the machine learning approach can be combined with classical control algorithms since an MPC controller was used for lateral control during agent teaching and real-time tests. Drivers of average ability find it challenging to solve such a tricky maneuver, even at relatively low speeds. The following research topic will be the avoidance of moving objects.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [2] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.
- [3] D. Gonzalez, J. Perez, V. Milan'es, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.
- [4] Y. Lin, J. McPhee, and N. L. Azad, "Longitudinal dynamic versus kinematic models for car-following control using deep reinforcement learning," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1504–1510.
- [5] F. Hegedüs, T. Bécsi, S. Aradi, and P. Gáspár, "Motion planning for highly automated road vehicles with a hybrid approach using nonlinear optimization and artificial neural networks," *STROJNISKI VESTNIK JOURNAL OF MECHANICAL ENGINEERING*, vol. 65, no. 3, pp. 148–160, 2019.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [7] A. Folkers, M. Rick, and C. Buskens, "Controlling an Autonomous Vehicle with Deep Reinforcement Learning," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 6 2019, pp. 2025–2031.
- [8] J. Lee, T. Kim, and H. J. Kim, "Autonomous lane keeping based on approximate Q-learning," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 6 2017, pp. 402–405.
- [9] P. Wolf, C. Hubschneider, M. Weber, A. Bauer, J. Hartl, F. Durr, and J. M. Zollner, "Learning how to drive in a real world simulation with deep Q-Networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 6 2017, pp. 244–250.
- [10] W. Xia, H. Li, and B. Li, "A Control Strategy of Autonomous Vehicles Based on Deep Reinforcement Learning," in *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*. IEEE, 12 2016, pp. 198–201.
- [11] M. P. Ronecker and Y. Zhu, "Deep Q-Network Based Decision Making for Autonomous Driving," in *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 6 2019, pp. 154–160.
- [12] Á. Fehér, S. Aradi, T. Bécsi, P. Gáspár, and Z. Szalay, "Proving Ground Test of a DDPG-based Vehicle Trajectory Planner," in *2020 European Control Conference (ECC)*, 2020, pp. 332–337.
- [13] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2020

Point Cloud Based Road Surface Modelling and Assessment

Tamás Lovas^{1a}, Dániel Baranyai¹, Árpád Somogyi¹

¹ Department of Photogrammetry and Geoinformatics, Faculty of Civil Engineering,
Budapest University of Technology and Economics

^a Corresponding author: lovas.tamas@emk.bme.hu

Abstract

Creating digital twins of the built environment based on point clouds broadens its application area; point clouds of buildings support BIM (Building Information Modeling) while the digital twins of road surfaces support transportation applications. Point clouds acquired by current TLS (Terrestrial Laser Scanning) or by MLS (Mobile Laser Scanning) systems represent the road surface with high accuracy, and resolution (i.e. with small point spacing). In particular applications extreme high accuracy and robustness is required; the paper discusses surveying and assessment of vehicle test track, including a braking platform. Both the surveying method (TLS supported by Total Station measurements) and the data processing workflow are presented. The potential evaluation results are discussed in detail; semi-automatically created 2D sections and deviation maps shows how the current state of the pavement differs from the as-designed geometry. The investigations proved that point-cloud based data acquisition methods enable deriving high accuracy and high-resolution road surface models.

Keywords: MLS, OpenCRG, point cloud, TLS

1 Introduction

As-built documentation reconstruction planning of roads are based on geodetic surveying of sections in every 25 meters [1]. These surveys fulfil the general construction requirements (earthworks calculations, accuracy assessment) but do not provide sufficient, detailed information to vehicle dynamics simulations or for detecting minor failures of the pavement. In such cases applying surface-based data acquisition technology is recommended that generally provide point clouds of the surveyed object/area. In engineering practice imagery-based spatial reconstruction (e.g. SfM – Structures from Motion) and laser scanning are the most widely used technologies to capture high accuracy, high resolution geometry of structures. The end products of such surveys enable creating the digital twins of the particular structures/facilities.

Application of point clouds to various engineering purposes has been investigated for years, e.g. engineering geology [2], cultural heritage protection [3], reverse engineering [4]. Abby et al (2015) captured road profiles and analyzed the IRI (International Roughness Index) values [5]. El Issaoui et al (2021) assessed the potential of MMS (Mobile Mapping System) to detect pavement failures [6]. Égető et al (2020) discussed how point clouds can support road construction works, they used total station measurements as reference [7]. Guan et al (2015) developed a method to automatically extract road features (road surface, road markings, pavement cracks) from MLS (Mobile Laser Scanning) datasets; the resolution of the point cloud limited the detection of small cracks [8].

Current paper discusses investigations for both TLS and MLS data sets from the ZalaZone Aquaplaning braking platform. The surface of the track was assessed by TLS data and the general applicability and reliability of MLS data is presented.

2 TLS based road surface modelling and assessment

The applied Surphaser 400 laser scanner enables high accuracy geometric assessment due to its technical capabilities: 1 mm ranging accuracy and 0.1 mm ranging noise. During pre-processing of the point cloud farthest points reflected with low intensity or by low incident angle has been filtered out therefore reducing the risks of certain errors. The aquaplaning test track is a 200 m long road segment; in order to achieve robust registration of point clouds captured from different scan positions, we applied control points all over the area that have been

measured by a Leica TS16i total station. This also enabled georeferencing the data set; further measurements from the same area can be compared to each other this way. The maximum residual value in the control points was 4 mm during the registration.

Based on the design plans the reference model has been created. The aquaplaning track starts with a 15 m long, 1% slope, then it has a 180 m long horizontal segment, closed by a 15 m long slope at the end. The reference surface composed by 3 planes enables to create a deviation map that shows the differences from the as-designed model.

Such high-resolution deviation map becomes easy to be managed if elevation data is structured in a regular grid, applying conditional formatting for color coding. Fig. 1 shows a grid with 1 m longitudinal- and 1 dm cross-resolution; elevation values have been interpolated with nearest neighbor method from the point cloud.



Fig. 1 Regular grid elevation map

From the pre-processed point cloud different engineering products have been derived. Cross-sections in every 5 meters, while points of longitudinal sections (Fig. 2) in the track axis and crossfall values have been defined by 1 m spacing using semi-automated technique. Annotations have been added in CAD environment.

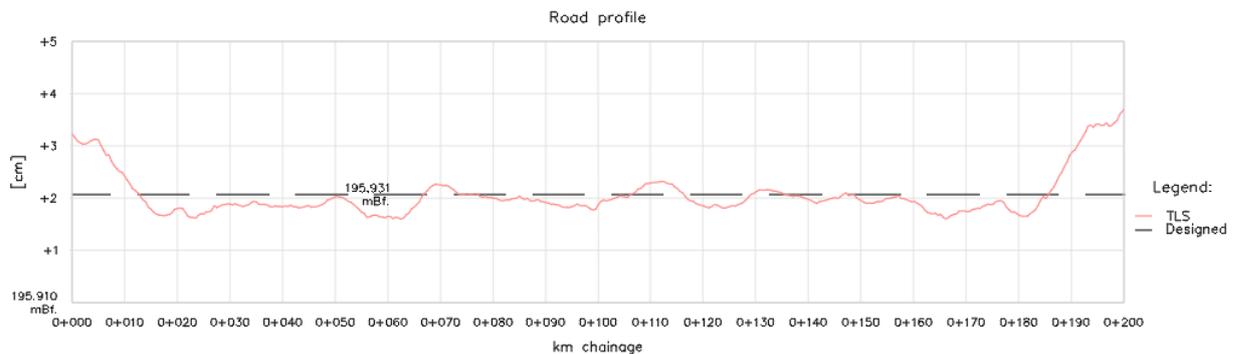


Fig. 2 TLS - Longitudinal section

Although point clouds have multiple application opportunities, deriving surface model is beneficial in multiple cases. TIN (Triangulated Irregular Network) enables effective data storage and manages interpolation between measurement points [9]. CRG (Curved Regular Grid) models can be created using the surface model and the as-designed model; such CRG models can be applied in vehicle simulation environments (e.g. IPG CarMaker) (Fig. 3) [10, 11]. According to OpenCRG standard road surface elevation data is to be structured in cm-resolution in a grid along the road axis. Having the axis line, the primary (x and y) coordinates can be defined, and the elevation (z) values can be interpolated on the surface model.

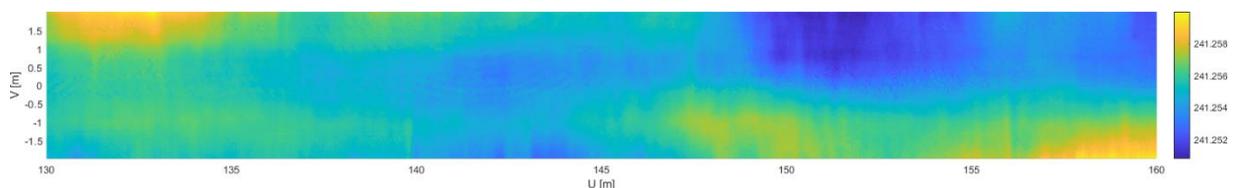


Fig. 3 TLS OpenCRG model UVZ map, section 0+130.00-0+160.00

The derived products show the potential of investigating road surfaces is such high accuracy and high-resolution point clouds are available. Cross-sections enable detecting small differences, while OpenCRG models support simulations with extreme accuracy requirements.

3 MMS data assessment

MMS data was captured by a Leica Pegasus 2 resulting connecting point cloud swaths. During pre-processing these datasets have been merged and the area of interest has been extracted. Compared to TLS data one main difference is in the number of points: MMS dataset contained 17 million, while that of TLS had 130 million points.

Regarding point distribution, TLS point cloud spacing is smaller close to the scan positions, while MMS produce more evenly distributed points.

One way to assess geometry is creating deviation maps; most of the differences between the TLS and MMS data lie in the 0-8 mm range (Fig. 4, Fig. 5)

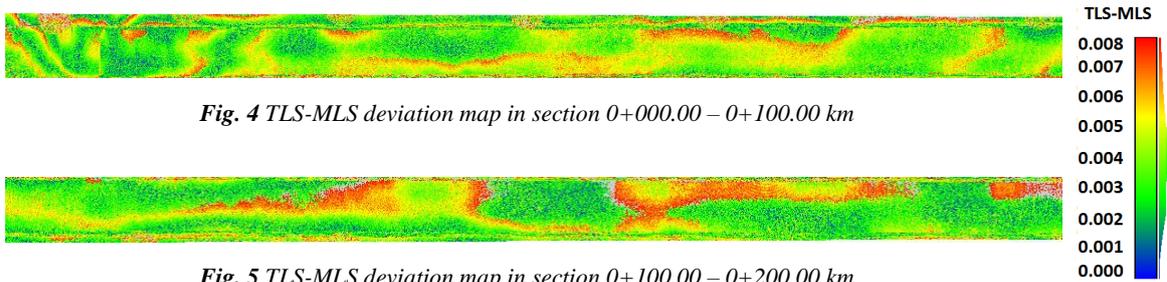


Fig. 4 TLS-MLS deviation map in section 0+000.00 – 0+100.00 km

Fig. 5 TLS-MLS deviation map in section 0+100.00 – 0+200.00 km

In order to further evaluate the MMS data quality, same products have been derived as it was shown in case of TLS. Crossfall values were computed from the longitudinal section points; the trend is similar, however, reasonable difference can be observed in some sections (Fig. 6).

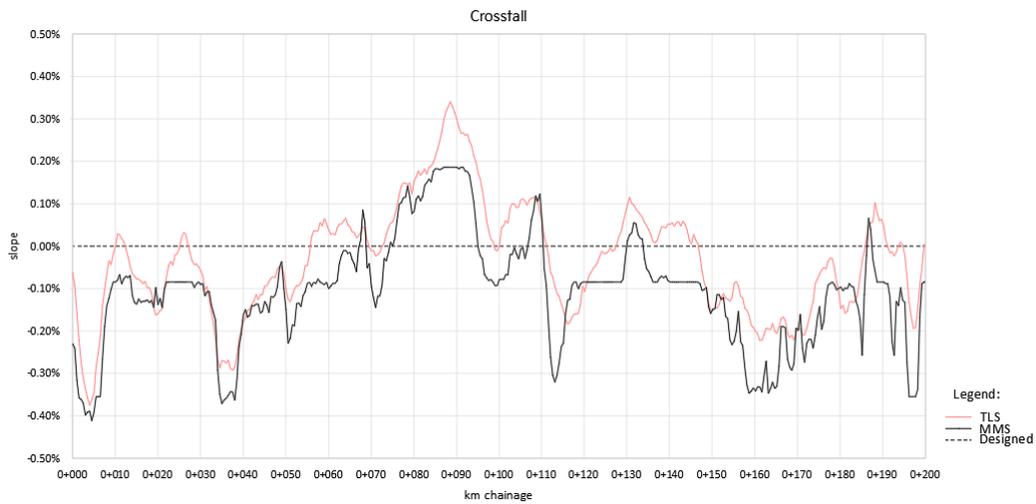


Fig. 6 TLS-MLS crossfall values

Analyzing the cross sections, no such small differences can be observed, as in case of TLS, however, the general trend of the road geometry seems to be correct (Fig. 7).

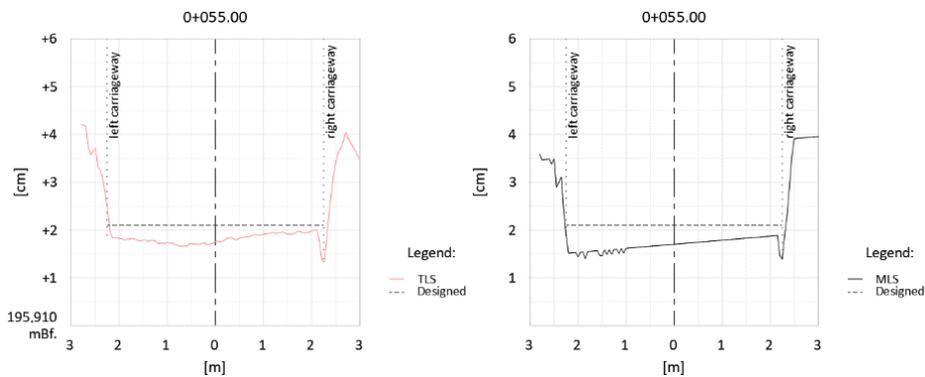


Fig. 7 TLS (left) and MMS (right) cross sections in 0+055.00 km

OpenCRG models derived from both data sources show similar results as in case of cross- and longitudinal sections. CRG models from MMS are less detailed compared to TLS-based models (Fig. 8).

