

GMM-Fachbericht **108**

AmEC 2024

Automotive meets Electronics & Control

15th GMM GMA Symposium
March 14 – 15, 2024 in Dortmund



Automotive meets Electronics & Control

AmEC 2024

Automotive meets Electronics & Control

15th GMM GMA Symposium, March 14 – 15, 2024 in Dortmund

www.ame-konferenz.de

Tagungsleitung:

Prof. Dr. Michael Wahl, Universität Siegen & Digital University Kerala

Prof. Dr. Martin Keller, Fachhochschule Südwestfalen

Prof. Dr. Georg Schildbach, Universität zu Lübeck

Veranstalter:

VDE/VDI Gesellschaft Mikroelektronik, Mikrosystem- und
Feinwerktechnik (VDE/VDI GMM)

Bibliographic Information of the German National Library

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at <https://portal.dnb.de/>.

ISBN 978-3-8007-6284-2 (CD-ROM)

ISBN 978-3-8007-6285-9 (E-Book)

ISSN 1432-3419

© 2024 VDE VERLAG GMBH · Berlin · Offenbach, Bismarckstraße 33, 10625 Berlin
www.vde-verlag.de

All rights reserved. Any utilization in breach of the strict limits of copyright law, without the prior approval of the publisher, is prohibited. Reproductions of common names, brand names, trademarks etc. in this publication are not subject to the acceptance that these names could be regarded as free or could be used by anyone, even without particular marking, in the sense of the trademark and brand protection legislation. Publication does not imply that the solutions described are not protected by intellectual property rights (e.g. patents and utility models). The publisher assumes no liability for the correctness and practicability of the programs, circuits, and any other arrangements and instructions published, nor for the correctness of the technical content of this publication. The up-to-date valid versions of the relevant statutory and official regulations and technical regulations (e.g. VDE body of regulations) have to be respected.

CD Production: DMS – Disk Media Service, Berlin
Produced in Germany

Preface

AmEC 2024 – Automotive meets Electronics & Control

Last year we decided to merge the two conferences Automotive meets Electronics and AUTOREG. Both conferences share many topics but there were also different foci. This joint conference covers a wider field.

The first core field to mention is on applications of artificial intelligence/machine learning (AI/ML) to many different domains. Obviously, the existing algorithms are not sufficient to solve the problem of dynamically changing environments, whereas AI/ML with their inherent flexibility are much more promising. The second core field is trajectory planning, where of course a strong interaction exists with AI/ML. Additional topics cover reliability, fuel cells, sensor technology and new computer architectures which become important for automotive.

The AmEC is a conference featuring scientific papers as well as industrial keynotes, providing an excellent mixture of both worlds and thus the unique opportunity to an effective cross fertilization.

Automotive meets Electronics & Control and we are looking forward to meeting you in Dortmund!

Your Conference Chairs

Prof. Dr. Martin Keller
Fachhochschule Südwestfalen

Prof. Dr. Georg Schildbach
Universität zu Lübeck

Dr.-Ing. Michael G. Wahl
Universität Siegen
Digital University Kerala

Program Committee

Scientific conference management

G. Schildbach, Universität zu Lübeck
M. Keller, Fachhochschule Südwestfalen
M. Wahl, Universität Siegen & DUK, Kerala

Organizational conference management

Ronald Schnabel, VDE/VDI GMM, Offenbach am Main

Programmkomitee

D. Abel, RWTH Aachen University
B. Alt, Robert Bosch GmbH, Renningen
A. Becciu, Nuraxys GmbH, Overath
K. Belhoula, Continental Automotive Technologies GmbH, Wetzlar
N. Beringer, CARIAD SE, Nürnberg
T. Bertram, Technische Universität Dortmund
R. Denkemann, Aptiv Services Deutschland GmbH, Wiehl
P. Farber, Robert Bosch GmbH, Reutlingen
M. Fohlmeister, Cognizant Mobility GmbH
S. Frei, Technische Universität Dortmund
E. Kamau, Technische Hochschule Köln
M. Keller, Fachhochschule Südwestfalen, Hagen
L. Kiltz, ZF Friedrichshafen AG, Friedrichshafen
T. Liebetrau, Infineon AG, Nürnberg
R. Montino, Elmos AG, Dortmund
R. Obermaisser, Universität Siegen
H. Pu, XCMG European Research Center GmbH
C. Ress, Consultant, Aachen
S. Sattler, Friedrich-Alexander-Universität, Erlangen-Nürnberg
A. Schäfer, KYOCERA AVX Components GmbH, Werne
G. Schildbach, Universität zu Lübeck
R. Schnabel, VDE/VDI GMM, Offenbach am Main
D. Schramm, Universität Duisburg-Essen, Duisburg
A. Stockem Novo, Hochschule Ruhr-West, Mülheim
N. Wagner, Adam Opel AG, Rüsselsheim
M. Wahl, Universität Siegen & DUK, Kerala
R. Wille, Consultant, Bönen
A. Zlocki, fka GmbH, Aachen

Contents

Automotive Architectures & Machine Learning

Session Chair: Georg Schildbach

- 01 Physics Informed Deep Learning for Motion Prediction in Autonomous Driving..... 7**
 P. Tischmann, R. Baumann, A. Stockem Novo, University of Applied Sciences Ruhr West,
 Mülheim a. d. Ruhr
- 02 Confidence Tuned Localization through Learning in the Loop 13**
 S. Schütte, T. Bertram, TU Dortmund; M. Kuhn, ZF Automotive Germany GmbH, Düsseldorf

Advanced Control Strategies & Trajectory Planning

Session Chair: André Schäfer

- 03 Application of Basis-Splines for Trajectory Planning in Highway Scenarios..... 18**
 P. Dorpmüller, T. Bertram, TU Dortmund; T. Schmitz, N. Bejagam, ZF Automotive Germany
 GmbH, Düsseldorf
- 04 On the Design of Interaction-Aware SCMPCC for Highway Merging Scenarios 24**
 R. Kensbock, G. Schildbach, University of Lübeck

Poster Session

Session Chair: Martin Keller

- 05 Coaction between Automobiles and Mobile Robots - Interoperability for Affordable Last
 Mile Delivery Solutions 30**
 M. Y. Khandelwal, S. Tendulkar, G. A. Kolbai, F. Schrödel, Schmalkalden University of Applied
 Sciences, Schmalkalden
- 06 AI-Based Localization and Classification of Visual Anomalies on Semiconductor Devices..... 36**
 M. K. Le, J. Z. J. Chia, D. Peskes, Elmos Semiconductor SE, Dortmund,
- 07 Machine learning for improving the trustworthiness of sensors..... 41**
 G. Hussain, L. G. Thekkumthala, P. A. William, M. G. Wahl, University of Siegen
- 08 Vision-based Autonomous Trajectory Drifting using Deep Reinforcement Learning 47**
 F. Domberg, B. Barkow, G. Schildbach, University of Lübeck

Infrastructure and Safety

Session Chair: Bendikt Alt

- 09 Challenges of Infrastructures for autonomous Buses in Cities: A review..... 53**
 A. Becciu, Nuraxys GmbH, Overath; E. N. Kamau, University of Applied Sciences Cologne

10 Automated failure and tolerance analysis as a combined consideration for the proof of safety of electronic systems.....	58
R. Müller-Hainbach, L. Ergün, S. Butzmann, University of Wuppertal	

Machine Learning/Deep Learning in the automotive context

Session Chair: Alessandro Becciu

11 CSAM anomaly detection with AI.....	63
J. C. Z. Jie, R. Krumm, Elmos Semiconductor SE, Dortmund	

12 Investigation of the real-time feasibility of NMPC for air-path control in automotive fuel cell systems	67
T. A. Nguyen, V. Neisen, D. Abel, RWTH Aachen University, Aachen	

13 Time-Triggered Organic Computing Architecture for Autonomous Driving Vehicles Using List Scheduling.....	73
M. Qosja, S. Meckel, R. Obermaisser, University of Siegen	

Future of Transportation

Session Chair: Edwin Kamau

14 Robust Navigation of Autonomous Transport Units in the Extractive Industry	79
D. Benz, D. Abel, RWTH Aachen University, Aachen	

Innovations in Electronics

15 Self-Locked Asynchronous Controller for RISC-V Architecture on FPGA.....	84
F. Deeg, S. M. Sattler, Friedrich-Alexander-University Erlangen-Nuremberg	

16 Integration of a 77GHz automotive radar system into plastic surfaces using MID-technology	89
T. Mager, J. Diri, Fraunhofer Research Institute for Mechatronic Systems Design IEM, Paderborn; P. Kneuper, S. Kruse, C. Scheytt, Paderborn University, Paderborn	

Physics Informed Deep Learning for Motion Prediction in Autonomous Driving

B. Sc. Patrik Tischmann^{a*}, M. Sc. Robin Baumann^a, and Prof. Dr. Anne Stockem Novo^a

^aUniversity of Applied Sciences Ruhr West, Institute of Computer Science, Duisburger Str. 100, 45479 Mülheim a. d. Ruhr, Germany

*Corresponding author: patrik.tischmann@edu.ruhr-uni-bochum.de

Abstract

This paper examines a physics-informed artificial neural network and its ability to learn to forecast car motion sequences using recorded real-world traffic sequences from the Argoverse dataset. It implements the Intelligent Driving Model (IDM), which models straight driving and is derived analytically and combines a data-driven model, a Long-Short-Term Memory (LSTM) neural network at the loss function level. Three networks, the standalone IDM, the LSTM Network, and their combination, a Physics-Informed Neural Network (PINN), are examined and compared in their performance and convergence rate, as well as inspected at the level of individual, visualized traffic scenes. Furthermore, this work covers the implementation details and challenges for the IDM, the combined Physics-Informed Neural Network, and how the data was prepared. We find that the inclusion of the simpler IDM into the training of the LSTM network yields a better initial performance as well as a stabilized model performance during training, we also find, however, that the IDM requires more extensive integration and preparation for working with the chosen Dataset since its simplicity does not lend itself to easily provide applicable collocation points and falls behind a purely data-driven approach.

1 Introduction

Deep learning has become the predominant method for tackling many high-dimensional problems, including the topic of autonomous driving, where deep learning-based models have enabled researchers and increasingly industry to push the feasible tasks from the simulation of general traffic patterns to multisecond-long predictions of highly complex traffic scenarios and the paths taken by the vehicles involved [1]. However, there are still many caveats to these systems, especially concerning the use of these models in real life; For example, how to verify that these models behave correctly in all possible scenarios remains an active area of research[10]. This is made especially clear when it can be observed that model performance differs significantly when tested on different datasets, raising the question of how well they would perform on unseen data in a real-life scenario[1].

As a result of these efforts to create more robust and explainable models, the idea of augmenting data-driven Deep Learning (DL) models with additional information that can provide stronger guidance for the model during and after training, or even more explicitly encode rules of the world that might be too subtle to be reliably picked up by the models during regular training, has steadily gained popularity.

Following these findings, we aim to examine the qualitative differences that are achieved by augmenting the currently popular data-driven DL methods with a symbolic model, which, in contrast to data-driven models, are analytically derived by experts. To this end, this paper pursues

several complementary goals: First, design and implement the IDM in a manner that integrates well with the used Argoverse dataset. Second, the requirements of the IDM for the given data must be considered to increase the likelihood that the IDM's predictions are at worst neutral training information for the reference LSTM model, which is being augmented. And third combine the IDM and the LSTM model into a PINN model, where the LSTM component is guided in part by the IDM's predictions during training. To ensure a successful implementation of the IDM, careful attention was paid to the data and its format; thus, we will explain the considerations we had to make.

In this paper, we introduce the IDM, LSTM, and PINN models and apply the latter two to the task of traffic scene analysis. We assess and compare their performance, concluding with a discussion of potential avenues for future research improvements.

2 Related Work

The idea of including classical symbolic models in the training of DL models as a form of knowledge integration was first presented in [3] and showed that it can improve the generalization of the data-driven model in fields where the underlying dynamics are already understood. This was used to help in few data regiments.

Calibration of the model parameters can be observed in [7] to align the symbolic model parameters more closely with the underlying data and help improve the model by capturing parameters such as common acceleration and deceleration, and driving distances between vehicles directly from

the data rather than a priori.

The PINN network as discussed in this work diverges somewhat from the stricter usage of the term that can be found in surveys such as [9] in which PINN refers to the special physics-informed networks used to solve equation systems such as PDEs, fractional equations and integral differential equations in a mesh-free way. Instead, the PINN acronym is used here to refer to a network that is augmented with additional knowledge of the general physical behavior of vehicles, these general physical behaviors themselves being encoded in the employed IDM. In this manner, the presented PINN strictly speaking occupies a space between the algebraic representation and the simulation representation mentioned in [8], as the knowledge is presented as a full simulation of the scenario according to the symbolic model which is then used as a collocation point for the DL model.

3 Method

3.1 Data

The *Argoverse*[4] dataset employed in this work is an open dataset designed for autonomous driving simulation. The dataset includes recordings of road scenarios in *Miami* (spanning 204 km) and *Pittsburgh* (spanning 86 km), totaling more than 324,000 high-resolution traffic scenarios. These scenarios encompass a variety of urban environments such as dense city streets, highways, and intersections. Each scenario provides a 2D, birds-eye view of a recorded driving scenario, with a duration of 5 seconds and a sampling rate of 10Hz. In order to accommodate the IDM's limited capabilities, the dataset was filtered based on the amount of turning present in each scenario.

This filtering process involves fitting a second-degree polynomial to the ground truth trajectories from which the curvature is determined and graded. This grading helps determine whether a scenario involves a straight drive, a left turn, or a right turn. While this method is not flawless, it was employed to establish a baseline for the scenarios used, in order to minimize the inclusion of data that the IDM cannot reasonably model or approximate due to its design.

3.2 Models

3.2.1 Intelligent Driving Model

The IDM is a mathematically devised model to emulate the driving behaviors of cars by computing accelerations based on predefined parameters and the behavior of traffic driving in front of the respective car. Although the IDM was originally introduced to forecast the acceleration of all road users, within the context of our study, we exclusively employ it to predict the acceleration of a single vehicle that is being tracked, namely, the ego vehicle. The IDM was first introduced in [6] and was further elaborated in [5].

The IDM is a microscopic Car Following Model (CFM) for single-lane traffic simulation, first used to match and predict data recorded on German highways. It is single-lane because it does not model lane-changing behavior, and it is

microscopic because it models the movements of discrete vehicles and their accelerations as opposed to macroscopic approaches, which model traffic in terms of abstractions like flow or density. The IDM can be divided into three equations, two to define the net distance s_α and the approaching rate $\Delta v_\alpha(t)$, while the third formula for v'_α denotes the final calculated acceleration.

$$v'_\alpha = a \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta - \left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2 \right) \quad (1)$$

with s_α and $\Delta v_\alpha(t)$ as :

$$s_\alpha = x_{\alpha-1} - x_\alpha - l_\alpha \quad (2)$$

$$\Delta v_\alpha(t) := v_\alpha(t) - v_{\alpha-1}(t) \quad (3)$$

The velocity formula in (1) computes the acceleration by scaling the desired maximum acceleration a with the free road term $\left(\frac{v_\alpha}{v_0} \right)^\delta$ that is controlled by the current and maximum velocity, as well as an acceleration exponent and the congestion traffic term $\left(\frac{s^*(v_\alpha, \Delta v_\alpha)}{s_\alpha} \right)^2$ which takes into account the desired and actual distance from the leading vehicle. One can see that in the case of no or distant traffic in front of the ego vehicle (when s_α is large), the traffic term will vanish, leaving only the free-road term to inform the acceleration. In this case, the remaining function $a \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta \right)$ is a decreasing function for the case where the current velocity v_0 approaches the desired maximum velocity v_α . The added acceleration exponent δ controls the acceleration fall-off rate, where according to [5] "the limiting case $\delta \rightarrow \infty$ corresponds to approaching v_0 with a constant acceleration a , while $\delta = 1$ corresponds to an exponential relaxation to the desired velocity"; it is commonly set to 4.

$$s^*(v, \Delta v) = s_0 + s_1 \sqrt{\frac{v}{v_0}} + T v + \frac{v \Delta v}{2\sqrt{ab}} \quad (4)$$

In the case of nearby traffic in front (s_α lower than s^*), the latter term dominates the equation, which includes the desired and actual distance from the leading vehicle. This desired distance s^* is calculated with the formula seen in (4). Where the minimum distance at standstill s_0 is added to the desired time headway T scaled by the current velocity to produce a safe driving distance to the leader proportional to the velocity and, last, by the term $\frac{v \Delta v}{2\sqrt{ab}}$ which induces the behavior of reducing the velocity difference while not exceeding the desired acceleration or deceleration boundaries.