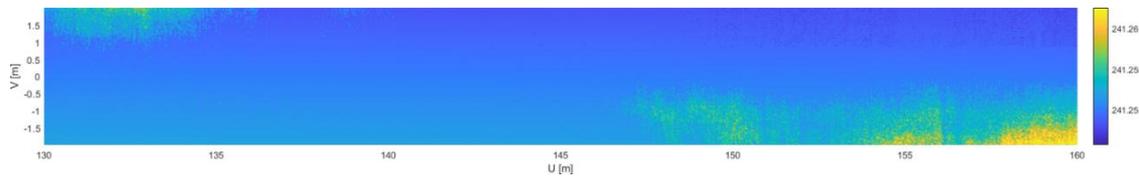


Fig. 8 MLS OpenCRG model 0+130.00-0+160.00

4 Conclusion

The results proved how high-resolution, accurate point clouds can effectively support road surface investigations, detecting pavement failures (e.g. potholes, cracks), differences compared to the as-designed models (e.g. in crossfall values). MMS data can describe the overall road geometry, shows the trends of changes, but TLS is to be recommended in case of extreme accuracy and resolution demands. Data acquisition method is to be chosen according to the requirements of the desired end products.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] e-UT 09.04.15:2018 Közutak geodéziai előírásai és geometriai követelményei, 16/2017 (V.25.) NFM rendelet, 2018.
- [2] Á. Török, G. Bögöly, Á. Somogyi and T. Lovas, "Application of UAV in Topographic Modelling and Structural Geological Mapping of Quarries and Their Surroundings—Delineation of Fault-Bordered Raw Material Reserves," *Sensors*, p. 20, 2020.
- [3] T. Lovas, N. Rehány and J. Á. Somogyi, "Történelmi épületek rekonstrukciós munkálatainak támogatása pontfelhők segítségével," *Geodézia és Kartográfia*, vol. 70, pp. 19-24, 2018.
- [4] T. Lovas, Á. J. Somogyi and G. Simongáti, "Laser Scanning Ship Hulls to Support Hydrodynamic Simulations," *Periodica Polytechnica Civil Engineering*, vol. 66, pp. 291-297, 2022.
- [5] C. Abby and M. J. Olsen, "Evaluation of Technologies for Road Profile Capture, Analysis, and Evaluation," *Journal of Surveying Engineering*, vol. 141, p. 13., 2015.
- [6] A. El Issaoui, Z. Feng, M. Lehtomäki, E. Hyypä, H. Hyypä, H. Kaartinen, A. Kukko and J. Hyypä, "Feasibility of Mobile Laser Scanning towards Operational Accurate Road Rut Depth Measurements," *Sensors*, vol. 21, 2021.
- [7] É. Csaba, S. Gherman, B. Hrutka, N. Nagy and B. Takács, "Pontfelhő az útéépítésben," *Geodézia és Kartográfia*, vol. 72, pp. 13-17, 2020.
- [8] H. Guan, J. Li, Y. Yu, M. Chapman and C. Wang, "Automated Road Information Extraction From Mobile Laser Scanning Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 194-205, 2015.
- [9] M. Kazhdan, M. Bolitho and H. Hoppe, "Poisson Surface Reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, Cagliari, Sardinia, Italy, 2006.
- [10] G. IPG Automotive, "ipg-automotive.com," [Online]. Available: <https://ipg-automotive.com/en/products-solutions/software/carmaker/>. [Accessed 13 02 2022]
- [11] A. Barsi, V. Potó and V. Tihanyi, "Creating OpenCRG Road Surface Model from Terrestrial Laser Scanning Data for Autonomous Vehicles," *Lecture Notes in Mechanical Engineering*, pp. 361-369, 2018.

Establishment of a Local GNSS Correction Service for the Localization of Autonomous Vehicles

Szabolcs Rózsa^{1a}, Ágnes Ács¹, Bence Turák¹

¹ Department of Geodesy and Surveying, Faculty of Civil Engineering,
Budapest University of Technology and Economics

^a Corresponding author: rozsa.szabolcs@emk.bme.hu

Abstract

Accurate localization of autonomous vehicles is a key component of the onboard control and guidance system. Global Navigation Satellite Systems (GNSS) are widely used in the transportation industry for positioning and navigation, but the generally used single point positioning (SPP) technique cannot meet the accuracy requirements of the autonomous vehicles. This paper briefly introduces other GNSS positioning techniques with the accuracies ranging from several meters to centimeters. The techniques are compared in terms of accuracy, latency and reliability. Highly accurate positioning techniques usually rely on a groundbased augmentation system (GBAS) that provides correction services for the users. We introduce the procedure and the first results of establishing a local GNSS correction service at the ZalaZONE Automotive Proving Ground. The results show that cm level positioning accuracy can be achieved with such a service in real-time that enables the users to track the trajectory of the vehicles with high accuracy.

Keywords: DGNSS, GNSS, RTK, SPP

1 Introduction

Determining the positions of observing sites on land, at sea, or in the air using observations of distant objects has been carried out for hundreds of years. However, it was only with the space-age and the appearance of artificial satellites that it became possible to develop a global system for high accuracy positioning and navigation. As a result, the global navigation satellite system (GNSS) was established. A historical review of the evolution of satellite-based positioning can be found in [1].

The principle of satellite-based positioning is the spatial trilateration process using ranges measured to satellites with known positions. GNSS receivers record the run time (τ_S) required for the signal to reach the receiver from the satellite antenna. This time interval can be easily converted to range by multiplying it by the speed of light. When the known position vector of the satellite is denoted by ρ^S and the unknown position vector of the receiver is ρ_R , the measured range takes the form

$$\rho_R^S = \|\rho^S(t - \tau_S) - \rho_R(t)\|, \quad (1)$$

where both vectors are relative to the geocenter. Since the satellite and receiver clocks are not synchronized, the receivers observe the so called pseudoranges:

$$P_R^S = \rho_R^S + c(dt_R - dt^S), \quad (2)$$

where dt_R and dt^S denote the clock offsets of the receiver and the satellite (transmitted by the satellites). Thus, at least four satellites are needed to determine a kinematic 3D position, since the receiver clock error is unknown and it is changing rapidly.

The fundamentally harmonic radio wave GNSS signals (termed the carrier) are modulated with a characteristic pseudorandom noise (PRN) code and with a low-rate navigation data message. This enables two methods to measure the pseudoranges to each satellite. Using the code ranges (C/A code) to observe the pseudoranges one can achieve an accuracy of approximately 3 m and the observation is unambiguous. On the other hand, using the carrier phase without the codes, one can improve the accuracy of the pseudoranges to several mm. However, phase range observations are ambiguous, only the fractional wavelength can be measured with such an accuracy and the integer ambiguities – the number of full waves between the satellite and the receiver – must be resolved during the position

computation. In the next sections we will introduce the fundamental GNSS positioning techniques, that could be used for positioning autonomous vehicles.

2 GNSS positioning techniques

In this paper we aim to present various positioning techniques tested in a real case study to assess their characteristics and accuracy measures. Approximately 2 hours of GNSS observations taken by a static low-cost U-blox F9P multi frequency multi GNSS receiver are processed with each positioning technique. The measurements were processed with the RTKlib open-source software [2] using GPS, Glonass, and Galileo dual-frequency observations.

2.1 Single point positioning

The equation of absolute positioning with code ranges is the following:

$$P_i^j(t_i) = \rho_i^j(t_i - \tau_i^j, t_i) - c\delta t_i(t_i) + c\delta t^j(t_i - \tau_i^j) + T_i(t_i) + I_i^j(t_i) + v_{pj}, \quad (3)$$

where P_i^j is the measured pseudorange (between satellite j and receiver i), ρ_i^j is the geometric distance between the satellite and the receiver. $c\delta t^j$ is the effect of satellite clock error, which is transmitted as part of a navigation message broadcast by the satellites. T_i and I_i^j denote the effect of the troposphere and the ionosphere on signal propagation. In real-time applications without any further information, these effects can be mitigated using the correction models defined either in the navigation messages or by empirical functions. More information about clock errors and the effects of the atmosphere is available in [3] or in [4].

Fig. 1 shows the time series of the horizontal coordinate residuals. One can observe that the spread of the position solutions is in the order of several meters. The GPS system provides the horizontal position accuracy of less than 13 meters at the 95% probability level (2σ) in 24 hours anywhere on the globe [5]. Our results shows a much lower uncertainty in the study period mostly because of the inclusion of the additional satellite systems.

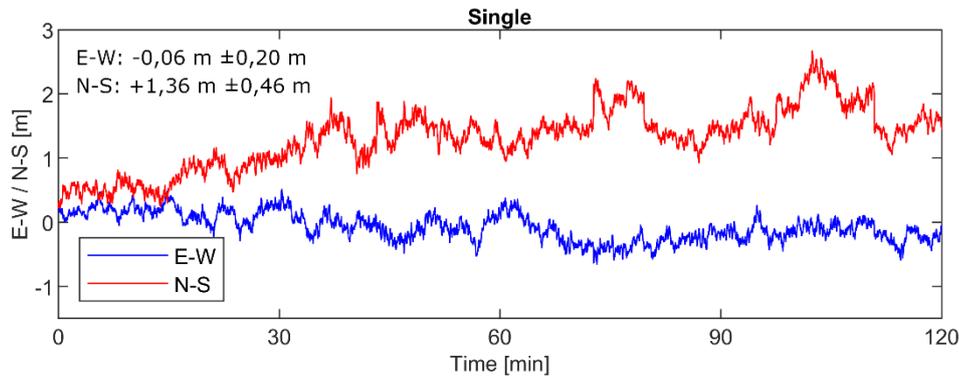


Fig. 1 Absolute positioning with code ranges (120 minutes of results)

2.2 Differential positioning with code ranges

Although SPP technique meets the requirements of navigational applications, several positioning applications of moving platforms require significantly higher accuracy, such as the guidance and control of autonomous cranes or the tracking of autonomous vehicles. The SPP technique has several limitations. Satellite clocks are synchronized on the level of ca. 5ns leading to a ranging error of ca. 1.5m, broadcast orbits has the accuracy of ca. 1m and other systematic error sources like ionospheric delays are not modeled on an accuracy level to achieve submeter positioning accuracy, To overcome this problem, the differential positioning technique was introduced (DGNSS). DGNSS uses additional simultaneous observations provided by a base station to reduce the effect of the mentioned systemic error and to improve the accuracy of the positioning in real-time. The base receiver is installed on a site with known coordinates and it tracks all the available GNSS satellites in view. By taking the pseudorange observations the base receiver is capable to calculate the pseudorange residual using its own coordinates and the known position of the satellite:

$$\Delta P_b^j(t_i) = \rho_b^j(t_0, t_i) - P_b^j(t_i) = c\delta t_b(t_i) - c\delta t^j(t_i - \tau_i^j) - T_b^j(t_i) - I_b^j(t_i). \quad (4)$$

This pseudorange residual contains information on the effect of the troposphere and the ionosphere, the orbit error and the clock biases of the satellite and the base receiver. The pseudorange residuals (corrections) are transmitted to the rover receiver in real time to correct the observations of the rover receiver:

$$P_r^j(t_i) + \Delta P_b^j(t_i) = \rho_r^j(t_i - \tau_r^j, t_i) - c\delta t_{br}(t_i) - \Delta T_{br}^j(t_i) - \Delta I_{br}^j(t_i) \quad (5)$$

Thus, one obtains a similar equation to (3), but the systematic error sources are either eliminated or significantly reduced. It must be noted that DGNSS positioning technique can be solved with exactly the same algorithm as SPP using the corrected pseudorange observations. Fig. 2 shows the results of the differential positioning relative to ground truth in our case study. The results show that submeter level accuracy can be achieved with this technique without the need for initialization, so the position solution is available instantly after observation gaps. Moreover one can observe that the experienced coordinate uncertainties are much lower compared to the SPP case reaching the sub-meter level.

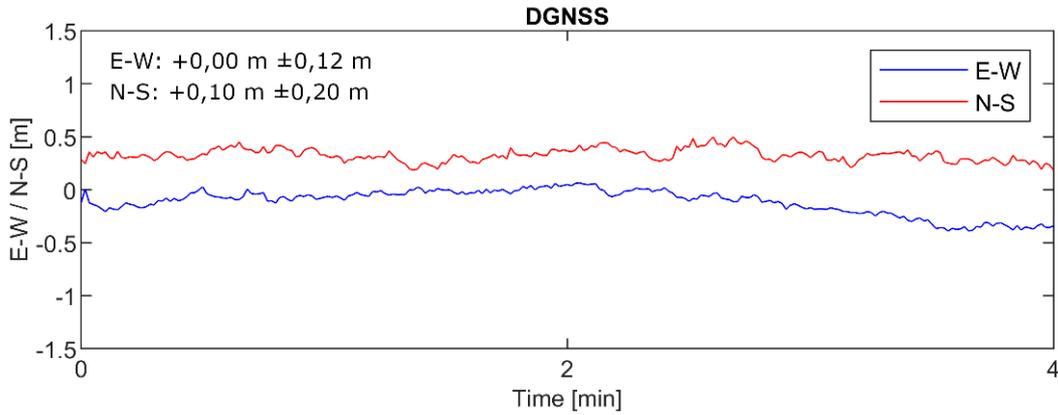


Fig. 2 Differential positioning with code ranges (first 4 minutes of results). The statistics show results of 120 minutes of observation.

2.3 Relative positioning with carrier phases

In case of the relative positioning technique the PRN (pseudo random noise) codes are removed from the modulated signal and the pure carrier signal is used to measure the satellite-receiver distance. The carrier waves have the wavelength of ca. 20 cm. GNSS receivers create a replica of the carrier wave and compare this replica with the received carrier wave signal to measure the phase angle difference between them. This phase angle difference enables us to measure the fractional distance (within one λ) with high accuracy. However, the number of full carrier waves between the satellite and the receiver is still to be found. Thus the equation of phase ranges are as follows:

$$\Phi_r^j(t_i) = \rho_r^j(t_i - \tau_r^j, t_i) - c\delta t_r(t_i) + -c\delta t^j(t_i - \tau_r^j) + \lambda \cdot N_r^j + T_r^j(t_i) - I_r^j(t_i) + v_{\Phi_r^j}(t_i). \quad (6)$$

where N is the phase ambiguity. Let's assume that one observes m satellites in s epochs in a kinematic application. Since the number of unknowns is $m+(3+1+m)\times s$ (m phase ambiguities and 3 coordinates, 1 receiver clock error and m satellite clock error per epoch) and the number of observations is only $s\times m$, the numerical solution of this problem is impossible. In case of PPP (Precise Point Positioning) additional high accuracy satellite clock offsets are introduced in the calculation to decrease the number of unknowns [2]. Another method is to use the relative positioning technique and eliminate the quickly changing receiver and the satellite clock error.

Similarly to the DGNSS positioning, the position of the receiver could be determined relative to a base station. Calculating the difference of the observations of the two receivers at the same epoch for each satellite the single-difference of phase ranges are obtained:

$$\begin{aligned} \Delta\Phi_{b,r}^j(t_i) &= \rho_r^j(t_i - \tau_r^j, t_i) - \rho_b^j(t_i - \tau_b^j, t_i) - c\delta t_r(t_i) + c\delta t_b(t_i) \\ &+ \lambda \cdot (N_r^j - N_b^j) + [T_r(t_i) - T_b(t_i)] + v_{\Phi_{b,r}^j} \end{aligned} \quad (7)$$

This equation gives the vector between the base and the rover receiver. Unfortunately the receiver clock error values are still changing rapidly over time causing a large number of unknowns. To eliminate them, double differences are calculated by differencing the single differences with respect to a reference (pivot) satellite:

$$\begin{aligned} \Delta\Delta\Phi_{b,r}^{j,k}(t_i) = & \rho_r^k(t_i - \tau_r^k, t_i) - \rho_b^k(t_i - \tau_b^k, t_i) - \rho_r^j(t_i - \tau_r^j, t_i) + \rho_b^j(t_i - \tau_b^j, t_i) + \\ & + \lambda \cdot (N_r^k - N_b^k - N_r^j + N_b^j) + [T_r^k(t_i) - T_b^k(t_i) - T_r^j(t_i) + T_b^j(t_i)] + v_{\Phi_{b,r}^{j,k}} \end{aligned} \quad (8)$$

In our example the number of double differences is $s \times (m-1)$, and $3s+(m-1)$ unknown parameters must be determined. Although the relative position vector can not be solved from observations taken in a single epoch, it can be solved using several epochs of observations. It must be noted that despite ambiguities (N) are integer numbers by definition, they can be only estimated as real numbers (float solutions), and they must be resolved in a subsequent step as integers (fixed solutions) to achieve cm accurate positions [3]. Fig. 3 shows the initialization part of solving kinematic relative positioning in our case study. One can clearly see the initialization phase of the positioning, and how the accuracy improves after fixing the ambiguities. Other results of using phase ranges in vehicle tracking can be found in [6] and [7].

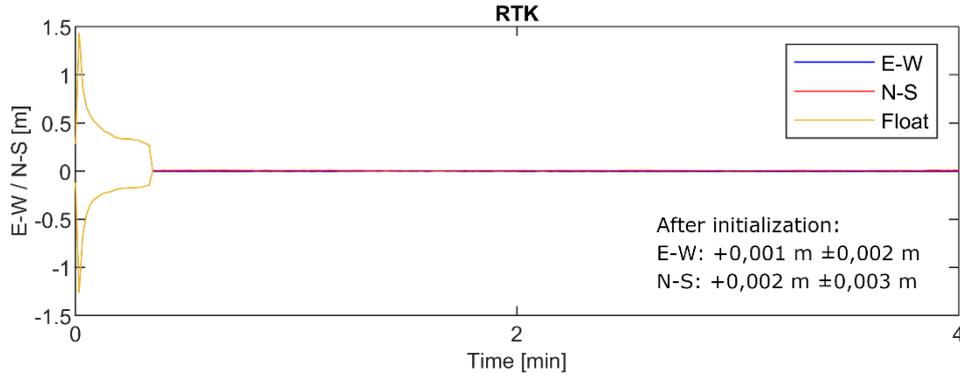


Fig. 3 Relative positioning with carrier phases (first 4 minutes of results). The statistics show results of 120 minutes of observation and only fix solutions.