Note that the authors of the original IDM formulation in [5] set $s_1 = 0$, which represents an additional distance term applicable to congested traffic. In our study, we follow the authors' decision regarding this parameter.

The prediction process in the IDM can be divided into three key stages. Firstly, the leader search phase focuses on identifying potential leading vehicles ahead of the ego vehicle. Secondly, the iterative prediction phase employs the IDM formulation to calculate the desired distance (s^*) and acceleration (v'_α) over a defined number of time steps.

Lastly, in the prediction collection phase, all ego prediction steps are gathered and transformed into a common relative coordinate layout, aligning with the output format of the DL models used in this study.

The IDM relies on a leader vehicle, that is, a road user positioned in front of it, for its computations. Since the Argoverse dataset does not contain ground-truth information about leader vehicles, we have to use heuristics to determine potential candidate vehicles in the scene.

Due to the unstructured format of the data and the IDM's structure, an incorrectly chosen leader vehicle would lead to erroneous predictions for its acceleration output, we chose a strict heuristic to find leaders, liberally applying the fallback of picking no leader to not break the underlying assumptions of the IDM. Detecting no leader vehicle will simply result in the vehicle accelerating to approach its desired speed, which, although it may not lead to accident-free predictions, is less prone to produce erroneous behaviors as tracking an incorrect leader would.

The search function operates on the idea that a line can serve as an approximated path that the ego vehicle is following, and this line is computed by taking the first and last measurement points of the ego vehicle p_1 and p_2 and using these to solve for a line's intercept and slope value.

It is then possible to select only the vehicles for which all measurement points have a distance to the line that is equal to or less than the specified maximum distance. A missing leader vehicle is handled by inserting a dummy entry at an infinite distance. This ensures that in the acceleration computation, the free-road term dominates appropriately when no leader can be set.

After the leaders are determined, we iteratively compute the next position of the ego agent at the given time step. A central part of this process is the generation of the desired acceleration, as given by the IDM formula, which is then translated into new coordinates for the ego vehicle taking into account the velocity of the agent up to this point, its direction and the measurement rate of the data.

As the IDM shows inaccurate accelerations in this discrete time-step setting, which can diverge to infinity in close proximity to another vehicle, it is necessary to clamp the values accordingly. The defined maximum desired acceleration and deceleration were therefore chosen as intuitive limits to ensure that the prediction could be completed without numerical error.

Due to the fact that the ego agent is effectively moved, with each prediction step being its latest position, it becomes necessary to also model the movement of any detected leaders beyond their already supplied movement history in order to avoid the scenario where the ego agent encroaches on a leader vehicle which is not moving throughout the prediction horizon. For simplicity's sake, we use a simple linear prediction that assumes that the leader vehicle has a constant velocity and direction of travel based on its supplied movement history.

The predictions are then used to calculate the loss that was implemented using a *Smooth L1 loss* [11] function. The *Smooth L1 loss*, is a commonly used loss function, particularly in regression tasks. It is similar to the MSE loss but was chosen because it transitions to the MAE loss as the er-

ror increases. This characteristic makes it robust to outliers compared to the traditional MSE loss because it does not grow quadratically for large errors but instead falls back to linear growth.

3.2.2 Long Short Term Memory Model

An LSTM network is a popular DL recurrent architecture for models that are intended to operate on sequential data points. This makes them a natural fit to be able to encode the agents in our traffic scene prediction task. In order to derive the final predictions, an additional multi-layer perceptron (MLP) is superimposed on the LSTM encodings, which facilitates the transformation of the LSTM output into the desired format. The LSTM is configured to take an input of size 4 corresponding to the features $x_{start}, y_{start}, x_{end}, y_{end}$, which correspond to the 2D starting and end coordinates for the given timestep and contains 128 features in the hidden state and 3 layers. The MLP receives the 128 features of the LSTM and reduces the feature space to the final output size. The predictions are then evaluated against the ground truth using the *Smooth L1* loss function.

3.2.3 Physics Informed Artificial Neural Network

The PINN is a combination of the two previous networks. The physics-informed approach integrates the complementary IDM model by adding the difference between its output and that of the DL model as an additional term to the overall loss function with which the LSTM model is trained. In this manner, the output of the IDM acts as a regularization term for the LSTM model, which should provide a loss signal that leads the LSTM to emulate the output of the IDM. The PINN loss is formulated as (5) where $loss_{lstm}$ denotes the regular loss calculated for the LSTM component and $loss_{pinn}$ is the loss between the predictions of the IDM component and the LSTM component which is then scaled using $loss_{idm}$, which represents how well the IDM model performed in the given scenario.

$$loss_{total} = loss_{lstm} + (1 - loss_{idm})loss_{pinn} \quad (5)$$

Importantly, this loss acts as a soft constraint placing no hard constraint on the LSTM contained in the PINN model as a significant reduction in any loss term can potentially compensate for any stagnation or increase in the loss term between the IDM's and LSTM's predictions.

3.3 Training

Prior to commencing the actual training, a parameter search was initiated to identify the most suitable set of hyperparameters. The search involved conducting 500 trials using the *Optuna* framework to determine the optimal hyperparameters for the network. The search identified a learning rate of $7.0616e - 05$ and a weight decay of $6.2742e - 06$ as the optimal choice for the LSTM and PINN model.

Subsequently, with these determined hyperparameters in place, the networks underwent 25 epochs of training on the filtered training dataset. Evaluation of the network performance was carried out at the conclusion of each epoch,

based on the evaluation split.

The IDM being a symbolic model, naturally, was not optimized during training.

4 Results

The training results for the LSTM and PINN model are depicted in figure 1 and 2. A comparison of all models is summarized in Table 1.

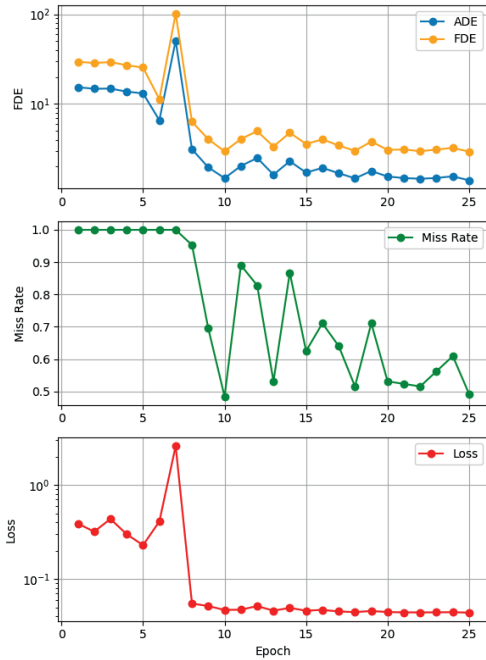


Figure 1 LSTM training progress indicated by the performance on the validation split

Model	Value	Epoch
IDM-ADE	2.79	6
LSTM-ADE	1.965	9
IDM-FDE	7.817	1
LSTM-FDE	6.335	8
IDM-Miss Rate	0.9297	5
LSTM-Miss Rate	0.6953	9

Table 1 Comparison of the LSTM and IDM training progress

5 Discussion

5.0.1 LSTM

Table 1 shows that the trained LSTM model achieves the best performance of all the models examined. Figures 1 and 2 show that the model starts at a comparatively high value for all metrics which begins to improve until the network experiences a short but notable divergence during training. This divergence gives way to a significant improvement of the loss and metric values, showing that after this divergence the values quickly converge to the mentioned best performance of all three examined models.