Summarizing the various positioning techniques, one can see that SPP and DGNS techniques has the advantage of providing instantaneous positioning solutions, while RTK and PPP has a longer initialization (convergence) time. Although RTK technique provides cm accurate coordinate solutions after fixing the ambiguities, it also needs a certain initialization time after each signal loss to achieve this accuracy level. The positioning solution accuracies show that DGNS technique is capable to provide lane level positioning accuracy with quick response time, thus it is ideal for vehicle tracking applications on the Proving Ground. However, the accuracy of RTK positioning is required for both dynamic testing applications and the development of the localization techniques of autonomous vehicles. Based on the comparisons it is clear that a ground based augmentation service is fundamental for the operation of the ZalaZONE Automotive Proving Ground. In the next section we will briefly introduce the establishment of such a station at ZalaZONE.

3 ZalaZONE permanent station

We have seen that several positioning technique used in vehicle tracking applications need correction services provided by a fixed permanent reference station. To assist the testing of vehicle dynamics and autonomous control and driving algorithms at the ZalaZONE Automotive Proving Ground, a dedicated reference station has been established. The station consists of a Leica GR30 geodetic multi-GNSS receiver capable to track all the available GNSS and SBAS (Satellite Based Augmentation Systems) satellites including GPS, GLONASS, Galileo, Beidou and EGNOS. The station is equipped with a Leica AR20 geodetic antenna manufactured with choke ring elements to mitigate multipath effects (Fig 4). The phase center offset and variation of the antenna has been determined using the absolute chamber calibration technique developed at the Geodetic Institute of the University of Bonn [8]. Thus, the equipment fully complies with the quality standards of the European (EUREF) Permanent Network.

The GNSS station provides three different correction streams through the NTRIP (Networked Transport of RTCM via Internet Protocol) broadcaster of the Department of Geodesy and Surveying at the Budapest University of Technology and Economics. The broadcast streams are:

- ZZON0 providing RTCM 3.2 MSM7 (RTCM Multiple Signal Messages v7) GNSS correction for GPS/GLONASS/Galileo and Beidou satellites realizing the highest accuracy corrections available;
- ZZON1 providing RTCM 3.2 corrections for GPS and GLONASS satellites only due to compatibility issues for older receivers;

- ZZON2 providing RTCM 3.2 MSM4 (RTCM Multiple Signal Messages v4) GNSS corrections for GPS/GLONASS/Galileo and Beidou satellites realizing GNSS corrections with a lower transmission bandwidth a lower accuracy with respect to ZZON0.



Fig. 4 The ZZON Permanent Station at the ZalaZONE Automotive Proving Ground

The station coordinates have been determined using weekly observations of nearby EPN reference stations using the Bernese GNSS processing software V5.2 [9]. The station coordinates have been calculated in the European Terrestrial Reference Frame (ETRF2000) and refers to the epoch of the 2007.4, which corresponds to the geodetic datum of the Hungarian Active GNSS Network (GNSSNet.hu). Thus, the GNSS correction service seamlessly fits to the national GNSS correction service provided by the GNSSNet.hu. The station coordinates are given in Table 1. One must note that differential and relative positioning techniques provide coordinate solutions in the reference frames used to defined the reference station coordinates, while absolute positioning techniques provide solutions in the WGS-84 (World Geodetic Systems 1984) or ITRS (International Terrestrial Reference System) global reference frames. Currently the lateral difference between the continental ETRF and the global WGS-84/ITRS coordinates is close to one meter.

Table 1 The ETRF2000 (Epoch 2007.4) coordinates of the ZZON station

X	4179601.3173
Y	1264445.0024
Z	4633710.2155

4 Outlook

The permanent GNSS reference station becomes a fundamental infrastructure of the ZalaZONE Automotive Proving Ground. However, it can play an important role in the maintenance of the Hungarian and the European geodetic infrastructure, too. The station currently holds the status of being a proposed station in the EUREF Permanent Network and transmits realtime corrections to the EUREF-IP network as well as provides hourly and daily observations for the geodetic community. The observations are routinely processed not only to obtain repeated coordinate solutions of the stations for quality monitoring purposes, but also to use the information content of the GNSS signals for remote sensing atmospheric water vapour to improve the accuracy of weather prediction.

In the near future we plan to include the permanent GNSS station in the Hungarian Active GNSS Network, too.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Ashkenazi, V. (2006). Geodesy and satellite navigation. *Inside GNSS*, April, 44–49.
- [2] RTKlib (2015). Author:T. Takasu and other contributors. Available online: <http://www.rtklib.com/>
- [3] Teunissen, P., & Montenbruck, O. (2017). *Springer Handbook of Global Navigation Satellite Systems*. Springer. <https://doi.org/10.1007/978-3-319-42928-1>
- [4] Hofmann-Wellenhof, B., Lichtenegger, H., & Wasle, E. (2008). *GNSS — Global Navigation Satellite Systems, GPS, GLONASS, Galileo, and more*, Springer-Verlag Wien, <https://doi.org/10.1007/978-3-211-73017-1>
- [5] Global Positioning System Standard Positioning Service Performance Standard (2001). Available at <https://www.navcen.uscg.gov/pdf/gps/geninfo/2001SPSPerformanceStandardFINAL.pdf>.
- [6] Knoop, V. L., De Bakker, P. F., Tiberius, C. C. J. M., & Van Arem, B. (2017). Lane Determination with GPS Precise Point Positioning. *IEEE Transactions on Intelligent Transportation Systems*, 18(9), 2503–2513. <https://doi.org/10.1109/TITS.2016.2632751>
- [7] Sun, Q., Xia, J., Foster, J., Falkmer, T., & Lee, H. (2017). Pursuing Precise Vehicle Movement Trajectory in Urban Residential Area Using Multi-GNSS RTK Tracking. *Transportation Research Procedia*, 25, 2356–2372. <https://doi.org/10.1016/j.trpro.2017.05.255>
- [8] Görres, B., Campbell, J., Becker, M., Siemes, M. (2006). Absolute Calibration of GPS antennas: laboratory test results and comparison with field and robot techniques. *GPS Solutions*, 10, 136-145. <https://doi.org/10.1007/s10291-005-0015-3>
- [9] Dach, R., S. Lutz, P. Walser, P. Fridez (Eds); 2015: *Bernese GNSS Software Version 5.2. User manual*, Astronomical Institute, University of Bern, Bern Open Publishing. DOI: 10.7892/boris.72297; ISBN: 978-3-906813-05-9

Routing Algorithms for Wireless Sensor Networks in Smart Cities

Dániel Pásztor^{1a}, Péter Ekler¹, János Levendovszky²

¹ *Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics*

² *Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics*

^a *Corresponding author: daniel.pasztor@aut.bme.hu*

Abstract

Wireless Sensor Networks (WSNs) are one of the most important parts of the advancements in smart city planning. The sensors provide valuable data about various metrics, such as air quality and traffic status. These measurements must be collected at a centralized data center to evaluate and create analysis results. Sending these messages is the responsibility of the network. Nodes inside a network are usually resource constrained. When the energy grid is not available, a portable energy source is required. To avoid long distance communication requiring high energy, multi-hop communication is recommended inside networks. The routing of the messages must consider the energy use of a node, of the whole network and the probability of the message received by the central base station. This paper focuses on such algorithms, focusing on the typical grid layout of a smart city.

Keywords: *communication, IoT, routing, smart city, WSN*

1 Introduction

Advancements in information technology made it possible to create detailed analysis of different processes in several key areas, such as industrial manufacturing processes, environmental observations and traffic control. This requires large amount of data, which is processed by different Big Data frameworks. This data is measured by different sensors placed at strategical locations and collected at a base station over some form of wired or wireless network, creating a wireless sensor network.

The surge of IoT and low-cost sensor devices drove the installation price of such a network lower. These networks are self-configured and dynamic, making them easy to set up and robust against sensor failures. The network consists of multiple nodes, each node containing one or multiple different types of sensors, and a communication device to send and receive messages from the network. A network typically consists of between 5 and 5000 nodes based on the number of needed measurements and the coverage area of the sensors.

To keep the costs down, each node is usually resource-constrained with limited computational power, and in some cases, limited energy source. Because of this, multi-hop communication is usually preferred, where transmission occurs through multiple intermediary nodes. When the power source is limited (for example, running on a battery), the goal is to keep the nodes working for as long as possible without needing to recharge them.

2 Related Work

In the literature, several different algorithms have been proposed to handle multi-hop routing in wireless sensor networks. LEACH [1] periodically assigns nodes to be cluster heads randomly whose responsibility are collecting the messages from the sensors close to them. After receiving the messages, the cluster heads combine these into a single packet, and apply a compression algorithm before sending it to the base station. The compression and two-hop communication reduce the energy needed for the communication. The cluster head approach is advantageous when the base station is placed somewhere near the wireless sensor network.

PEGASIS [2] creates a chain between the nodes close to each other. At every round, the measured values are aggregated and sent towards one particular node through the chain, which in turn transmits to the base station. This

transmitting node is changed after every message sent. PEGASIS is mainly used when only aggregated data is needed at the base stations (such as minimum, maximum and average value), and the base station is placed farther away. In this algorithm, the nodes themselves handle the aggregation process, and only one node communicates with the base station at a time, reducing the energy needed for the long-distance communication. However, this algorithm is highly inefficient if the base station must receive every measured value.

E2MR2[3] is an advanced routing algorithm designed specifically for sensor networks in smart city configurations. The algorithm uses a model based on link loads to minimize the network's bit energy consumption parameter.

Most of the routing algorithms suppose a model where the transmission of a packet is only based on the length of the message and the distance of the communication. However, wireless communication is usually lossy, and so the packet can be lost. Our goal is to provide a model where a quality-of-service criteria can be met providing a guaranteed probability of successful message transmission.

3 Model

To derive algorithms and conduct simulations, we first defined the following: our network consists of a base station and N stationary nodes, each of which might want to send a message to the base station. Nodes can be placed randomly or according to some pre-defined topology without any kind of obstruction, so every node can send and receive message from every other node. A randomly generated network can be seen on Fig. 1. We also assume that the nodes are energy limited, and each node starts with an initial energy E.

We assume that for a given transmission, the sending node can choose how much energy it plans to use, which determines the probability of a successful transfer according to the Rayleigh fading model, which will be explained later. Our Quality-of-Service criterion is that the base station must receive the message sent by the node with a given probability, denoted by the P_s probability value. The model runs as long as this criterion can be accomplished, meaning that there is enough energy left in the nodes to complete the transmission with the given probability.

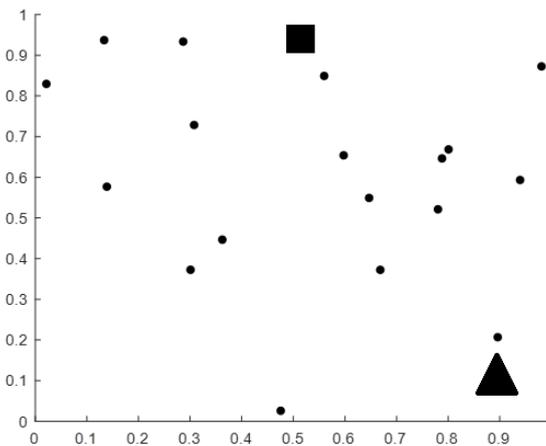


Fig. 1 An example of a randomly generated wireless sensor network. The square represents the base station, while the triangle shows the next node sending a message.

To calculate the probability of successful transmission, we based our equation on the Rayleigh fading, which is a statistical model for the effect of a propagation environment on a radio signal.

$$g_{ij} = -d_{ij}^{\alpha} \frac{\theta \sigma_z^2}{\ln P_{ij}} \quad (1)$$

As can be seen, the energy needed for a given transmission (g_{ij}) is a function over the distance between the nodes (d_{ij}), environmental and transmission factors (θ, σ_z, α), and the probability of the successful transfer (P_{ij}). Since the nodes are stationary, we can simplify this equation to the following formula:

$$g_{ij} \ln P_{ij} = \omega_{ij}, \quad (2)$$

where ω_{ij} is a constant value between two nodes in the network.

4 Energy aware routing algorithm

After defining our model, we started developing algorithms to route messages between nodes in a multi-hop scenario. For this, we needed to decide which property of the network should be optimized. Our first idea was to minimize the energy used during the multi-hop transmission while maintaining the quality-of-service criteria. This however resulted in a static routing algorithm, where the energy used during a transmission would not depend on the energy levels of the nodes.

To solve this, we have decided to incorporate the current energy level of the nodes. This is accomplished by maximizing the lowest node energy level after a transmission happens. This way, higher-energy nodes will be more likely chosen, and they will participate with more energy. We have proved that the lowest node energy level is maximized if and only if the energy levels of the nodes participating in the transmission reach an equilibrium.

Another property of our routing algorithm is the maximum number of hops allowed in a transmission. While higher number of hops will usually provide a higher common node energy level for any given transfer, it will also require more energy due to the number of nodes participating. We differentiate between different strategies by calling it k -hop routing, where k is the maximum number of hops allowed.

5 Topology based routing

In many situations the topology of the network has a special structure and this structure can be used to further optimize the routing algorithm.

A practical example for a given network structure is the ZalaZONE facility which is a large test field of sensor networks used to test autonomous and self-driving cars. The test field has a *smart city area* where the sensors are distributed in a grid, which is similar to larger city layouts as can be seen on Fig. 2.



Fig. 2 Comparison of the layout of ZalaZONE and New York

6 Results

To test our algorithm, we have implemented the model and algorithm in Matlab. For testing our proposed algorithms, we have devised different configurations and scenarios. In every simulation, we have chosen the environmental constants to be the same, so we can compare the results between different executions. We have placed the nodes based on the model of the ZalaZONE Smart City Area. We have also chosen the ω_{ij} values so that sending a message diagonally can only be done with significantly higher energy usage. With this setup, we have run 100 different simulations, and calculated the average number of messages successfully sent on the network.

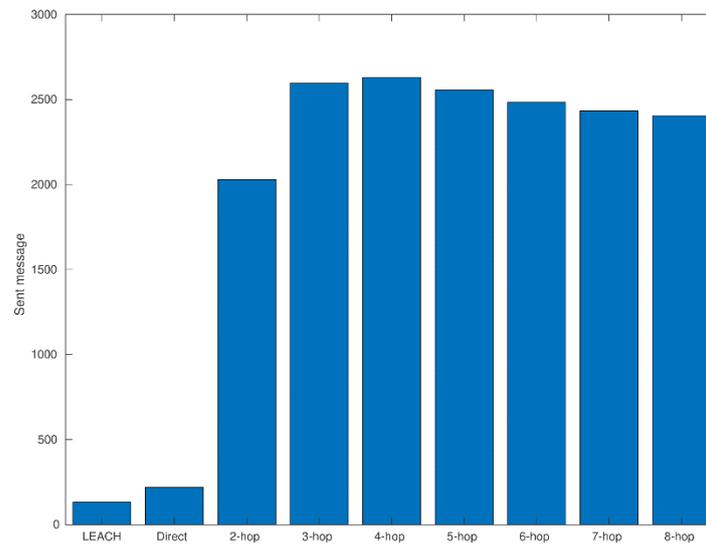


Fig. 3 Results of different k -hop algorithms and LEACH

As can be seen on Fig. 3, the k -hop algorithms outperformed both Direct sending and LEACH algorithms. Unfortunately, while the LEACH algorithm was implemented to also prefer the main axes, where transmission is easier, the random cluster head selection results in situations where not every column or row is covered. Due to this, sending messages to cluster heads, and transmission between cluster heads and base station will require higher energy use. Meanwhile, our k -hop routing algorithms can always choose an optimal path for message transmission. It can also be seen that in the case of the ZalaZONE layout with the given parameters, 4-hop routing seems to be the optimal strategy.