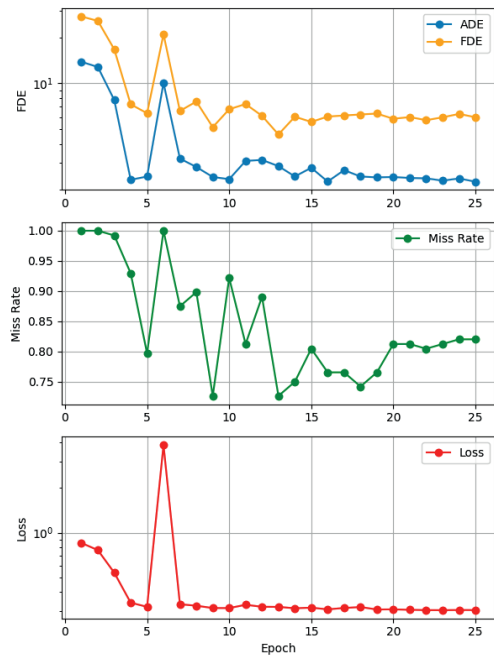


Figure 2 PINN training progress indicated by the performance on the validation split

Said divergence can be seen occurring around the sixth and seventh epoch where the loss jumps to 2.5; the loss, however, drops significantly after this to the 0.05 - 0.04 range where it largely remains for the rest of the training.

The metrics follow this pattern, starting off with slow convergence, spiking between in the seventh epoch, and then radically dropping afterward while then continuing a slow convergence until the end of the training. The result of this training is an LSTM that appears to quickly find a good fit for the data presented to it, achieving the best end scores of all models presented here, as well as good objective scores. We examined several scenarios individually for all models. Examining the LSTM's predictions on specific scenarios shows that, unlike the IDM and PINN models, the LSTM predicts a noticeably longer trajectory, often in line with the ground truth while also maintaining a smoothed trajectory. However, some of the reviewed scenes reveal that unlike what the significantly better metric values presented in 1 would suggest, the LSTM seems to struggle with slight turns, similar to IDM and PINN. Although this would be expected from the IDM, the fact that this effect appears both for the LSTM and the PINN, visible in figure 3, suggests that the filtering of the dataset was too strong and may be causing overfitting for the LSTM in either the standalone version or as a component in the PINN.

5.0.2 PINN

As the figure for the PINN training 2 shows, the training contains a single notable divergence across all metrics, similar to that found in the LSTM, indicating that this is a phenomenon specific to the LSTM component. The spike here reaches a loss of 3.8, higher than the spike observed in the standalone LSTM, and then unlike the pure LSTM model, it reduces down to levels in the neighborhood of

0.31 that are reached just before the spike. This is in contrast to the LSTM which achieves significantly better results after its loss spike, not returning to the pre-spike levels. The loss then continues with a small but persistent convergence to the loss value of 0.3034 at the end of training.

The final performance scores of the model show that it reaches a middle ground between the standalone LSTM and IDM implementations, indicating that even with the filtering of the data set from the previous chapters and the scaling of the influence of the IDM component by its own confidence, the LSTM component might be limited by the addition of the IDM in terms of reaching its full expressive potential to encode the dynamics seen in the data.

On the positive side, it should be observed that the PINN network, similarly to the IDM model starts out with lower ADE, FDE and Miss-Rate scores and maintains this lead over the pure LSTM network until after the spike. This would seem to indicate that the IDM model provides initial stability to the prediction of the LSTM network by providing a generally reasonable driving behavior but, by the same token, prohibits it from fully capturing the details of the driving scenarios the IDM cannot model and for which its integrated loss will subsequently punish the LSTM component for attempting to model ground truth driving that is not captured by the IDM.

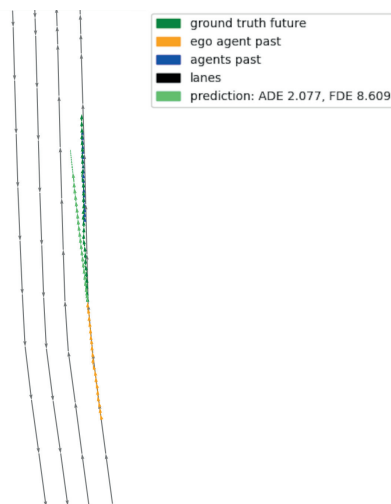


Figure 3 PINN network prediction on a driving scenario with a slight bend

In reviewing the predictions of the PINN, we noticed that it is able to produce a good prediction that shows little deviation from the ground-truth future in scenarios similar to the LSTM as can be seen in figure 3. However, it can be seen that while the prediction mirrors the ground truth in heading and shape, it stops short of the ground truth future in terms of longitudinal distance covered, unlike the LSTM. We noticed that the PINN model generated shorter distances between the prediction steps the longer the simulation went, suggesting that the PINN predicts a gradual deceleration, possibly induced by the IDM component pre-

dicting such breaking to increase the distance to the leader vehicle.

Similarly to the LSTM model, we observed that the PINN model exhibits strong problems with modeling hard turns, some of which still made it into the filtered dataset. This can be attributed to both the IDM component in the PINN network, which cannot predict curvatures, and a general lack of turning scenarios in the filtered dataset, which likely leaves the LSTM component unable to learn these more drastic maneuvers and as a result, cannot augment the IDM model in this scenario.

5.0.3 IDM

The IDM shows a significant performance difference from the trained LSTM model. This is not at all unexpected, as the variety of the underlying data is unlikely to be fully captured by a model that only considers straight driving and does not receive information about the surroundings outside of that of a leader vehicle, should it be present.

On this note, it should be mentioned that the rate at which the IDM identifies leaders for a scenario was measured to gain insight into how well the current techniques for this subtask work. Both the filtered evaluation and the training split were tested, and the IDM identified 2931 leaders in the 10000 scenarios contained in the training split and 24 leaders in the 100 scenarios in the validation split. This averages to a detection rate of 26%, which while leaving significant room for improvement, is a better than expected rate. However, it is important to note that this number does not reflect how many scenarios contain an agent that would be a good candidate for leader detection but was not detected. Instead, the current rate only resembles the distance to the absolute best-case upper bound, not to the actual upper bound of existing leader vehicles captured in the scenarios.

As might be predicted, the IDM can predict the path of the ground-truth-future with good accuracy in scenarios that feature straight or close to straight driving with little lateral movement. Compared to the PINN network, it stands out that the IDM prediction does not reach the longitudinal distance covered by the PINN network, suggesting that the PINN network also shorter-than-ground-truth prediction is influenced by the IDM component.

A detail to note also is the small lateral deviations in the IDM predictions, although not significantly notable performance-wise, these show that the computation of the predictions heading reproduces lateral deviations from the ground truth past of the ego agent, whereas the PINN and LSTM networks learn to smooth out these trajectories.

It can be safely stated that in the current configuration, the IDM is totally unable to model turns in the manner implemented, though there are possible remedies that, once implemented, could enable the model to capture the rather simple turns we observed appearing even in the filtered dataset.

As can be seen in the previous sections, there is room for improving the presented implementation of a PINN network. First and foremost would be the refinement of the utility functions that the IDM uses to determine headings,

velocities, and the leader vehicles and their behavior. In particular, the heading is currently approximated by fitting a line through the first and last data points of the specified agent; naturally, this method is vulnerable to outliers and does not capture even small curvatures due to its linear character.

A straightforward improvement would be to exchange these line approximations with curves of a second degree or higher that are fitted to all agent data points, for example, in a least-squares process. This would enable the model to better withstand outliers and capture simple trajectories with one or two inflection points. This could potentially improve the predictions by allowing the IDM to continue curves that are evident from the ground-truth history and also by enabling the model to identify and acquire leaders that are preceding it in the path but are too far off the current heading to be captured by the current linear algorithm. In a similar manner, the velocity, which is currently also approximated by the start and end point, could be refined by observing the rate of change during the history window and continuing it appropriately.

These same techniques would be equally applicable to the approximate forecasting that is currently done for the leader's future, although it may be feasible to employ the DL model to predict the future of the identified leader vehicles in tandem with the ego vehicle, possibly alleviating the IDM model of continuously approximating the leader vehicle with, presumably, an ever compounding error.

6 Conclusion

This paper presents an implementation of the intelligent driving model, which is used to augment an LSTM-based motion prediction model and stabilize its performance during the initial training stages in the combined PINN network. The three different networks are qualitatively compared through metrics during training, as well as through inspection of selected scenes. The results show that the combined PINN model achieves a performance middle ground between the purely data-driven and symbolic models.

It is also shown that the amount of data filtering that was carried out under the assumption that it would aid the IDM and as a result the PINN is likely too restrictive and prevents the LSTM network from learning crucial turning movements that the IDM cannot model, preventing the models from complimenting each other to the expected extent. With this, the results highlight that knowledge integration must be carefully considered depending on how well the physics-informed model is capable of modeling the data and how much it is weighed as a loss term as a result, as it will otherwise negatively influence the DL model directly or through the constrained data.

7 Literature

- [1] A. Stockem Novo et al., "A review on scene prediction for automated driving", *Physics*, vol. 4, no. 1, 2022, pp. 132-159.
- [2] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps", in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 16-20, 2019, pp. 8748-8757.
- [3] Maziar Raissi and Paris Perdikaris and George Em Karniadakis. *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential*. 2017. arXiv: 1711.10561 [cs.AI] (cit. on pp. 4, 21).
- [4] Ming-Fang Chang and John W Lambert and Patsorn Sangkloy and Jagjeet Singh and Slawomir Bak and Andrew Hartnett and De Wang and Peter Carr and Simon Lucey and Deva Ramanan and James Hays. *Argoverse: 3D Tracking and Forecasting with Rich Maps*. 2019. *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [5] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. "Congested traffic states in empirical observations and microscopic simulations". In: *Physical Review E* 62.2 (Aug. 2000), pp. 1805–1824. doi: 10.1103/physreve.62.1805. url: <https://doi.org/10.1103>
- [6] Martin Treiber and Dirk Helbing. *Explanation of Observed Features of SelfOrganization in Traffic Flow*. 1999. arXiv: cond-mat/9901239 [cond-mat.stat-mech] (cit. on p. 14).
- [7] Zhaobin Mo, Rongye Shi, and Xuan Di. "A physics-informed deep learning paradigm for car-following models". In: *Transportation Research Part C: Emerging Technologies* 130 (Sept. 2021), p. 103240. doi: 10.1016/j.trc.2021.103240. url: <https://doi.org/10.1016%2Fj.trc.2021.103240> (cit. on pp. 25, 32).
- [8] Laura von Rueden et al. "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. doi: 10.1109/tkde.2021.3079836. url: <https://doi.org/10.1109%2Ftkde.2021.3079836> (cit. on pp. 4, 5, 7).
- [9] Salvatore Cuomo et al. *Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next*. 2022. arXiv: 2201.05624 [cs.LG] (cit. on p. 7).
- [10] Julian Wormann et al. *Knowledge Augmented Machine Learning with Applications in Autonomous Driving: A Survey*. 2023. arXiv: 2205.04712 [cs.LG] (cit. on p. 5).
- [11] Pytorch. *Smooth L1 Loss description and implementation*. 2023. url: <https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html#torch.nn.SmoothL1Loss> (visited on 09/26/2023) (cit. on p. 19).

Confidence Tuned Localization through Learning in the Loop

Stefan Schütte, TU Dortmund University, Institute of Control Theory and Systems Engineering, 44227 Dortmund, stefan.schuette@tu-dortmund.de

Dr. Markus Kuhn, ZF Automotive Germany GmbH, 40549 Düsseldorf, markus.kuhn@zf.com

Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram, TU Dortmund University, Institute of Control Theory and Systems Engineering, 44227 Dortmund, torsten.bertram@tu-dortmund.de

Abstract

Learning based methods for localization in robotics and automated vehicles are a topic of ongoing research. While the methods that are currently in use work with a variety of sensor setups and show remarkable performance in pose estimation, research into uncertainty estimation of learning based methods is limited. This work presents a flexible method that allows to take the confidence of the localization method into account during training. Leveraging uncertainty estimates derived from learned features, we reduce overconfidence of the model and improve pose tracking performance purely derived from the training data.

1 Introduction

Solving the localization problem for an automated ego vehicle in an unstructured environment poses a set of challenges to the localization methods that are used. To enable flexible pose estimation using exteroceptive sensors, machine learning based methods have gained popularity in recent years. These methods typically extract features from both map and environment that are useful for the localization task. Afterwards, a registration of these features is performed, resulting in a pose estimate that best matches the current sensor data onto the map. In this configuration, the machine learning model performs the role of a sensor model in traditional localization methods. One problem that arises in this configuration is the often insufficiently accurate measurement covariance of the sensor model. Modern deep neural networks used for classification tasks have a tendency to be overconfident in their estimates [1]. These problems appear in learning based methods used for various automated driving tasks, ranging from odometry [2] to image classification [1]. Localization, as one of the fundamental tasks in the AD (Automated Driving) stack, is no exception in this regard. Models for scan matching may produce overconfident pose estimates [3], [4]. Here, a confident but wrong pose estimate can result in consequences up to crashing [5].

Mitigation methods for this property come from the realm of classification, where *confidence scaling* is used to make the network estimates match a calibrated confidence score. This method is usually applied after training of the network has finished and is applied only to the sensor model [2], [3]. One disadvantage of this approach is the inability to calibrate sensor model and pose tracking jointly. Correlation of consecutive measurements and the optimization target of filter consistency are not explicitly taken into account. For localization, this can be addressed by integrating the filter

into the training task and minimizing not only the tracking error but also filter inconsistency. Sensor model, uncertainty estimation and pose filtering can be trained end-to-end, due to all components being partially differentiable with respect to their inputs.

1.1 Related Work

Combination of deep learning and traditional tracking methods have been an area of active research with the onset of application of deep learning methods in the realm of robotics [6]. For mobile robot applications, state estimation based on a learned feature extractor and a classical tracking method has been used extensively in localization using an existing map [7], [8] and Simultaneous-Localization-and-Mapping (SLAM) [9]. One aspect of state estimation that is crucial for safety critical applications like automated vehicles is the notion of consistency. Consistency, in this context, means an accurate estimate of the uncertainty of the estimate has to be provided by the method. This is required to decide on the availability of the AD system. For the Extended Kalman Filter (EKF), automatic tuning methods exist to ensure accurate estimation of the filter's uncertainty [10]. This work introduces the filter method into the training process to guide the training signal with respect to the overall goal of an accurate and consistent tracking result.

2 Method

In this contribution, a method for the explicit inclusion of measurement and model uncertainty into the training process of a machine learning based localization method is presented. The localization method is set up to run GNSS (Global Navigation Satellite System) free after coarse initialization and only relies on a ranging sensor like radar or

lidar and an IMU (Inertial Measurement Unit). To allow fusion of pose estimation and odometry, an EKF with a constant-turn-rate-and-acceleration model (CTRA) is employed. Sensor model, motion model and filter are all end-to-end differentiable and are implemented in PyTorch.

The model is unrolled for a number of steps on the training dataset and the pose estimate $\hat{\mathbf{x}}$ and the estimated covariance $\hat{\mathbf{P}}$ are included in the loss function. This strategy enables the model to take previous measurements into account when estimating the uncertainty. Overconfidence in areas of high uncertainty can potentially be reduced, as prior pose estimates and the motion model act as a separate source of information. Additionally, the noise parameters of the EKF can be tuned during training by decomposing their covariance matrices via the LDL^T decomposition (square-root-free Cholesky Decomposition). The resulting matrices can be directly implemented as trainable PyTorch parameters, as all constraints on the covariance matrices are implicitly fulfilled by the LDL^T composition. To avoid negative influences of the covariance adaptation on the localization accuracy, the training process distinguishes between two paths through the filter: Covariance estimates only influence the auxiliary confidence loss, while backpropagation into the maximum likelihood pose estimator itself is only enabled for the MSE (Mean-Squared-Error) loss term.

The Mahalanobis distance of the ground truth to the filter estimate, also called *Normalized Estimation Error Squared* (NEES) [10] is used as a proxy for filter confidence. The average of this measure approaches the number of degrees of freedom of the system in a consistent KF. The full model configuration for training is shown in Figure 1.

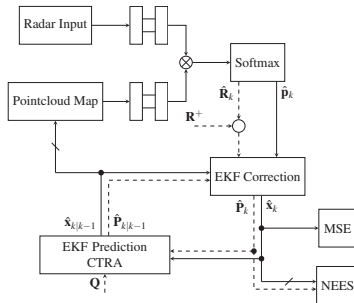


Figure 1 Training configuration of end-to-end learnable localization method. Dashed lines denote path of covariance matrices. Strike through denotes parts of the system where no backpropagation is taking place.