7 Conclusion

In this paper we have looked at different routing algorithms for energy aware IoT data communication in smart city layouts. As can be seen, most algorithms ignore the lossy nature of wireless transmission. Our proposed algorithm provides a quality-of-service criteria for successful message transmission. We have also seen that the k -hop algorithm could be applicable for wireless sensor networks in smart city layouts.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, pp. 10 pp. vol.2-, doi: 10.1109/HICSS.2000.926982.
- [2] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power-efficient gathering in sensor information systems," Proceedings, IEEE Aerospace Conference, 2002, pp. 3-3, doi: 10.1109/AERO.2002.1035242.
- [3] D. Jiang, P. Zhang, Z. Lv and H. Song, "Energy-Efficient Multi-Constraint Routing Algorithm With Load Balancing for Smart City Applications," in IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1437-1447, Dec. 2016, doi: 10.1109/IJOT.2016.2613111.

Predictive Maintenance in Distributed Environments

Dávid Sik^{1a}, Péter Ekler¹, János Levendovszky²

¹ Department of Automation and Applied Informatics,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics

² Department of Networked Systems and Services,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics

^a Corresponding author: sik.david@aut.bme.hu

Abstract

In this paper we present a concept to implement predictive maintenance in distributed environments. A distributed environment can involve several problems to be solved such as the different speed and frequency of data, the different priorities of data sources, the different dimensions and data structures and also the volume of the data to be processed. There are software, frameworks, algorithms to work out these problems, however the interoperability is harder to maintain between these. Recently it become possible to use historical and real-time datasets of parameters and key performance indicators of environments in order to prevent failures and monitor the system state. The aim is to outline a solution, using open-source tools, to support the lifecycle from the data extraction, through the transformation and algorithmic steps, until the usage and visualization of the gained information and feedback of the data to the environment.

Keywords: data lake, distributed environment, predictive maintenance, time series

1 Introduction

Data is all around us. In the 21th century using the newest digital and industrial systems and environments, we generate a large amount of data. This is a consumer-driven approach: using our smartphone, opening a webpage, driving our car etc. While on the producer side these generated data can be used to extract information, make business decisions, personalize advertisements, recommend products, improve services etc. It is up to us, how we would like to use this data.

To make it possible to effectively use the generated data, we need to think about many steps. How can we collect the data? How can we store the data? How can we extract information of the data? What are the key performance indicators? How can we feedback the knowledge to support business decisions, to improve services? How can we manage the interoperability between the different participant of the process? How can we visualize and highlight the achieved results? Is it worth it at all?

In this paper we would like to present a concept to answer these questions. The chosen domain is predictive maintenance in distributed ZalaZONE related environment. The main goal of predictive maintenance is to use historical and real-time data to predict and prevent failures and monitor the system state in a given multi-user, multi-participant environment without having critical system downtimes and outages.

The organization of this paper is as follows:

- In Section 2 we present the background of distributed environments.
- In Section 3 the data lake based architecture is overviewed.
- In Section 4 the topic of predictive maintenance is introduced.
- In Section 5 we validate the previously detailed approaches in a ZalaZONE related case study.
- Finally, in Section 6 we wrap up the paper and discuss the further research, development and innovation directions.

2 Distributed environment

The distributed methods already have a long history in the information technology. The creation of a distributed system, program, application, computing, or computing environment become possible by the usage of different distributed architectures and patterns. In a distributed environment there are multiple participants communicating between each other with messages on wired or wireless media in a networked organization [1].

The common examples are the internet, the mobile telecommunication systems, computer clusters, wireless sensor networks etc. The distributed approach can be found in the cloud infrastructures as well, such as Google Cloud, Amazon AWS etc. The participants can be distributed in a geographical way or role/task-oriented way too.

What strategy do we receive data from data sources? How can we manage different speeds of data sources? Is it master-client or peer-to-peer? Is there a master element polling the clients or the clients pushing to a master? Many questions have to be covered. There are several solutions such as Hadoop, Amazon AWS, SAP HANA etc. These are including many components to support the communication and interoperability of the participants [2].

In our concept we are going to use Hadoop ecosystem with Hadoop Distributed File System (HDFS) which will act as the master element. The distributed participants will be able to publish their messages over a MQTT broker and these messages will be saved simultaneously in the distributed file system as well.

3 Data lake

To support storing historical data, Data Warehouse (DWH) solutions are preferred instead of Operational Data Store. However, to store data in a DWH it is necessary to go through an extract-transform-load (ETL) process [3]. This transformation process usually includes the steps of data cleansing, filtering, altering the data in order to make it possible to store it in a previously designed data scheme, created relations, structured tables in a given normalization level. During these steps we might lose important information, so a new approach was introduced.

In a Data Lake we store the data in raw format, as we received it from the source system(s). This usually means semi-structured data in CSV, XML or JSON format. It is also possible to store unstructured or structured data as well in a Data Lake. A Data lake can be established "on premises" or "in the cloud" [4].

Most of the current solutions are based on distributed environments such as in Hadoop, where the entry point of the data can be a corresponding Hadoop component such as Flume, Kafka, Nifi etc. including load balancing. The raw data can be stored as blob or files on the HDFS for example in a date-time based folder structure: /user/lake/{appId}/Y/%m/%d/%H/ or can be loaded into relational or non-relational (NoSQL) databases.

4 Predictive maintenance

Maintenance has an important role since the industrialization of our society. It is originated from the fields of machinery, mechanical and vehicle engineering sciences, however by the newer industrial revolutions, many other fields of sciences introduced and applied it such as engineering and computer sciences as well.

In the beginning maintenance had only a corrective approach (deferred or emergency), later the preventive approach was introduced such as a scheduled or state based maintenance (condition, time, risk etc.) [5, 6]. Then with the integration of statistical algorithms or artificial intelligence based solutions the predictive maintenance also become available to the participants of industrial processes [7, 8]. Based on the EN 13306:2017 standard predictive maintenance is an extended aspect of condition-based maintenance where a forecast derived from repeated analysis or known characteristics and evaluation of different significant parameters [9].

In our model we are using a time series based approach, where the information can be scalar or multi-dimensional as well. X_t is the value in moment t , θ is a threshold value, δ is an optimal value. We created two different criteria:

- If $X_t \leq \theta$ the system operates normally, if $X_t > \theta$ the system is malfunctioning.
- If $|X_t - \delta| \leq \theta$ the system operates normally, $|X_t - \delta| > \theta$ the system is malfunctioning.

Based on the past observation of the time series, it is possible to estimate that the system remains in the normal operating state for the next K steps with ε confidencey:

$$P(X_{t+K} \leq \theta, X_{t+K-1} \leq \theta, \dots, X_t \leq \theta | X_{t-1} = x_1, X_{t-2} = x_2, \dots, X_{t-T+1} \leq x_T) \geq 1 - \varepsilon, \text{ or} \quad (1)$$

$$P(|X_{t+K} - \delta| \leq \theta, |X_{t+K-1} - \delta| \leq \theta, \dots, |X_t - \delta| \leq \theta | X_{t-1} = x_1, X_{t-2} = x_2, \dots, X_{t-T+1} \leq x_T) \geq 1 - \varepsilon \quad (2)$$

It is possible to train a neural network with the known part of the conditional probabilities in Eq. (1) and Eq. (2) and at the output we get whether there are at least K steps to the failure or not.

5 Case study

We found the ZalaZONE Industrial & Science Park as an excellent distributed environment to validate our concept. ZalaZONE supports testing and validation of standard vehicle dynamics, fully integrated autonomous vehicles, it also provides test features of environment preparation, complex driving and traffic situations and smart city features [10, 11]. At the Department of Automation and Applied Informatics, we established a Smart Data Lake Environment, including Hadoop Core (HDFS, YARN, MapReduce) and Oozie, Hive, Hue and Flume components. This is already used by several Big-Data related intelligent services such as correlation, anonymization, clustering and prediction as a service [12].

At ZalaZONE for its Tracker System the test vehicles emitting messages over MQTT protocol about their current status in a corresponding topic. This message includes the latitude-longitude coordinates, speed, session id, timestamp, accuracy etc. An example of an emitted message is shown by Fig. 1.

```
{
  "accuracy": 9.503912,
  "bearing": 209.36278,
  "lat": 47.5854435866086,
  "lon": 19.367416469843047,
  "sessionId": "a89e09ab-7999-400f-a136-456dec740cbe",
  "speed": 3.085256,
  "timeStamp": "2021-11-10T06:41:09.328Z",
  "type": "fused"
}
```

Fig. 1 An example ZalaZONE data message

Our purpose was to wire the messages publish on the MQTT broker into the Data Lake in order to make it possible to run algorithms and execute queries on the received data, to implement a predictive maintenance solution. We initialized the Flume component to handle the dataflow from the MQTT source into the HDFS sink. In the HDFS the raw data are stored in the folder structure introduced earlier.

In this stage from the HDFS it is possible to use the necessary messages filtered by the timestamp or session id for the prediction step. Due to the current limited attributes of the messages, we tested two predictions about speeding (whether the speed tends to be greater than a threshold value) and geofencing (whether the vehicle tends to leave an area). After an offline training process is done, it is possible to validate the prediction and visualize it on different charts and dashboards. If prediction is not necessary, then it is also possible to execute simpler transformation on the data and load it into the monitoring component. Using the Tracker mobile application is it also feasible to send feedback to the vehicle about the warnings and results. The concept is shown by Fig. 2.

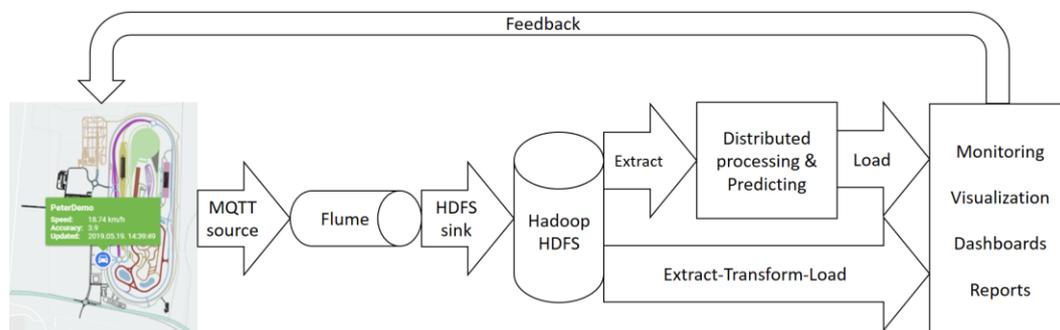


Fig. 2 ZalaZONE related predictive maintenance concept

The real-time based approach is still under development. Fortunately, the Hadoop ecosystem provides several components and tools to support the staging and processing of real-time data, for example: Spark Streaming and Storm [13]. In this case the online training of the neural network in the predictive maintenance component also becomes possible. Also the offline trained neural network can be validated on the real-time data and predictions can be extracted as well. It is also necessary to find the most appropriate architecture and parameters to support online training. Based on real-time data, real-time notifications will be available as well by installing and using the Kafka component. However, storing the data on the HDFS is still important in real-time approach, for data mining, visualization or exploratory data analysis as the set of real-time data subsequently becomes historical data.

6 Summary and future work

In our paper we highlighted that the current solutions have to support the distributed environments and solve the challenges of the related circumstances. We pointed out the importance of storing the data in raw format, how it is received from the source system(s). We explained our time series based predictive maintenance model, that can be used in different fields of sciences. Based on these we outlined and introduced a concept supporting predictions, predictive maintenance in a distributed environment with a concrete ZalaZONE based case study.

We hope that the presented concept, including its real-time extension will be usable in different ZalaZONE related scenarios, services and applications or even in other domains as well. As a next step we would like to predict on more maintenance related parameters as well such as temperature values, engine metrics etc.

In the future we would like to validate our concept in the terms of performance measurements. A major improvement will be the ability to analyze and recognize object of camera images or past/real-time streams to support object detection, labelling and further applications in a smart vehicle, smart city environment [14].

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Bartal, Y., Fiat, A., & Rabani, Y. (1995). Competitive algorithms for distributed data management. *Journal of Computer and System Sciences*, 51(3), 341-358.
- [2] Singh, V. K., Taram, M., Agrawal, V., & Baghel, B. S. (2018). A literature review on Hadoop ecosystem and various techniques of big data optimization. *Advances in Data and Information Sciences*, 231-240.
- [3] Kimball, R., & Caserta, J. (2004). *The data warehouse ETL toolkit*. John Wiley & Sons.
- [4] Zagan, E., & Danubianu, M. (2021). Cloud DATA LAKE: The new trend of data storage. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-4). IEEE.
- [5] Krupitzer, C., Wagenhals, T., Züfle, M., Lesch, V., Schäfer, D., Mozaffarin, A., ... & Kounev, S. (2020). A survey on predictive maintenance for industry 4.0. *arXiv preprint arXiv:2002.08224*.
- [6] Merkt, O. (2019). On the use of predictive models for improving the quality of industrial maintenance: An analytical literature review of maintenance strategies. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 693-704). IEEE.
- [7] Ruschel, E., Santos, E. A. P., & Loures, E. D. F. R. (2017). Industrial maintenance decision-making: A systematic literature review. *Journal of Manufacturing Systems*, 45, 180-194.
- [8] Lee, J., & Wang, H. (2008). New technologies for maintenance. In *Complex system maintenance handbook* (pp. 49-78). Springer, London.
- [9] EN 13306 (2017). *Maintenance. Maintenance terminology*. ISBN: 978-0-580-90370-0.
- [10] Szalay, Z. (2017). ZalaZONE Test Environment to the Future Cars. In *4th Hungarian Future Internet Conference (in Hungarian)*, Budapest.
- [11] Szalay, Z., Tettamanti, T., Esztergár-Kiss, D., Varga, I., & Bartolini, C. (2018). Development of a test track for driverless cars: vehicle design, track configuration, and liability considerations. *Periodica Polytechnica Transportation Engineering*, 46(1), 29-35.
- [12] Baross, A., Soós, S., Egyed, T. B., & Ekler, P. (2021). Building an Extended Smart Data Lake Architecture with Big Data-Related Intelligent Services. *Proceedings of the Automation and Applied Computer Science Workshop 2021*, pp. 69-80.
- [13] da Silva Morais, T. (2015). Survey on frameworks for distributed computing: Hadoop, spark and storm. In *Proceedings of the 10th Doctoral Symposium in Informatics Engineering-DSIE (Vol. 15)*.
- [14] Tschürtz, H., Wagner, F., Schröter, W., Szalay, Z., & Török, Á. (2021). System of Systems Safety Analysis and Evaluation in ZalaZONE. *Periodica Polytechnica Transportation Engineering*, 49(4), 317-323.

Conflict Management Algorithms Development Using the Automated Framework for Autonomous Vehicles

Dávid Szirczák^{1a}, Dániel Rohács¹

¹ *Department of Aeronautics and Naval Architecture,
Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics*

^a *Corresponding author: szirczak.david@kjk.bme.hu*

Abstract

The fields of autonomous ground and aerial vehicles are a young, dynamically expanding field of industry. The current traffic management solutions are inadequate to support the predicted high volume of operations efficiently and safely. As such, novel, automated management solutions are required. This paper presents the research work performed on the automated conflict management prototype system, combining conflict management for autonomous ground and aerial vehicles. This paper briefly presents the background and architecture of the system prototype developed, then highlights the research directions related to the conflict management algorithm development. Algorithms need to provide the desired amount of safety, while at the same time ensuring acceptable levels of performance, cost, and efficiency. Potential conflict management algorithms are investigated, and a computational methodology developed for objective comparison, to enable the future, data-based development of the conflict management framework.

Keywords: *Autonomous conflict management, UAV, UGV, UTM*

1 Introduction

Autonomous vehicles are getting more and more attention these days. The field is a young dynamically expanding market, just the unmanned aerial vehicles market is estimated at USD 100 billion, with a predicted annual growth of 16-24% in the near future [1-3]. In terms of operations this can be translated to about 350 million flight hours by 2050, only in the lower altitudes, referred to as U-Space in the EU [4]. Likewise, it is predicted that by 2045 half of newly sold vehicles, and by 2060 half of all active vehicles will be autonomous [5]. Just at EU level (in 2018) 268 million road vehicles were registered, and there is an annual 2% growth trend [6].