2.1 Scan Matching Model

The learning-based scan matching model performs a correlation based matching of the features extracted by a set of two neural networks that process the sensor data and map respectively. As the architecture is based on [3], direct processing of 3D data is possible.

Starting from the networks described in [3], we leave the PVConv (Point-Voxel-Convolution) architecture [11] for point cloud processing unchanged. For radar processing, a

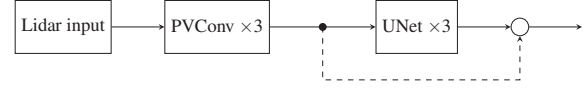


Figure 2 Structure of the embedding networks

PVConv $\times 3$: Only part of map embedding network. Dashed: only part of sensor embedding network. U-Net $\times 3$ denotes a U-Net [13] structure with three downsampling blocks and three upsampling blocks.

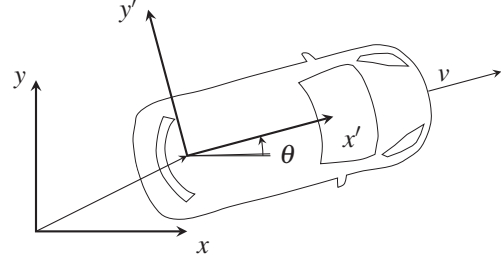


Figure 3 Components of the state vector describing the vehicle state in the world frame. The point on the vehicle that is tracked by the state vector corresponds to the center of the rear axle. x' and y' denote the local coordinate system of the vehicle. $\dot{\theta}$ and a are the time derivatives of θ and v , respectively.

smaller residual formulation is selected to reduce the memory footprint of the network and simplify the learned problem [12]. Figure 2 shows an overview of the embedding network structure.

This allows backpropagation through time for longer sequences and larger batch sizes. These are necessary to increase the sample size for the optimization of the NEES during training.

2.2 Pose Tracking

The pose of the vehicle is tracked by an EKF with the state vector $\hat{\mathbf{x}} = (x \ y \ v \ \theta \ \dot{\theta} \ a)^T$. Figure 3 shows a visual representation of the components of the state vector. The acceleration and angular velocity innovations are taken directly from the IMU of the vehicle in [14]. Velocity is estimated from the pose update via finite differences. The velocity (co)variance is provided based on the position innovation covariance and the covariate factors are set accordingly. x , y and θ make up the current pose estimate $\hat{\mathbf{p}}$ and are estimated by the network based on the current estimate via $\hat{\mathbf{p}} = \hat{\mathbf{p}}_{k|k-1} \boxplus \hat{\mathbf{p}}_d$. Here, $\hat{\mathbf{p}}_{k|k-1}$ is the current prediction based on unrolling the previous estimate and $\hat{\mathbf{p}}_d$ is the difference predicted by the network. \boxplus is the pose composition operation in 2D coordinates.

2.3 Training

The model is trained on a section of the Boreas dataset [14]. Hyperparameters are selected based on a Bayes optimization, where learning parameters and model parameters

Table 1 Training parameters

Parameter	Value
Learning Rate	$1.2 \cdot 10^{-4}$
Weight Decay	$1.2 \cdot 10^{-6}$
Batch Size	3
Sequence length	5
PVConv Voxelization Size	5 · 5
Scan Resolution	0.2 mpx^{-1}
Embedding Resolution	0.2 mpx^{-1}
Number of Rotations	17

like rotation count are jointly optimized. Optimization of hyperparameters is performed on a subset of training and validation datasets to accelerate training. Table 1 gives the results of this optimization. After selection of the best hyperparameters on this subset, a new training is performed from scratch on the full dataset.

2.3.1 Optimizing the Pose Estimate

During training, the localization model is fed consecutive radar scans and predicts the current pose based on the radar features and map. At each time step, the mean-squared-error $\mathcal{L}_{\text{MSE}}(k) = (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)^T (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)$ is computed. Adam [15] with weight decay is used for optimization. The embeddings generated by the network are additionally forced to be rotation equivariant by feeding rotated copies of the map into the network and minimizing the difference to the rotated original embedding. Unrolling the CTRA model, the estimate for the next timestamp is then predicted by the EKF, before another correction step is performed. Due to memory limitations, the model can be unrolled for 3 time steps while maintaining a batch size of 4.

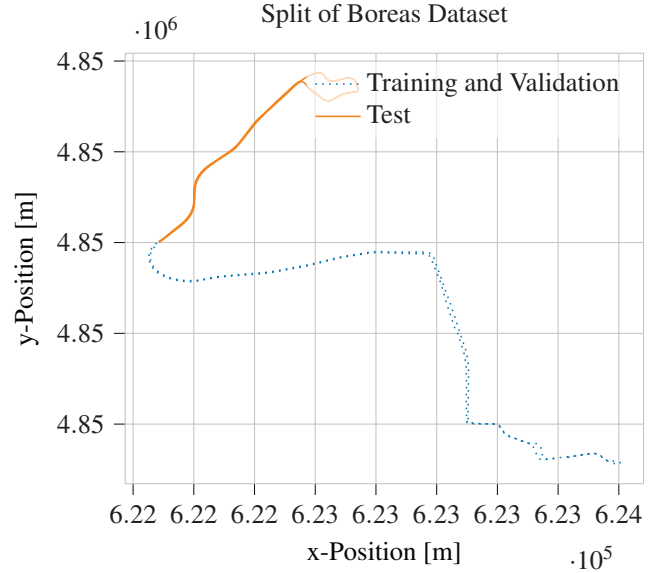
2.3.2 Enforcing Consistency

A separate loss function enforces consistency of the filter output. For this, the Mahalanobis distance $d_m = (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)^T \mathbf{P}^{-1} (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)$ is used. \mathbf{p}_k^* denotes the ground truth pose derived from the real-time-kinematic GNSS present in the vehicle [14]. The NEES loss is selected to be

$$\mathcal{L}_{\text{NEES}} = \left| \frac{\sum_{k=0}^K (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)^T \mathbf{P}^{-1} (\mathbf{p}_k^* - \hat{\mathbf{p}}_k)}{K} - 3 \right|. \quad (1)$$

where K is the number of time steps for which the filter is unrolled during training.

As the correlation output may not directly model the pose uncertainty, an additional covariance matrix \mathbf{R}^+ is added to the estimate from correlation. This covariance matrix is forced to be positive semi definite by the same mechanism that is used for the EKF implementation. The final measurement uncertainty $\hat{\mathbf{R}}$ is thereby composed of $\hat{\mathbf{R}} = \tilde{\mathbf{R}} + \mathbf{R}^+$, where $\tilde{\mathbf{R}}$ denotes the covariance derived from the correlation result as in [3]. Without this mechanism, scenes with high uncertainty in only one dimension might force the feature extractor networks to diffuse sharp features in a way that impacts overall localization perfor-

**Figure 4** Training and test data split of the Boreas dataset.

mance. To further decouple consistency and the pose estimate itself, the estimate $\hat{\mathbf{p}}_k$ is *detached* from the computation graph in the consistency loss to prevent PyTorch from backpropagating any error signal from the consistency check directly into the pose estimator. This is done to keep the training algorithm from pushing the estimate further from the ground truth to achieve consistency instead of lowering the covariance estimate of an underconfident filter.

Both losses are added to the overall loss function $\mathcal{L} = \sum_{k=0}^K \mathcal{L}_{\text{MSE}}(k) + \mathcal{L}_{\text{NEES}}$

3 Evaluation

The method is evaluated on the Boreas dataset [16] using a cross modality localization approach that matches the 2D radar measurements to a precomputed 3D lidar map. As the sensor model is based on [3], it can directly process unstructured 3D data, allowing the model to select 3D features in the lidar point cloud that best match radar targets. This approach avoids manual selection of geometrical structures that are detectable for both sensors.

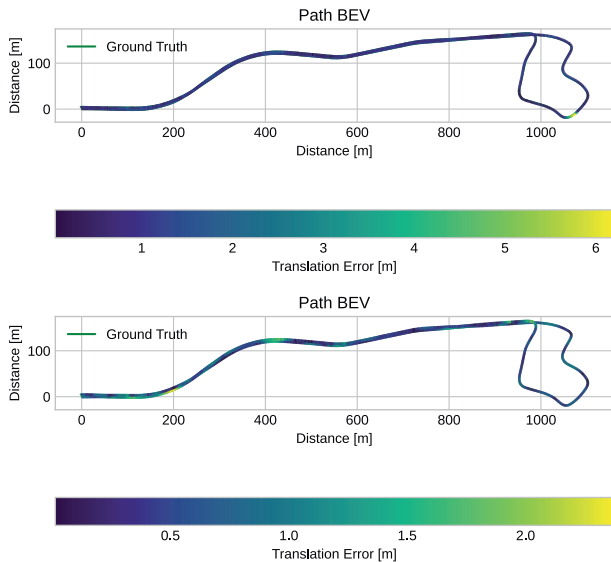
From the dataset, a mapping drive is selected, part of which also serves as the training data. The lidar point clouds from this drive are accumulated and dynamic points are removed [17].

Evaluation is performed by tracking the ego vehicle pose over the previously unseen parts of the map in a previously unseen drive. Figure 4 shows the applied split of the dataset. The test set for this evaluation was selected to be a continuously driven loop to avoid reinitialization during testing, whereas the training set is necessarily split. This does not influence the training procedure, as only short five step sequences are used for training. The of the pose tracking experiment, overlaid with the translation error, is shown in Figure 5. Special focus is placed on the evalua-

Table 2 Results for different calibration methods, BL: Baseline Loss, CL: Consistency Loss

Model	MAE			
	Transl. [m]	Lat. [m]	Lon. [m]	Hdg. [°]
BL	0.93	0.56	0.61	0.4
CL	0.62	0.4	0.39	0.36

tion of filter consistency.

**Figure 5** Results of the pose tracking experiment for the test set with BL and CL. The figures show the Bird's-Eye-View (BEV) of the estimated trajectory.

3.1 Metric Accuracy

To assess the accuracy of the discussed localization approach, we perform pose tracking on a previously unseen part of the dataset. This allows showing generalization capabilities of both the network and the combination of network and tuned filter parameters to new scenes. Table 2 shows the results of the pose tracking experiment. The errors shown are mean absolute translation error (MAE Transl.), mean absolute lateral error (MAE Lat.), mean absolute longitudinal error (MAE Lon.) and mean absolute heading error (MAE Hdg.).

3.2 Filter Consistency

To provide a good estimate of the accuracy of the localization result, the pose covariance computed by the EKF can be taken into account. As the model is trained to provide χ^2 -consistent covariance estimates, the filter should automatically come to a calibrated confidence. This can be shown by evaluation of the NEES on the test set. While the model trained with CL achieves accurate confidence estimates on the training set, the same cannot be said for the test set, as shown in Table 3. This indicates problems with overfitting to the training data that is insufficiently captured

Table 3 NEES Results of the pose tracking experiments

Model	Mean NEES
BL	91
CL	67

by the validation set, as no similar effect can be observed on the validation set. Nevertheless, the model trained with CL shows greatly decreased NEES.

3.3 Discussion

As shown by the results in the previous sections, the presented method is capable of improving the pose tracking performance of a learning based scan matching model. The model trained with NEES optimization generally produces lower confidence pose estimates, but achieves better tracking performance by relying more on inertial odometry. This leads to higher overall tracking accuracy and lowered NEES on the test set. Nevertheless, generalization capabilities of the approach leave room for improvement. Whereas the uncertainty estimation on the training set is well calibrated, on the test set the model remains overconfident. This can potentially be improved by a more diverse training set, as the model may overfit to the training data. Long training times with slow improvement on the validation set further corroborate a correlation between validation and training data that may be less pronounced on a more spatially diverse dataset.

4 Conclusion

In the work at hand, we present an approach to combine learned feature extraction and matching using a neural network with an established tracking approach from traditional state estimation. The coupling of neural network and EKF during training eliminates the need for post-training filter tuning to achieve consistency. Filter parameters tuned alongside network parameters are optimized on the training dataset to achieve consistency. This can be shown by evaluation of the NEES measure. The model achieves accurate pose tracking performance using only radar and IMU measurements. Estimation of the uncertainty of the jointly trained method, while greatly improved during training, is still overconfident in the pose estimate on the test set.

4.1 Outlook

The model, while tuned for a generally suitable confidence measure, does not explicitly consider scene geometry as well as potential viewpoint differences between map and sensor data. This cannot be easily addressed by the current model, as the embedding networks do not model these interactions of map and sensor well, i.e. the map encoder is unaware of the current sensor view and the sensor encoder cannot consider unseen similar geometries. These issues could, however be addressed by adding a global context to the feature maps through mechanisms like attention. One major open question of the presented approach is the gener-

alization capability on a more diverse dataset that contains rural data. Due to the lower feature density of open, rural spaces, confidence estimates of the proposed method may vary drastically. This will be addressed in future work.

References

- [1] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On Calibration of Modern Neural Networks”, in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Aug. 6, 2017, pp. 1321–1330. [Online]. Available: <https://proceedings.mlr.press/v70/guo17a.html>.
- [2] D. Barnes, R. Weston, and I. Posner, “Masking by Moving: Learning Distraction-Free Radar Odometry from Pose Information”, Sep. 9, 2019. arXiv: 1909.03752.
- [3] N. Stannartz, S. Schütte, M. Kuhn, and T. Bertram, “Toward Precise Ambiguity-Aware Cross-Modality Global Self-Localization”, *IEEE Access*, vol. 11, pp. 60 005–60 027, 2023. DOI: 10.1109/ACCESS.2023.3286310.
- [4] F. Fervers, S. Bullinger, C. Bodensteiner, M. Arens, and R. Stiefelhagen, “Uncertainty-aware vision-based metric cross-view geolocalization”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 21 621–21 631.
- [5] T. G. R. Reid, S. E. Houts, R. Cammarata, *et al.*, “Localization Requirements for Autonomous Vehicles”, *SAE Intl. J CAV 2(3):2019*, Jun. 3, 2019. DOI: 10.4271/12-02-03-0012. arXiv: 1906.01061.
- [6] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, “Backprop KF: Learning Discriminative Deterministic State Estimators”, in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: <https://proceedings.neurips.cc/paper/2016/file/697e382cfd25b07a3e62275d3ee132b3-Paper.pdf>.
- [7] P. Karkus, D. Hsu, and W. S. Lee, “Particle Filter Networks with Application to Visual Localization”, May 23, 2018. arXiv: 1805.08975.
- [8] X. Chen, T. Labe, L. Nardi, J. Behley, and C. Stachniss, “Learning an Overlap-based Observation Model for 3D LiDAR Localization”, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4602–4608. DOI: 10.1109/IROS45743.2020.9340769.
- [9] X. Chen, T. Labe, A. Milioto, *et al.*, “OverlapNet: Loop Closing for LiDAR-based SLAM”, in *Robotics: Science and Systems XVI*, Robotics: Science and Systems Foundation, Jul. 2020. DOI: 10.15607/rss.2020.xvi.009.
- [10] Z. Chen, C. Heckman, S. Julier, and N. Ahmed, “Weak in the NEES?: Auto-Tuning Kalman Filters with Bayesian Optimization”, in *2018 21st International Conference on Information Fusion (FUSION)*, IEEE, Jul. 2018. DOI: 10.23919/icip.2018.8454982.
- [11] Z. Liu, H. Tang, Y. Lin, and S. Han, “Point-voxel cnn for efficient 3d deep learning”, *Advances in Neural Information Processing Systems*, vol. 32, 2019. arXiv: 1907.03739.
- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [13] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, in *Lecture Notes in Computer Science*, Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.
- [14] K. Burnett, D. J. Yoon, Y. Wu, *et al.*, “Boreas: A multi-season autonomous driving dataset”, *The International Journal of Robotics Research*, vol. 42, no. 1-2, pp. 33–42, Jan. 2023, ISSN: 0278-3649, 1741-3176. DOI: 10.1177/02783649231160195. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/02783649231160195> (visited on 01/23/2024).
- [15] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, Dec. 22, 2014. arXiv: 1412.6980. [Online]. Available: <http://arxiv.org/pdf/1412.6980v9>.
- [16] K. Burnett, D. J. Yoon, Y. Wu, *et al.*, “Boreas: A Multi-Season Autonomous Driving Dataset”, Mar. 18, 2022. arXiv: 2203.10168.
- [17] G. Kim and A. Kim, “Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images”, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 758–10 765. DOI: 10.1109/IROS45743.2020.9340856.