It can be seen that the magnitude of operations is significant. In the aerospace sector, conflicts are managed by the Air Traffic Control Officers (ATCO), which is a manual operation, that requires a highly qualified workforce, and consequently, it is a costly and resource-constrained service. Due to these characteristics, it is not feasible to directly apply the current procedures for the management of unmanned traffic. In the case of autonomous road vehicles, the current industry focus is on the various driving assistance systems of individual vehicles, such as collision warning, adaptive cruise control, lane assist or road sign detection. This is because apart from the highway codes, there is no regulatory oversight or intervention in how the individual drivers operate and manage conflicts, which would be impractical due to the huge volumes of traffic anyway. As such, the most appropriate solution is to assist drivers to help avoid conflicts. New concepts, such as smart cities or connected vehicles are aiming to provide safer, more efficient, and more sustainable mobility, however, they primarily aim to control traffic flow, not necessarily individual vehicles. The tools are for example variable speed limits, dynamic traffic signal timing, dynamic lanes and similar.

As such, it can be concluded that current approaches, neither from the aerial nor the ground side, are adequate to safely and efficiently manage the conflicts between the predicted large volumes of autonomous vehicles. In an urban environment, it can be expected that unmanned aerial and ground vehicles interact with each other, and cause conflict situations. This does not necessarily mean head-on collisions between the two classes of vehicles, but situations such as low-level surveying missions, take-off or landing during delivery missions, emergency and crash landing or evading obstacles or other users. Also, UAVs can be affected by wind generated by traffic,

especially in closely confined urban environments, often referred to as urban canyons [7]. In addition to the interaction possibilities, developing a conflict management system that handles both classes of vehicles benefit from the similarities and synergies provided by these individual classes, as opposed to developing separate tools, and spending effort overcoming similar, or even the same challenges for either of them.

In this paper, the development of a prototype system, capable of autonomously handling conflicts between unmanned aerial and ground vehicles is presented. The authors already published an article describing the theory and practical aspects of the early prototype level development [8]. As such this paper only briefly presents the architecture of the system and rather focuses on the follow-up research directions that are identified and are in development, based on the lessons learnt to test and us the prototype solution.

2 Conflict management system overview

The conflict management system prototype developed implements the short- to mid-term tactical conflict management methodology. Generally, conflict management methods can be divided into 2 categories: tactical and strategic management. Strategic management refers to tools that can be used before an actual operation is started. The aim of the methods is to develop systems and procedures where the potential effects of lack of information and faulty planning can be minimized. Typical example tools are rerouting, rescheduling, or cancelling specific operations. Also, infrastructure developments, policy making, market development or traffic research can be generalized as very long term (up to 10 years or even more) strategic management tools. The system presented does not implement strategic management, these approaches work separately and in addition to the one proposed.

The second category is tactical conflict management, which spans a time horizon of approximately up to 10s (possibly more, for example in the case of TCAS system in aviation), up to about a second or less before impact. These solutions include active trajectory change, speed, or direction adjustment or as a last resort evasive maneuver. The proposed system implements short- to mid-term tactical management, which focuses on trajectory adjustments. All other additional safety systems, such as automatic brakes or last-minute swerve and avoid solutions act as additional safety nets operating independently from the system.

To develop the system, the following high-level requirements were defined:

- Detect and resolve conflicts in the 10s timeframe,
- Fully autonomous operation,
- Technology and solution independent,
- Wide range of users served,
- Plug and play integration,
- Modular software solution,
- XITL tools integration.

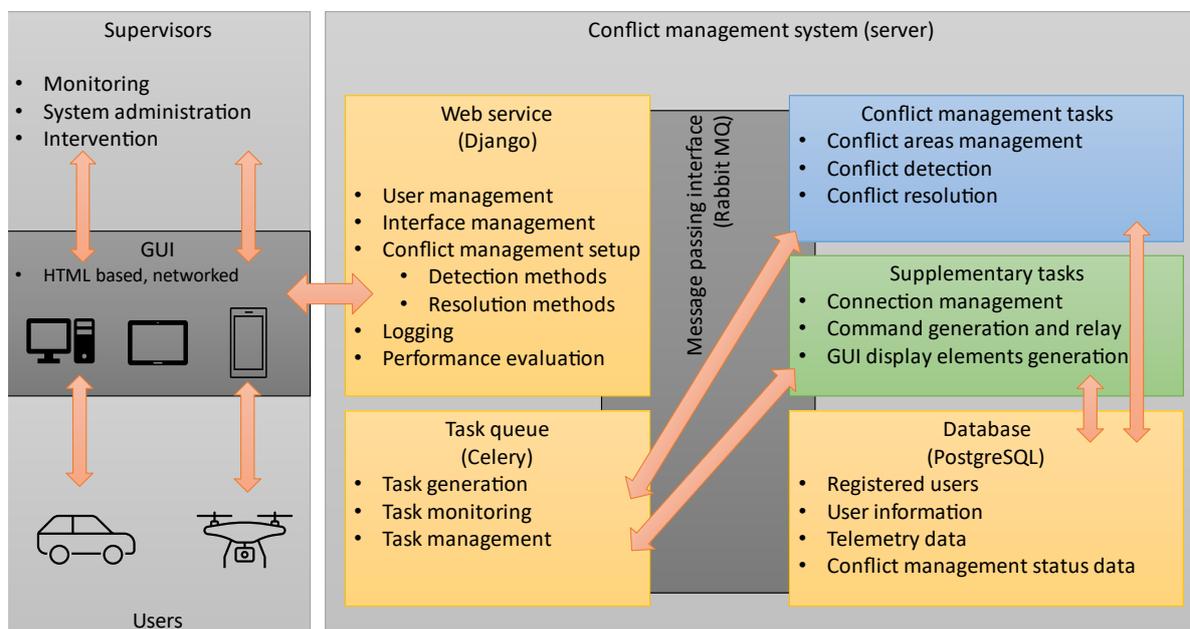


Fig. 1 Architecture of the current implementation of the automated conflict management system

Fig. 1 shows the architecture of the implemented conflict management system. The core of the system is a web service that is used to register and track system users. The users are connected to the system via their own communication solutions and are controlled using their own onboard systems; the conflict management system does not require modifications in the flight/drive controllers of individual vehicles. The users' telemetry data is passed to the system, and the conflicts are detected automatically and in real-time. Based on the type of users involved in the conflict, an appropriate resolution method is calculated, and the resolution commands are sent to the vehicles, using their own communications systems, and are executed by the onboard controller. There is a database supporting the webservice, enabling persistent data storage, and allowing robust operations, as in the case of system errors, the status of the system can be recovered quickly and efficiently, and conflict management can be continued. The system provides a web-browser based GUI, where the operation of the system can be tracked, administrative actions performed if/when required and if necessary, provides an opportunity for human intervention into the otherwise autonomous operation. Fig. 2 shows the GUI of the conflict management system running in a standard browser window, displaying 3 vehicles, their conflict detection zones and their trajectories.

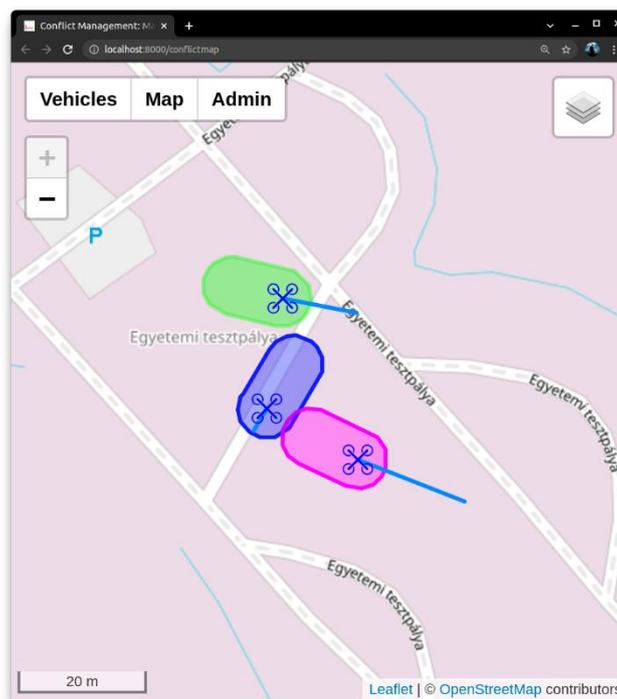


Fig. 2 Web-browser based GUI of the conflict management system depicting a 3 vehicles conflict situation at ZalaZONE

3 Conflict management development directions

3.1 Algorithm development

The current version of the prototype system has demonstrated that the tool is capable of providing a reliable and safe conflict management solution to a mix of aerial and ground vehicles. However, the system has only been demonstrated for a small number of users, and safe operation was the only concern in these demonstrations. However, in the case of a complete system, 3 primary evaluation criteria can be defined:

- Safety: The system needs to minimize the impacts and near impacts between users during operations
- Performance: The system needs to handle a large number of users
- Efficiency: The system needs to be the least intrusive; impose minimal additional time and detours

In order to fulfil these 3 criteria, the best as possible, the system must have a set of appropriate algorithms, which are tuned and optimized to handle the appropriate conflict situations. These algorithms can be selected from those available in the literature, existing methods modified or combined, or even new ones developed. Conflict management algorithms can be separated into 2 categories: detection and resolution. Classically detection methods can be categorized as deterministic, worst-case, probabilistic or hybrid methods. These methodologies have widespread literature available, and new ones are being developed as an active research field. Effective detection of conflicts is essential, as the time available to resolve conflicts at the tactical level is by its nature very short. In

the initial implementation of the system, two detection methods were implemented. The first, Pairwise waypoint-based static area detection relies on known paths that the users are following and predicts potential conflict areas in advance. It is a static solution, because as long as the trajectories stay fixed, there is no possibility to develop conflicts anywhere else and as such, it is sufficient to estimate conflict areas only once per pair of users. Due to this, the method is exceptionally efficient. However, the method does not provide the required level of safety, as in the case of unplanned actions, or path deviations due to wind, skidding, etc. the system can't detect the new conflicts. The second method is the Pairwise dynamic projected area detection, in which case potential conflict areas are dynamically updated based on the trajectories of the individual users, as can be seen in Fig 2.

Resolution methods are also well studied, generally the most common are:

- Heading changes,
- Heading and altitude changes,
- Velocity adjustment,
- State-based,
- Intent-based.

Obviously, ground vehicles are not capable of altitude changes, and any similar constraints (such as fixed-wing UAVs can't stop in mid-air) must be taken into account when the system selects the appropriate methods to manage the conflict between vehicles. In the first prototype version, the stop-and-go resolution method was implemented. In this method, the vehicle to be stopped receives a stopping point considering the projected conflict locations and its dynamics characteristics, and will stay there stopped, until the conflict is resolved. While this method is simple, safe and effective in most cases it is incapable of handling situations, where 2 vehicles come into conflict with colinear trajectories or at very shallow angles. Also, when one vehicle terminates its path in a conflict, the situation can't be reasonably resolved with this method. Evaluating the possibilities for implementing additional methods, the following solutions were chosen as the development options:

- Altitude adjustment: this method implements vertical avoidance, and by its nature, it is only applicable to control UAVs. While the method is simple to implement theoretically, in practice vertical obstacle information is required to enable safe maneuvers. Furthermore, climbing is an energy-intensive action that is usually already constraining for UAVs, and the implementation of the algorithm needs to consider the flight mechanics of the vehicles to ensure sufficient power is available for the safe execution of the maneuver.
- Horizontal detour: this method defines avoidance paths in the horizontal plane that the vehicle must follow in order to resolve the conflict. The algorithm is simple for one-to-one conflicts, however, in the case of multi-vehicle conflicts a robust planning methodology is required to ensure no additional conflicts arise from the planned detours. Also, especially in the case of multi-vehicle conflicts, the areas involved can be large, and obstacle information is desirable to ensure no obstacle-to-vehicle conflicts are generated by the resolution.

3.2 Evaluation system

In order to achieve the safety, performance and efficiency criteria, it is not only important to know if the conflicts are managed, but also how well they are managed. To answer this question, a system is required to evaluate the workings of the conflict management system. Using these evaluations, objective measurements can be made, and the system can be fine-tuned by adjusting the methods, relevant parameters, etc. The evaluation system examines the operation of the conflict management system and provides metrics in 2 categories:

- Safety measures: This includes metrics related to the safety of the provided management solution. In this category, data like the number of system users, conflicts detected, impacts and near impacts are evaluated.
- Performance measures: This category relates to the performance and efficiency of the system and contains metrics such as time and distance spent due to detours, additional energy consumed, time spent landed or parked, etc.

An ideal system should operate with 0 safety incidents and close to 0 additional time, distance and energy spent due to conflict management. While this is likely impossible to achieve in practice, the aim is to get as close to this ideal solution as possible. The XITL tools integrated into the system enable the simulation of various management solutions, so the systems can be analyzed and tested rapidly and efficiently without causing real-world accidents or losses. Due to the persistent storage and logging of data, even if accidents do occur, the reasons behind the event can be investigated, and the tools further developed to avoid repeating the same event in the future. The numerical

metrics generated by the evaluation system can be used to drive an optimization process in order to algorithmically tune the parameters of the conflict management approach. Likewise, translating the behavior of the management system into well defined numerical data allows the potential application of AI solutions within the developed framework, as a long-term goal of the system development.

4 Conclusion

This paper presented the conflict management methods development for the autonomous conflict management framework. The paper briefly highlighted the need for an autonomous combined solution to manage the high volumes of both aerial and ground-based autonomous vehicles. The key requirements towards such an automated system are listed, and the current implemented webserver-based solution of the conflict management system prototype is presented, its key elements listed and briefly discussed. The methodologies for the detection and resolution of conflicts are presented, along with brief technical background and terminology. The currently implemented detection and resolution methods are described along with their benefits and drawbacks. The new resolution methods being currently developed are shown, and their key theoretical and practical specifics are presented. In order to objectively evaluate and compare the different conflict management approaches, an evaluation system development is in progress, its key points and applicability are discussed. Any developed conflict management solution must have an acceptably high level of safety, while at the same time enabling efficient operations for a large number of users predicted.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Poponak, N.; Porat, M.; Hallam, C.; Jankowski, S.; Samuelson, A.; Nannizzi, M. Drones Flying into the Mainstream. 2016. Available online: <https://www.goldmansachs.com/insights/pages/drones-flying-into-the-mainstream.html> (accessed on 22 March 2022).
- [2] MarketsandMarkets. Unmanned Aerial Vehicle (UAV) Market by Point of Sale, Systems, Platform (Civil & Commercial, and Defense & Government), Function, End Use, Application, Type, Mode of Operation, MTOW, Range, and Region—Global Forecast to 2026. In Markets and Markets; MarketsandMarkets Research Private Ltd.: Hadapsar, India, 2021. Available online: https://www.marketsandmarkets.com/Market-Reports/unmanned-aerial-vehicles-uav-market-662.html?gclid=CjwKCAiA78aNBhAIEiwA7B76p3r10j hv4HYaPU5K6iYWi-9Uq1S_SIE_W5bEvNb2U6_unbt5QHILZhoC_FoQAvD_BwE (accessed on 22 October 2021).
- [3] Global Drone Market Report 2021–2026; Research and Markets: Dublin, Ireland, 2021.
- [4] European Aviation in 2040, Challenges of Growth; EUROCONTROL: Brussels, Belgium, 2018. Available online: <https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/reports/challenges-of-growth-2018.pdf> (accessed on 22 October 2021).
- [5] Litman, A.T. Autonomous Vehicle Implementation Predictions, Implications for Transport Planning. 2021. Available online: <https://www.vtpi.org/avip.pdf> (accessed on 22 October 2021)
- [6] European Automobile Manufacturers Association. Vehicles in Use Europe 2019; ACEA Report: Brussels, Belgium, 2019
- [7] Logan, M.J.; Bird, E.; Hernandez, L.; Menard, M. Operational Considerations of Small UAS in Urban Canyons. In Proceedings of the AIAA SciTech 2020 Forum, Orlando, FL, USA, 6–10 January 2020
- [8] Szirczák, D.; Rohács, D. Automated Conflict Management Framework Development for Autonomous Aerial and Ground Vehicles. *Energies* 2021, 14, 8344. <https://doi.org/10.3390/en14248344>

Control Design of an Autonomous Moving Platform for Test Tracks

László Kocsány^{1a}, Tibor Gergely Markovits¹

¹ *Department of Control Engineering and Information Technology,
Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics*

email: kocsany@iit.bme.hu

Abstract

In recent years, artificial intelligence, deep learning, and computer vision systems have paved the way for the development of various self-driving vehicles. The risk of testing these vehicles is not negligible, given the high kinetic energy, so testing methods should be chosen carefully. In addition to static objects, the problem of handling dynamic objects during the test cases which are carried out on test tracks as ZalaZONE. Hence dynamic objects are carried by self-driving platforms that do not cause significant material damage to either the test device or the test subject in the event of a loss of control. This paper presents two important aspects of the development of a universal Moving Platform. These are the safety analysis of the onboard architecture to ensure a highly reliable emergency stop and the trajectory tracking methods and the implementation of the associated low-level control loops.

Keywords: *autonomous platform, safety analysis, test vehicle, trajectory tracking*

1 Introduction

Autonomous cars are also known as self-driving cars, are vehicles that require significantly less human involvement for operating them. In the past few years, there has been a leap in the development of self-driving cars, but still, some human interactions are required, regulated by the so-called automation level [1]. This kind of classification of automated vehicles is done by dividing them based on the need of human interactions, which also refers to the basis of the extent of automation.

Categories are defined from 0 to 5, where Level 0 means that there is no automation, and Level 5 means that the vehicle is capable of full automation. For this level, there is no human involvement required. The vehicle will not allow the passenger to take control of the car's operations. On this level, the detection and accurate mapping of all objects around the vehicle are inevitable. The above-mentioned objects can be stationary or moving objects, self-driving cars must be able to handle each case without obstructing the traffic and endangering people's life.

The literature provides several solutions for the vehicle detection of the autonomous vehicle environment. Gluhakovic et al. [2] propose the use of the YOLO v2 algorithm to detect vehicles in an environment and to provide potential collision warnings. Zinchenko et al. [3] present an automatic collision avoidance system that can avoid both static and dynamic objects throughout the manoeuvring. These systems are tested in simulated environments, however, when the real tests are conducted, significant material damage even in case of a loss of control should be avoided.

This paper introduces a universal self-driving *Moving Platform*, developed for the test cases described above, to maintain safety during the development phase. This *Moving Platform* can emulate different types of dynamically moving objects, for example, pedestrians, motorcyclists, or even other cars.

2 External connections of the moving platform

The *Moving Platform* is a wireless test equipment for testing autonomous functions during a self-driving car development. The external connections organized by fully defined interfaces determine the capability and the different operational modes of the vehicle, for example, the range, speed, and other important aspects. The *Moving Platform* has two major operating modes:

- manual steering based on optical inspection,
- autonomous steering by onboard sensors and predefined trajectories.

The manual driving mode of the *Moving Platform* is based on a handheld controller, and the camera system, installed on the test platform. There is an emergency stop button on the right side of the controller in case of any emergency either in manual or autonomous mode.

The autonomous driving mode of the *Moving Platform* is based on a predefined trajectory and a highly accurate Differential Global Positioning subsystem (DGPS). From these inputs, the test platform evaluates the steering and accelerating/decelerating properties, implementing a moving object on the road, that the unit under test (UUT) vehicle should handle properly.

The following interfaces are specified for the *Moving Platform*:

- IF-RC: HMI interface for radio controlling the test platform for manual steering,
- IF-CAM: HMI interface for the optical inspection of the test platform with onboard camera system,
- IF-CFG: HMI interface for determining the trajectory for autonomous driving,
- IF-GNSS: wireless interface for determining the position of the test platform with high accuracy,
- IF-PS: wired interface for charging the onboard battery of the test platform.

3 Safety principles

During the development of a safety-related product, like an autonomous moving test platform, the choice of the safety architecture always depends on which part of a control system the product implements. The closer we are to the actuator controller unit, as well as to the process sensors, the more stable the system should be, i.e., in addition to safety, availability should also be taken to account. As Fig. 1 shows, the external interface could be divided into two types: the Human Machine Interfaces (HMIs) and control interfaces.

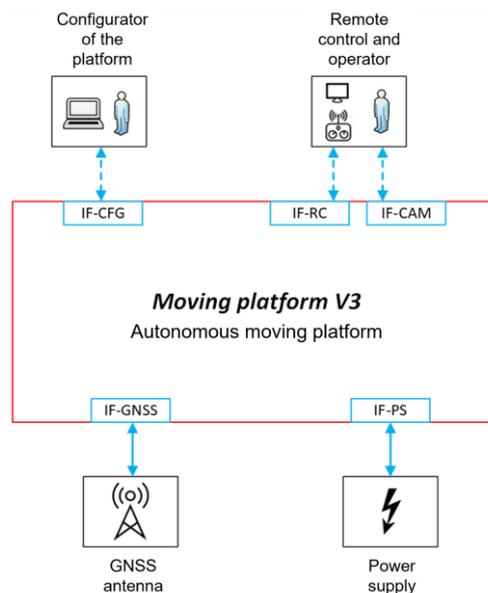


Fig. 1 External connections

In the case of the *Moving Platform*, the *1oo2* [5] safety architecture implemented as diverse redundant channels. The *1oo2* is a two-channel safety architecture, that usually consists of two identical channels for redundancy purposes. This safety architecture protects against dangerous failure with the following method: if the results differ when evaluating the two channels, the system can reach a safe state.

4 Internal architecture

The *Moving Platform* has a *1oo2* based architecture but is organized in a diverse channel implementation. The logical processing units, the actuator controller, and the physical actuator themselves are arranged in *1oo2* architecture format to maintain the safety properties of the system.

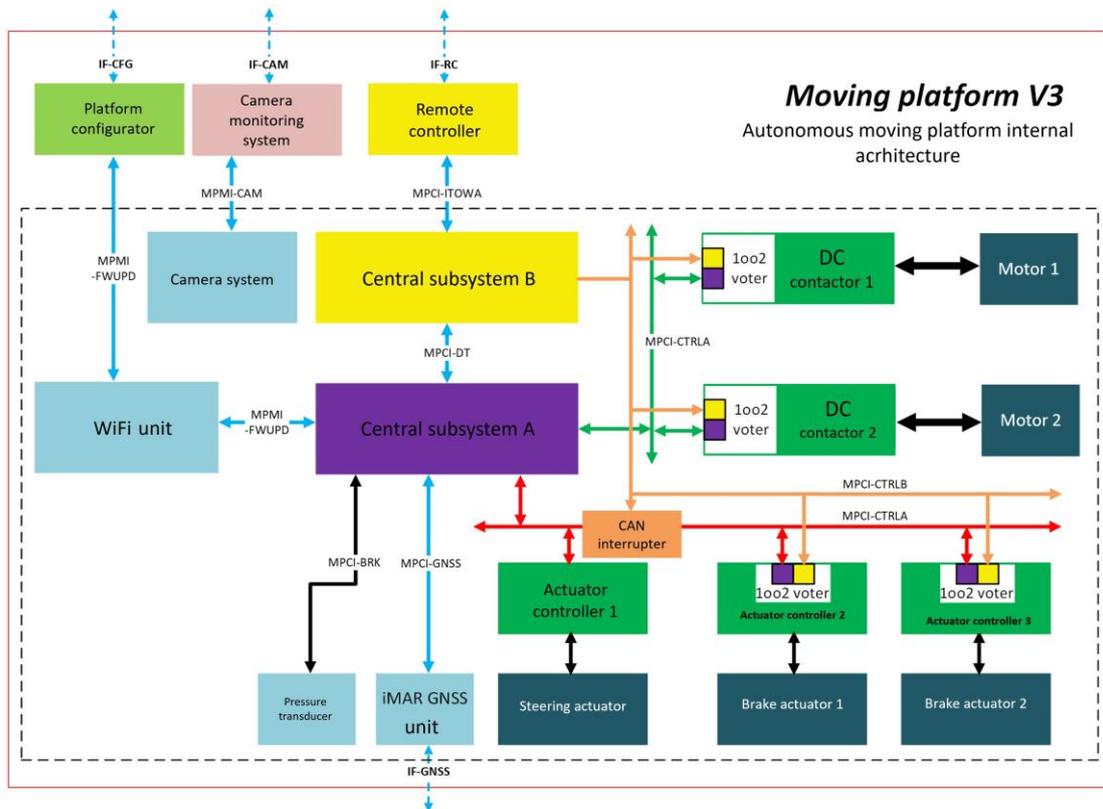


Fig. 2 Moving Platform architecture

The internal architecture of the *Moving Platform* can be found in Fig. 2. The system can be divided into two major parts: the *Moving Platform* itself, and the remote-control subsystem, which contains all the necessary hardware and software to operate the *Moving Platform* hardware remotely.

The two major parts are connected to the following wireless interfaces:

- MPMI-FWUPD: management interface for updating the predefined trajectory or the firmware,
- MPMI-CAM: management interface for transmission of the onboard camera broadcast signals,
- MPMI-ITOWA: control interface for manual driving and handling the emergency shutdown.

4.1 Moving platform subsystem

Inside the *Moving Platform*, there are two central subsystems, denoted by yellow and purple colours, implementing the *1oo2* safety architecture. These central processing units are diverse in hardware also, for increasing the safety properties of the system, but also for the different purposes of each subsystem.

The *Central subsystem A* marked by purple colour is the one that implements the different types of control loops for the autonomous operating mode. It can communicate with all the actuator controllers precisely, allowing the proper trajectory tracking. This subsystem is based on a *dSPACE MicroautoBox II* [6] embedded computer.

The *Central subsystem B* marked by yellow colour is the one that implements the wireless communication and the emergency stop function and it acts as a bridge between the handheld remote control and the *MicroautoBox II* subsystem. This subsystem is implemented by an *ITOWA IT3RBUS* device [4].

Interfaces in the moving platform:

- The MPMI-DT interface is responsible for the data transmission to the *MicroautoBox II* subsystem and checks its status. This interface has been implemented via the *CANopen* protocol, using a custom dataset configuration. The interface uses the heartbeat feature of the *CANopen* protocol, maintaining the availability of the *MicroautoBox II* subsystem. In case of a connection loss between either the handheld remote controller or the *MicroautoBox II* cause an immediate emergency stop event, sending a stop message directly to the actuator controllers. This event can still occur if the supervisor of the *Moving Platform* presses the stop button on the handheld controller.
- The MPMI-GNSS interface for receiving position information from the *iMAR GNSS* subsystem.
- MPMI-BRK: This interface collects the pressure data from the braking subsystem.

- The MPC-CTRLA interface provides control data to the actuator controllers and makes it possible to get feedback from these controllers. This protocol was implemented with a subset of the *CANopen* protocol, which has the highest priority. As a result, a CAN interrupter on this interface, so in case of an emergency stop event, it cannot send incorrect messages for the brake actuators.
- The MPC-CTRLB interface provides simply one-bit information about the emergency stop event, which is a low active signal. The energy-free state drives the *Moving Platform* into a safe mode by starting the braking process in both brake actuators.

4.2 Isolation

The *Ioo2* based safety architecture can work properly only if the two main central processing subsystems and the two brake actuators are completely separated from each other. This means, that they need separate power supplies, and at every point, where the information crosses, galvanic isolation is needed for the actual interfaces.

For the MPC-CTRLB interface using a simple relay can solve this problem, but in the case of the CAN-based MPC-CTRLA interface a CAN isolator must be inserted between the *ITOWA* Receiver and the *MicroautoBox II* subsystems. While the motor controllers are powered by 48V, which is a different power domain that the central processing subsystems have, an isolation between the motor controllers and the MPC-CTRLx interface is also needed.

5 Kinematic model of the vehicle

The high-level control loops are implemented in *Central subsystem A*, while the low-level controllers are implemented in the actuator controllers, which directly control the actuators. To formulate the control rules, the kinematic model of the vehicle should be analysed.

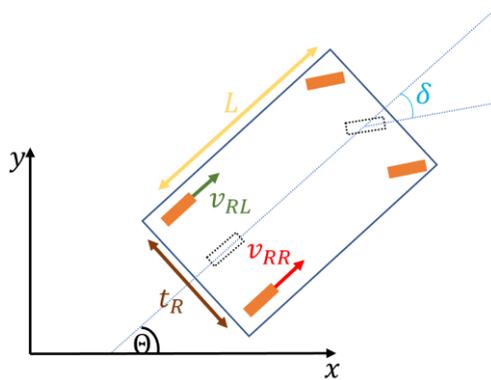


Fig. 3 Vehicle model

Fig. 3 depicts the simplified vehicle model of the *Moving Platform*. The vehicle consists of steering gear, that is driven by a *Thomson Electrak® 1 SP* [7] linear actuator, and two separately driven rear wheels, that are driven by two *Curtis 1234SE-5421* [8] motor controllers.

The state of the vehicle is depicted by $q = [x, y, \theta]^T$. The values of x, y represent the position of the vehicle, while θ is the orientation. With this simplification, the state of the vehicle can be calculated by (1), where v is the speed input of this system, and κ is the curvature of the path.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \kappa \end{bmatrix} \cdot v \quad (1)$$

To be able to calculate the κ curvature of the path by (2)-(4), the wheelbase L constant parameter, and the actual steering angle δ are required. The value of κ is limited, as the steering angle δ has a minimum δ_{min} and a maximum δ_{max} .

$$\kappa = \frac{\tan \delta(t)}{L} \quad (2)$$

$$|\kappa| \leq \kappa_{max} \quad (3)$$

$$\delta(t) \in [\delta_{min}, \delta_{max}] \quad (4)$$

6 Control loop of steering

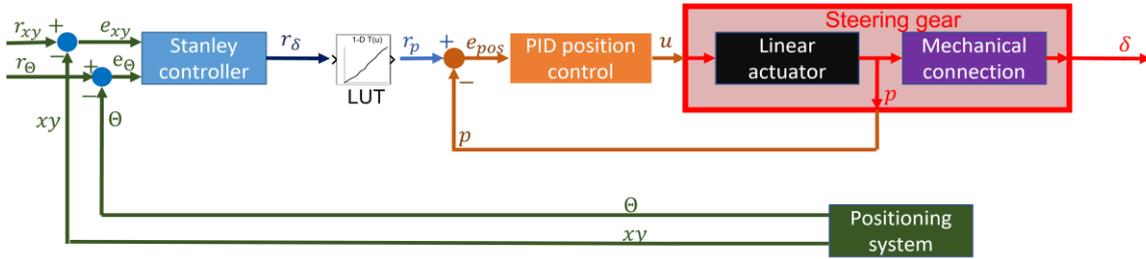


Fig. 4 Control loop of steering

Fig. 4 depicts the control loop of the steering. As it can be seen, it includes a high-level *Stanley controller* [9] and a low-level control *PID* controller. The control loop receives the reference trajectory describing the sequence of the desired configurations q of the vehicle. The desired position $r_{xy} = [x, y]$ provides the reference point of the trajectory to be reached, while r_θ is the desired orientation at point r_{xy} .

The high-level control loop calculates the deviation from the desired path. For this step, the feedback of the measured orientation θ and the position data $xy = [x, y]$ are required. Fig. 5 depicts a case, where it is not only the orientation that is differing from the desired one but there is a position deviation too. To eliminate both the heading and the tracking errors, a *Stanley controller* has been implemented. The algorithm of the *Stanley controller* is described by (5).

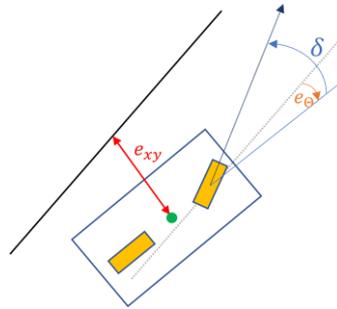


Fig. 5 Stanley controller

$$r_\delta(t) = e_\theta(t) + \tan^{-1} \left(\frac{k \cdot e_{xy}(t)}{v(t)} \right) \quad (5)$$

In (5), $r_\delta(t)$ is the output of the *Stanley controller*, providing the reference steering angle for the low-level control. The parameter k is called the position gain, which is a tuneable parameter. This value determines how much the position error affects the steering angle. The goal of the controller is to eliminate the heading error: e_θ , and to eliminate the tracking error: e_{xy} , too.

For the low-level control, a *PID* controller is tuned. The steering gear consists of a linear actuator, and a mechanical system connecting the linear actuator and the wheels of the vehicle. The position of the actuator is proportional to the voltage. However, due to the mechanical connection, there is no linear connection between the voltage and the steering angle. As a result, a lookup table (LUT) has been introduced, between the low-level and the high-level controllers of the steering, thus the low-level controller should only perform a position controlling task on an LTI-system, where the actual feedback is the position data from the linear actuator.

7 Electronic differential

The *Moving Platform* is a Rear-wheel drive vehicle (RWD), with separate in-wheel motors. Due to this design, there is no mechanical differential. Therefore, it is required to allow different wheel speeds when $|\kappa| > 0$, thus the slipping of the rear wheels can be avoided. When the vehicle model depicted in Fig. 3 is taken into consideration, and the diameter of the rear wheels d is known, the required rotational frequency of the rear wheels f_{RL} and f_{RR} can be calculated by (6)-(7).

$$f_{RL} = \frac{v(t)}{d \cdot \pi} \cdot \left(1 - \frac{t_R}{2} \kappa(t) \right) \quad (6)$$

$$f_{RR} = \frac{v(t)}{d \cdot \pi} \cdot \left(1 + \frac{t_R}{2} \kappa(t)\right) \quad (7)$$

The rear wheels are driven by two separate *Curtis motor controllers*, where two PID controllers are implemented. The motors have encoder feedback; thus, the RPMs of these motors are known. As a result, f_{RL} and f_{RR} are the reference values of the implemented control loops in the *Curtis controllers*.

8 Emergency brake control

Fig. 6 depicts the structure of the emergency brake system. Any of the *Central Subsystems*, described in Section 4, can trigger the emergency stop of the vehicle. The structure realizes a dual-circuit brake system; thus, any brake actuation provides a full vehicle stop.

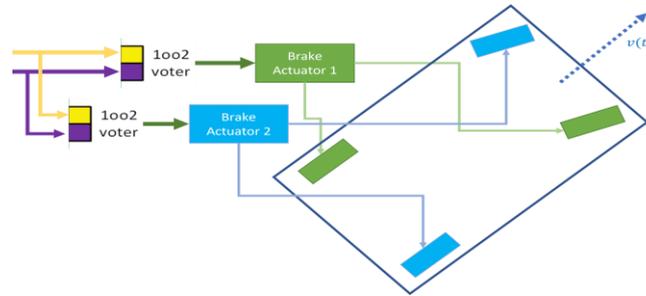


Fig. 6 Emergency brake actuation

9 Conclusion

In this paper, we have presented the control design of an autonomous *Moving Platform* that is used for test tracks. It has been shown that the designed architecture of the *Moving Platform* consists of two separate subsystems: *Central subsystem A* and *Central subsystem B*. The base principle of the presented architecture was to fulfil the high safety requirements. The low-level control loops utilize simple PID controllers, while the high-level control loops utilize more complex computations. To increase the safety metrics of the system *1oo2* voters have been realized in a diverse channel implementation, thus any *Central subsystem* is able to initiate an emergency stop procedure of the platform. Future work includes the tests of the platform both in simulations and test tracks.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, 2021, https://www.sae.org/standards/content/j3016_202104/
- [2] Gluhaković, Mario, et al. "Vehicle Detection in the Autonomous Vehicle Environment for Potential Collision Warning." 2020 Zooming Innovation in Consumer Technologies Conference (ZINC). IEEE, 2020.
- [3] Zinchenko, Serhii, et al. "Automatic collision avoidance system with many targets, including manoeuvring ones." (2019).
- [4] ITOWA Radio Remote Control Systems, <https://www.itowa.com/eng/telemandos.php>
- [5] International Electrotechnical Commission IEC/EN 61508. International standard 61508 functional safety: Safety Related systems: Second Edition; Geneva; 2010
- [6] MicroAutoBox II product information, dSpace, <https://www.dspace.com/en/inc/home/products/hw/micautob/microautobox2.cfm>
- [7] Electrak® 1 (SP) Linear Actuators, Thomson Industries, Inc, <https://www.thomsonlinear.com/en/products/linear-actuators/electrak-1>
- [8] AC Induction and PMAC Motor Controllers, Curtis Instruments
- [9] Hoffmann, Gabriel M., et al. "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing." 2007 American control conference. IEEE, 2007

Testing and Validation of the FRC08 Formula Student Race Car at the ZalaZONE Automotive Test Track

Zsolt Farkas ^{1a}

¹ *Department of Machine and Product Design, Faculty of Mechanical Engineering
Budapest University of Technology and Economics*

^a *Corresponding author: farkas.zsolt@gt3.bme.hu*

Abstract

The development and adjustment process of a race car consists of countless steps that require a lot of decisions to be made. These decisions can only be effective if we have the right amount and quality of information. A significant part of this information is provided by computer simulations, which, unfortunately, often do not cover reality due to the inaccuracy and error of the model setup. A simulation can be considered credible if certain elements of it can be replicated in reality or give almost the same results. In order to map the characteristics of the tire on the FRC08 car of the BME Motorsport Formula Student racing team, we carried out tire temperature measurements. The tests were carried out at the ZalaZONE automotive test track, which provides a unique opportunity and working environment internationally for testing and validation of simulation models. Getting to know the behaviour of the race car in as much detail as possible also revealed the strengths and weaknesses of the vehicle. This provides the team with the opportunity to adjust the optimal operating parameters of the vehicle and provides appropriate information for the development of a new racing car.

Keywords: *Formula Student, race car, tire grip, tire temperature measurements, ZalaZONE*

1 Introduction

Formula Student [7] is an international constructors' competition for university students established in 1981 by the Society of Automotive Engineers (SAE). Teams must design, build and operate a single-seat formula race car. The tournament begins with a strict machine takeover, where the regularity of the vehicle is examined. Next are the static competitions (Engineering design, Cost & Manufacturing, Business plan presentation) If the car has passed the machine takeover, it can participate in dynamic competitions (Skid-pad, Acceleration, Autocross, Endurance).

The first Formula Student team at the Budapest University of Technology and Economics, the BME Formula Racing Team [8] has been building electric and driverless vehicles for years. Thus, in the autumn of 2018, it was possible for the students and teachers of the University to create the second Formula Student team competing in the internal combustion engine category, BME Motorsport [9]. In addition to the successful racing, the secondary goal of the FRC-08 car was to be able to perform as many measurements and tests as possible on it. This decision proved to be worthwhile, as the FRC-08 became the second most successful internal combustion engine car in the history of the Hungarian Formula Student. Overall, the team scored 238% more points than in the first season.

The development and adjustment process of a race car consists of countless steps that require a lot of decisions to be made. These decisions can only be effective if we have the right amount and quality of information. A significant part of this information is provided by computer simulations, which, unfortunately, often do not cover reality due to the inaccuracy and error of the model setup. A simulation can be considered credible if certain elements of it can be replicated in reality or give almost the same results.

One of the most important components of the race car is the tire, as the power emitted by the engine is converted into traction through the contact patch of the rubber and asphalt. Maximizing the grip of the rubber is one of the most important factors in terms of vehicle performance. This is supported by the lap time simulation used very

often in motorsport [2]. In such a simulation, by examining the model of the vehicle, it is determined what lap time it can perform on a given racetrack.

For the simulation study, we used the Optimum Lap [6] software. The model shall be constructed on the basis of known vehicle parameters (engine characteristics, downforce, weight, etc.). During the test, the value of each parameter was increased and reduced by 10% in three steps, the percentage of changes to the lap time by changing the given parameter was determined by increasing and decreasing. Based on the results of parameter sensitivity studies related to simulations [4], the most important thing is to increase grip. One of the main reasons for this is that the engine of the race car has high power and the tires cannot transfer such power to the asphalt. On average, a 10% increase in grip improved lap time by 3.9%. Therefore, in order to increase wheel grip, it is worth further analyzing the characteristics that determine it in detail and validating the results of the simulation tests. Therefore, in order to map the characteristics of the tire on the race car, we carried out rubber heat measurements, as well as validations of the simulation model.

2 Materials and methods

The adhesion shall be greatest if the contact surface of the rubber and asphalt is maximum and the pressure distribution is uniform on it. The side force causes the rubber to be deformed and this contact surface deformed, it becomes smaller than the static state. This effect can be offset for the given lateral force and normal force by camber provided by chassis characteristics [1]. The causes of heat in the tire are deformation (hysteresis energy loss) and tangential interaction of rubber-road (heat generation of surface friction). Heat formation is greatest at the exiting edge of the contact surface. Thanks to this property, by measuring the surface temperature of the rubber, we can infer the current pressure distribution of the contact surface [3]. This allows us to determine and analyze the relationship between the vehicle's current camber and grip in different driving and load conditions.

The current temperature distribution on the surface of the tire shows which part of the wheel is in contact with the road with what pressure distribution. The higher the normal force, the higher the temperature will be. And where the normal force is lower, the temperature will be lower. If one part of the tire works better than the other, it creates less grip than if all parts of it received the same load.

The change in temperature is not only caused by the current camber. Another important factor is tire pressure. During the tests, this parameter can be excluded, because measurements were always started at the same temperature, cold state and at the same tire pressure. For the tests, the FRC08 race car was fitted with three infrared sensors for contactless temperature measurements per wheel. One measured the temperature of the outer part of the rubber, one on the inside and one in the middle. Figure 1 shows a race car equipped with rubber temperature sensors.



Fig. 1 FRC08 race car equipped with sensors at ZalaZONE test track

The measurements were made by turning right and left with three different 15°, 25° and 35° steering angles and the corresponding 35m, 20m and 12m turning radius. The measurement was carried out near the adhesion limit, at a working point where both the vehicle speed and the steering angle could be kept at a near constant value. During the tests, based on the results of the preliminary simulation tests, three different camber values were set for the front axle -2°, -1.3° and -0.6° and -1.3°, -0.6° and 0° for the rear axle. When adjusting the angle of the front axle, the rear remained at -1.3°. And when adjusting the angle of the rear axle, the front remained at -2°.

The tests were carried out at the ZalaZONE automotive test track, which provides a unique opportunity and working environment internationally for testing and validation of simulation models.

3 Results

The evaluation of the measured data was helped by the creation of a Matlab script. The diagrams (Fig. 2, Fig. 3) show the results of the values of the tire temperature sensors, the turning radius, the steering angle, the chassis roll, and temperature differences. On the left tire, the data of the central temperature measuring sensor are shown in red and on the right tire in green. The data of the inner tire sensor (towards the chassis) are indicated by the black, and the data of the outer tire sensor are indicated by the grey curves. In the diagram showing temperature differences, red with left and green can be seen as the difference between the external and internal temperatures of the right wheel.

Figure 2 shows the results of the front axle for turn left (Fig. 2/a) and turn right (Fig. 2/b) with the steering angle of 15° and with -2° front axle and -1.3° rear axle camber. In the case of turn left (Fig. 2 / a), the temperature of the left tire is a few degrees higher than the right tire. The temperatures of the right tire move together in time. From this we can conclude that the distribution is roughly even, the camber may be close to ideal. In the case of the left tire, however, during the measurement, the inside and outside temperature values are greatly separated. The inner side heats up, and the outer side of the tire cools. From this, it can be concluded that the pressure distribution is drastically shifted towards the inner side of the wheel. Where appropriate, so much so that the outer surface of the wheel no longer, or only minimally, comes into contact with the road at certain points. This can be explained by the large negative camber due to steering, which is further helped by the negative static camber.

In the case of the right turn (Fig. 2/b), the temperature of the outer part of the left wheel increased to such an extent that it exceeded the inner part by the last third of the measurement. This means that the pressure distribution is shifted towards the outer half of the tire, which indicates a positive camber. This result fully supports the critical situation that caused the positive camber based on the results of previous studies. Looking at the right tire, we can say the same thing as in the case of the turn left. The outer temperature decreases, while the inner one increases and the pressure distribution is drastically shifted towards the inner side of the tire.

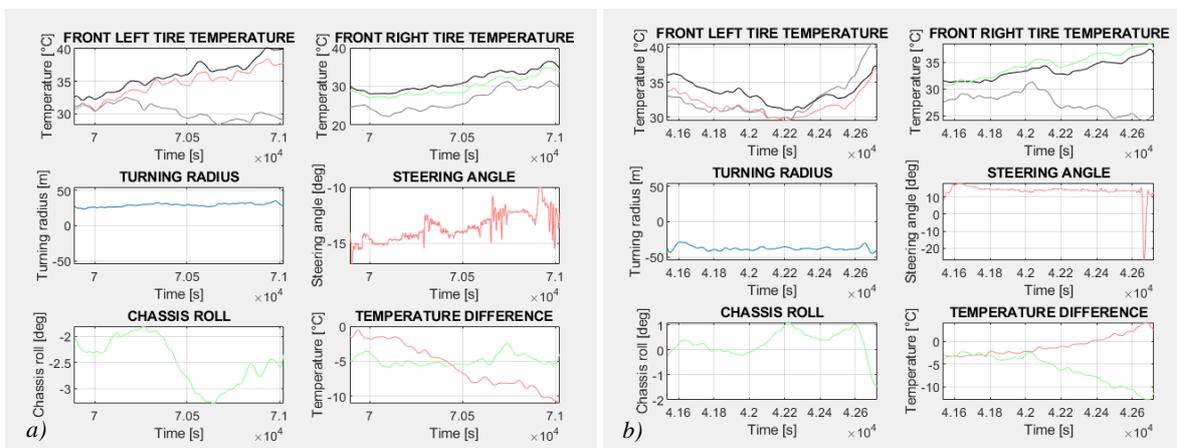


Fig. 2 Results of the front axle for turn left (a) and turn right (b) with the steering angle of 15° and with -2° front axle and -1.3° rear axle camber

The middle sensor on the right rear wheel, unfortunately, received moisture and broke down, which affected the measurements on the wheel and the filtering of the results and thus the evaluation of the results. Figure 3 shows the results of the rear axle for turn left (Fig. 3/a) and turn right (Fig. 3/b) with the steering angle of 15° and with -2° front axle and -1.3° rear axle camber. In the case of turn left (Fig. 3/a), the inner half of the right tire is warmer, followed by its outer surface at an almost constant value distance. The pressure distribution is even; no positive camber is formed. In the case of the left tire, the phenomenon seen in the front tires can also be observed. The inner half of the tire warms up, while the outer half begins to cool. It is smaller and slower than in the case of the front tire. This may be due to the fact that the rear wheels are not steered and there is no drastic negative change in camber. In the case of turn right (Fig. 3/b), the inner half of the left tire is warmer than the outer and the two curves follow each other. In the case of the right tire heating of the inner half of the wheel and cooling of the outer half can also be observed.

The static camber changes resulted in the heating of the inner half of the tire and cooling of the outer half, which supported the results so far. In addition, the gradients of the various static camber settings were very similar, so the modification did not have a large effect on the temperature change.

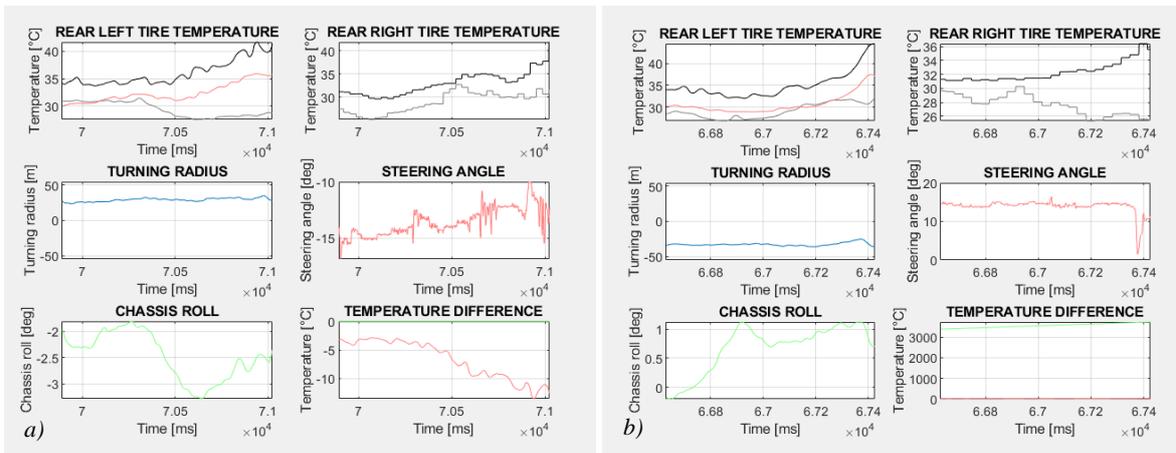


Fig. 3 Results of the rear axle for turn left (a) and turn right (b) with the steering angle of 15° and with -2° front axle and -1.3° rear axle camber

However, the evaluation of the results resulted in further questions. How much temperature difference can lead to complete loss of grip? How much side acceleration can result in a decrease in temperature per unit? To answer these questions, the team plans to make new measurements in the future.

4 Conclusion

After evaluating the measurement results, we can say that they support the results of the simulation tests of the camber calculation. When turning right on the left front wheel, the wheel produces a positive camber. For the other tires, the negative camber remains, and in any case, the inner half of the tire is warmer. On the inner tires of the turn, we noticed a significant increase in temperature differences, both on the front and rear axles. The inner half of the inner tires of the turn warmed up, and the outer part began to cool. This was probably caused by a negative camber of higher value. Based on the comparison of temperature differences, the gradient of differences is not greatly affected by the change in static camber.

Getting to know the behaviour of the race car in as much detail as possible also revealed the strengths and weaknesses of the vehicle. This provides the team with the opportunity to adjust the optimal operating parameters of the vehicle and provides appropriate information for the development of a new racing car. The results of the tests carried out on the ZalaZONE automotive test track provided an opportunity and relevant information for future tests and for the development of the suspension [4] and damping system [5] of the new racing car.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Serra, L. (2005). Optimised Contact Patch System on an F3000 car. *AutoTechnol* 5, 48–50.
- [2] Trzesniowski M. (2010) *Rennwagentechnik: Grundlagen, Konstruktion, Komponenten, Systeme*. ATZ/MTZ-Fachbuch.
- [3] Holloway C. (2015). White Paper: Influence of Temperature on Tire Grip.
- [4] Dén D. (2021). Az első magyar hibrid Formula Student versenyautó futóművének tervezése. BME
- [5] Mihók M. (2021). Formula Student versenyautó lengéscsillapító rendszerének tervezése, BME
- [6] <https://optimumg.com/product/optimumlap/>
- [7] <https://www.fsaonline.com/>
- [8] <https://frt.bme.hu/>
- [9] <https://www.bmemotorsport.com/>

Collaboration Possibilities for Autonomous Industrial Transport Vehicles

Gábor Bohács^{1a}, András Máté Horváth¹

¹ *Department of Material Handling and Logistic Systems,
Faculty of Transportation Engineering and Vehicle Engineering,
Budapest University of Technology and Economics*

^a *Corresponding author: gabor.bohacs@logisztika.bme.hu*

Abstract

Autonomous industrial transport vehicles are already in operation in the industry, but the area is very rich in further development topics. Collaboration among these vehicles makes up a special problem. First, this paper surveys the current area in detail. There is not only collaboration among the same type of vehicles like platooning discussed, but also the special tasks that are necessary for material handling operations. Second, proper conclusions are drawn for the elaborated concept of our former research regarding the possibilities of enhancement towards vehicle collaboration and its conditions.

Keywords: *autonomous transport vehicles, forklifts, material handling*

1 Introduction

Autonomous transport systems are very common in the industry. Their main use is to increase the material handling capacity at companies. This is also an essential development to counter the lack of forklift drivers. As acting like integrated components of the whole production system their application largely increases overall efficiency. Currently, the most applied type of equipment in the autonomous transport system segment is the automated guided vehicles (AGVs). These machines operate adequately safely due to their detailed standardization [1]. Despite these are already in operation for more than 50 years their popularity is still ongoing. The most important feature of them is that these are following a well-defined route without inbuilt intelligence to avoid for example unexpected obstacles. Due to the novel requirements of Industry 4.0 systems regarding increased complexity, in some systems, the applicability of the AGVs is not sufficient. This market gap is filled by the autonomous mobile robots, which main characteristics compared to AGVs is described in [2]. We can point out here that the main advantage of AMRs (autonomous mobile robots) is that these can also operate together with people unrestricted. This has however come with a cost, AMRs are more limited in capacity and speed compared to AGVs.

Despite the long history of autonomous systems in operation, there is still active research in this area (see [3] and [4]). Actual research approaches are for example the fusion of AGVs and collaborative robots; the use of machine-to-machine communication to integrate AGVs and the manufacturing environment and AI-driven analytics focused on the data produced and consumed by AGVs [5].

The current paper focuses on an actual topic that is also very relevant for the praxis. The area of optimizing a single autonomous mobile robot in a factory environment is already a well-researched area. Besides, there are many papers like [6] where a swarm of same-type robots are optimized. The researched systems are no longer restricted to indoor applications, there are papers focusing on outdoor use that are relevant as well [7]. Our focus is to assess the various types of collaborations among different vehicle types, used during industrial transport tasks. Finally based on the findings we set up a concept based on our former research.

2 Vehicle collaboration for industrial transport vehicles

However the collaboration possibilities among various types of autonomous vehicles and mobile robots are almost unlimited, there are typical applications that have both scientific and practical relevance. The first application is the collaboration among the same type of vehicles. We don't mention here the most obvious task of

avoiding each other in order to fulfil the transport requirements. An interesting approach however is the direct, physical and informatic connection of vehicles from the same type to achieve the handling ability of greater loads. Authors of paper [8] present a new class of Intelligent and Autonomous Vehicles (IAVs), from the Intelligent Transport in Dynamic Environments (InTraDE) project. This type of vehicle is technologically superior in many respects to existing automated guided vehicles. They offer greater flexibility and intelligence to manoeuvre in confined spaces where logistics operations take place. This includes the ability to pair/disassemble, allowing 1-TEU (20-foot equivalent unit) IAVs to dynamically mate, transport containers of any size between 1-TEU and 1-FFE (40-foot equivalent unit), and disassemble again. The deployment of IAVs helps port operators to efficiently cope with the ever-increasing container traffic in ports and avoid the need to deploy multiple 40-foot transport vehicles in very confined port areas.

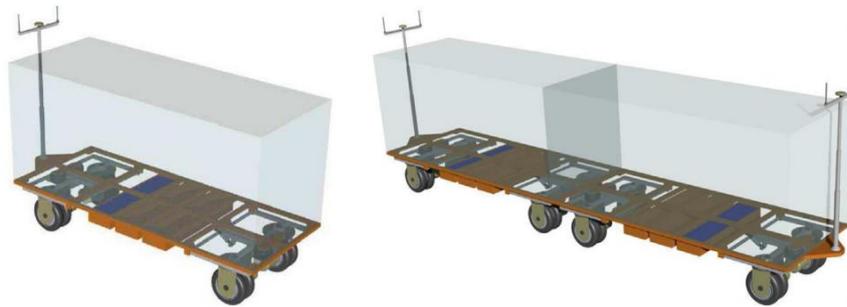


Fig. 1 Intelligent and Autonomous Vehicles (IAVs) for container transportation of containers of different sizes [8]

There is also an increasing number of applications for the collaboration of different vehicle types. In industrial transport, the most obvious task is to automate the loading and unloading of transport vehicles using forklift type machines. In most cases the carrier vehicle is not automated, it is like a conventional, human-driven truck [9]. In this case, collaboration is also needed between machine and human in order to enable the start and finish of automated loading. A good example is presented in [10] in which the loading forklift AGV autonomously operates together with a truck. It should be underlined that in this case, the autonomous vehicle is not a special design, it is based on the transformation of a conventional forklift truck, upgraded by intelligent components (see Fig. 2.).



Fig. 2 Autonomous forklift for truck loading [10]

An interesting collaboration concept is presented in [11] in which a mobile robot, with the use of its sensors and robotic arm, moves a conventional pallet jack, which is also operated by a human. The worker carries out the loading and unloading operations, while the robot executes pallet transport. This is again an example of human-robot collaboration in material handling using the same transport equipment.

Finally a well-researched area should be mentioned: platooning. This has particular significance in road transport where closely driven interconnected trucks can achieve saving on fuel. The authors of the paper [12] focus on a different area – construction site traffic – where the platooned manner improves safety, security and efficiency, control overall traffic flow and reduces resource usage.

As a summary, we concluded that practical applications in the discussed area can have very different structures. The use of already existing machines is common, thus human collaboration can more easily be achieved.

3 Description of the concept

In our former research [13] a low-cost vision-based towing truck has been developed and tested for factory site material transport operations. The towing truck had formerly been tested during the development in the ZalaZONE proving ground. Formerly we have also researched methods for material loading and unloading [14] using a forklift. From these and from the results of the literature analysis a new concept for autonomous vehicles' collaboration has been worked out. Here following principles have been identified:

- The solution should be cost-effective because today it is the most decisive factor regarding implementations. That means for us primary that we would place intelligent and costly units only where it is necessary.
- As vehicles we intend to use high-capacity, industrial class products for development basis.

The elaborated concept is depicted in Fig. 3. High capacity is secured using a towing truck and trailers. The towing truck will be equipped only with the necessary, basic navigation functionality, which is supplemented by basic safety features. Besides the towing truck can implement platooning function. This is the so-called “strong” component of the system. Its capacity can be multiplied by using double tugger trains following each other.

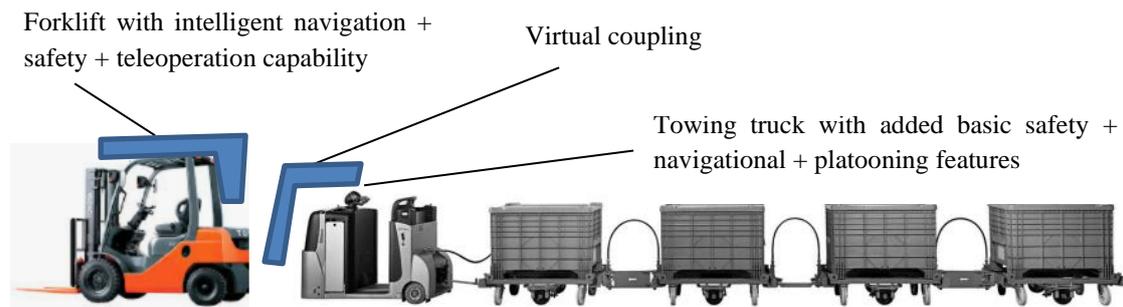


Fig. 3 Collaborative vehicle concept for factory material handling tasks

Another part of the system is an automated forklift, which is the “smart” component. It is equipped with a more intelligent navigational component, a more capable safety system and also a teleoperating capability. The solution has a single disadvantage: the intelligent forklift travels together with the towing truck. However, it is countered by the multiple advantages, which are as follows. There is no need to duplicate the forklift unit at the material source and drain stations, therefore a single unit is enough. Using this travelling loading machine, mid-stops can be easily implemented, making the to be implemented tasks more versatile. By having a teleoperation capability the forklift can be assisted during loading, unloading and travelling as well.

4 Conclusion

In this paper, a novel vehicle collaboration concept has been presented using the literature analysis and applying rational principles. The concept's main novelty is the separation of the system components into high-capacity and smart units, which enables several advantages. Future research will be aimed to assess applicability details and exact definition of necessary functionalities. There is also an ambition to test the functionality using real vehicles. For this purpose existing vehicles using the ZalaZONE infrastructure would be ideal. In this case a manually driven vehicle can be the leader and the follower can be an already automated towing truck.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

- [1] Industrial trucks. Safety requirements and verification. Part 4: Driverless industrial trucks and their systems (ISO 3691-4:2020)
- [2] Fedorko, G., Honus, S., Salai, R. (2017). Comparison of the Traditional and Autonomous AGV Systems. MATEC Web of Conferences 134, 00013 (2017) DOI: 10.1051/mateconf/201713400013
- [3] Bader, M., Richtsfeld, A., Holl, W. (2015). Balancing Centralized Control with Vehicle Autonomy in AGV Systems. Conference on Autonomic and Autonomous Systems (ICAS)At: Rome, Italy.
- [4] Draganjac, I., Petrović, T., Miklić D., Kovačić Z., Oršulić, J. (2020). Highly-scalable traffic management of autonomous industrial transportation systems. Robotics and Computer-Integrated Manufacturing, Volume 63, ISSN 0736-5845, <https://doi.org/10.1016/j.rcim.2019.101915>.
- [6] Cupek R. et al. (2020) Autonomous Guided Vehicles for Smart Industries – The State-of-the-Art and Research Challenges. In: Krzhizhanovskaya V.V. et al. (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12141. Springer, Cham. https://doi.org/10.1007/978-3-030-50426-7_25
- [7] Jinwen Chen, Xiaoli Zhang, Xiafu Peng, Dongsheng Xu, Jincheng Peng (2022). Efficient routing for multi-AGV based on optimized Ant-agent. Computers & Industrial Engineering, Volume 167, ISSN 0360-8352, <https://doi.org/10.1016/j.cie.2022.108042>.
- [8] Clauer, D., Fottner, J., Rauch, E., Prüglmeier, M. (2021). Usage of Autonomous Mobile Robots Outdoors - an Axiomatic Design Approach. Procedia CIRP, Volume 96, Pages 242-247, ISSN 2212-8271, <https://doi.org/10.1016/j.procir.2021.01.081>.
- [9] Gelareh S., Merzouki, R., McGinley, K., Murray, R. (2013). Scheduling of Intelligent and Autonomous Vehicles under pairing/unpairing collaboration strategy in container terminals. Transportation Research Part C: Emerging Technologies, Volume 33, Pages 1-21, ISSN 0968-090X, <https://doi.org/10.1016/j.trc.2013.04.006>.
- [10] Kono, T., Ogawa, K., Hirogaki, T., & Aoyama, E. (2010). Autonomous Distributed AGV System Based on Taxi Transportation Strategy: Effect of Multiple-Load AGVs on Conveyance Efficiency. ASME 2010 International Manufacturing Science and Engineering Conference, Volume 2. doi:10.1115/msec2010-34088
- [11] Bouguerra, A., Andreasson, H., Lilienthal, A.J., & Åstrand, B. (2009). MALTA: A System of Multiple Autonomous Trucks for Load Transportation. (2009). Proceedings of the 4th European conference on mobile robots (ECMR) / [ed] Ivan Petrovic, Achim J. Lilienthal, p. 93-98
- [12] P. Balatti, F. Fusaro, N. Villa, E. Lamon and A. Ajoudani, "A Collaborative Robotic Approach to Autonomous Pallet Jack Transportation and Positioning," in IEEE Access, vol. 8, pp. 142191-142204, 2020, doi: 10.1109/ACCESS.2020.3013382.
- [13] Javed, M. A., Muram, F. U., Punnekkat, S., Hansson, H. (2021) Safe and secure platooning of Automated Guided Vehicles in Industry 4.0. Journal of Systems Architecture, Volume 121, ISSN 1383-7621, <https://doi.org/10.1016/j.sysarc.2021.102309>.
- [14] Bohács, G., Horváth, A. M., Gáspár, D. (2021). Development of a low-cost material handling vehicle solution using teleoperation and marker recognition. Proceedings of the 11th International Conference on Logistics, Informatics and Service Sciences. ISBN 978-981-1686-55-9
- [15] Bohács, G., Rózsa, Z., Bertalan, B. (2020). Mono camera based pallet detection and pose estimation for Automated Guided Vehicles. Proceedings of the 10th International Conference on Logistics, Informatics and Service Sciences. ISBN/GTIN978-981-3343-58-0.