Application of Basis-Splines for Trajectory Planning in Highway Scenarios

Philip Dorpmüller¹, Dr. rer. nat. Thomas Schmitz², Naveen Bejagam², and Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram¹

¹TU Dortmund, Institute of Control Theory and Systems Engineering, 44227 Dortmund, Germany, philip.dorpmueller@tu-dortmund.de

²ZF Group, Automated Driving and Integral Cognitive Safety, 40547 Düsseldorf, Germany

Abstract

Many approaches plan an automated vehicle's future motion by solving an optimal control problem (OCP). This work discusses the criteria for the OCP design in automated highway driving. The criteria are applied to extend an existing spline-based motion planning algorithm with an additional target manifold and cost features. The resulting planning algorithm is applied to a merge-in maneuver, where the impact of cost weights on the driven trajectory and OCP complexity is evaluated. It is observed that the positive effects of additional features on the closed-loop trajectory are limited by the trajectory parameterization. However, it provides a regularization if the trajectory consists of multiple polynomial segments.

1 Introduction

The control architecture of an automated vehicle is hierarchical [1]. In the architecture, the motion planner usually receives high-level commands from the behavior layer. The motion planner aims to find a dynamically feasible and collision-free trajectory when executing the high-level command while considering the passengers' comfort needs. Formulating an optimal control problem (OCP) allows considering the mentioned demands. Several motion planning approaches apply the OCP of the form in equations (1) and (2), which can be found, e.g., in [2]. In general, the state $\mathbf{x} : \mathbb{R} \mapsto \mathbb{R}^m$ and input trajectory $\mathbf{u} : \mathbb{R} \mapsto \mathbb{R}^n$ on the time horizon $T \in \mathbb{R}$ are subject to optimization.

$$V(\mathbf{x}_0) = \min_{\mathbf{x}, \mathbf{u}, T} F(\mathbf{x}(T)) + \int_0^T l(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \quad (1)$$

$$\text{such that: } \mathbf{x}(T) \in \mathcal{F}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (2)$$

$$\mathbf{x} \in \mathcal{X}_f, \quad \mathbf{u} \in \mathcal{U}_f, \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}).$$

The cost consists of the running cost $l : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}$ and the final cost $F : \mathbb{R}^m \mapsto \mathbb{R}$. Also, a vehicle model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ and the initial vehicle state $\mathbf{x}_0 \in \mathbb{R}^m$ must be considered. The state and input trajectories are constrained to the feasible state space $\mathcal{X}_f \in \mathbb{R}^m$ and input space $\mathcal{U}_f \in \mathbb{R}^n$ considering, among other things, the model's input limitations, free space, and traffic rules. Finally, the state trajectory must end in the terminal set $\mathcal{F} \subset \mathcal{X}_f$. A transcription from an infinite- to a finite-dimensional OCP is required to solve it numerically. Therefore, one can choose from three methods: the Direct Method, the Indirect Method, and Dynamic Programming.

Several criteria need to be considered when developing a motion planning algorithm.

1. Automated driving objectives: the motion planning al-

gorithm must be designed to consider possibly contradicting objectives of different priorities. The objectives include the desire to move towards a destination quickly, a comfortable ride experience for vehicle occupants, and compliance with traffic laws.

2. System stability: the planning algorithm must stabilize the automated vehicle near the target manifold. Design criteria can be found in the optimal control literature [2]. In essence, a sufficiently accurate vehicle and environment model is required, while the cost must resemble a Lyapunov function for the controlled system.

3. Computational complexity: real-time constraints must be considered to achieve an applicable planning algorithm. At the same time, a sufficient solution quality must be achieved.

This paper provides an overview of the criteria for designing an OCP for an automated driving motion planning algorithm. The criteria are applied to extend the OCP of the previous work [3]. Firstly, an additional terminal set is introduced to implement vehicle-following maneuvers. Secondly, the cost function is extended to consider the squared acceleration to improve the perceived comfort. The evaluation demonstrates how the cost adaption influences the driven trajectory and the limitations under the used trajectory parameterization.

The section 2 provides an overview of motion planning for automated driving, particularly the application of polynomial splines. Section 3 introduces the used environment and vehicle model. The cost functional and terminal manifold design are discussed in section 4, followed by a description of the trajectory parameterization in 5. The resulting motion planner is evaluated in 6. Finally, section 7 provides a summary and an outlook.

2 Related Work

The OCP is transcribed into a finite-dimensional optimization problem using the Direct Method. A nonlinear program (NLP) is used frequently since it can consider complex models of the vehicle and environment. [4] formulate a NLP for motion planning in urban environments using polygons to represent obstacles. However, solving the problem can be computationally expensive, depending on the problem's dimensionality and the initial guess. Additionally, the approximation of vehicle dynamics introduces many degrees of freedom.

The differential flatness property of the kinematic vehicle model is used to reduce the OCP's number of variables. A polynomial trajectory parameterization yields the exact solution to the differential equation while being optimal in a squared-jerk manner [5]. Therefore, the number of optimization variables can be reduced while the solution quality remains sufficient, especially in highway scenarios.

However, a single polynomial can be too restrictive in situations close to the kinematic limits, narrow environments, or complex maneuvers. Multiple polynomials can be attached to increase the degrees of freedom, resulting in a polynomial spline function. Splines are used in [6], where an unconstrained NLP is solved to find trajectories in highway scenarios.

In many motion planning algorithms, the feasibility of the trajectory is only ensured at finitely many time steps. Thus, a constraint violation between the time steps cannot be excluded. The problem can be approached by leveraging the basis form of polynomial splines [7]. The coefficients of the basis form bound the spline function from below and above. Therefore, the feasibility of constraints formulated based on such coefficients is guaranteed on the whole planning horizon.

Basis-splines (B-splines) are applied by [8] to control nonlinear systems with guaranteed constraint satisfaction. Furthermore, an approach is proposed to compute the sum and product of splines, allowing the exact consideration of polynomial systems. [9] apply the ideas to plan scaled minimum time trajectories for holonomic mobile robots in dynamic environments.

Inspired by [9], the prior work [3] proposes a shrinking horizon approach into a selected terminal manifold for highway environments. A nonlinear optimization algorithm searches for optimal coefficients and breakpoints (BPs) for a spline trajectory in the Frenét frame.

3 Environment and Vehicle Model

The following summarizes the constraints representing the feasible sets \mathcal{X}_f and \mathcal{U}_f together with the vehicle model. A more detailed explanation regarding the constraints' derivation and implementation is provided in [3].

A linear vehicle model is assumed with the state space $\mathbf{x}_i = [\mathbf{R}p_i \ \mathbf{R}v_i \ \mathbf{R}a_i] : \mathbb{R} \mapsto \mathbb{R}^3$. The index $i \in \{x, y\}$ determines the direction of motion. The ego vehicle trajectory is described in the Frenét coordinate system, relative to the target lane center, which is indicated with the prescript $\mathbf{R}\square$.

$\mathbf{R}p_i$ denotes the position, $\mathbf{R}v_i$ the velocity, and $\mathbf{R}a_i$ the acceleration in the i -direction. Limitations on the vehicle's acceleration ensure the dynamic feasibility of the trajectory. Therefore, the acceleration in the ego vehicle's normal coordinates, indicated with $\mathbf{N}\square$, is approximated with $\mathbf{N}a_x \approx \mathbf{R}a_x$ and $\mathbf{N}a_y \approx \mathbf{R}a_y - \kappa_{\mathbf{R}} \mathbf{R}v_x^2$, using the path curvature $\kappa_{\mathbf{R}} : \mathbb{R} \mapsto \mathbb{R}$. The acceleration in the x -direction is bounded to $-\bar{a}_x \leq \mathbf{R}a_x \leq \bar{a}_x$ with the limit $\bar{a}_x \in \mathbb{R}^+$. The bound in y -direction is simplified to $-\bar{a}_y + \hat{\kappa} \bar{v}_x^2 \leq \mathbf{R}a_y \leq \bar{a}_y - \hat{\kappa} \bar{v}_x^2$ and considers the maximum road curvature $\hat{\kappa} \in \mathbb{R}^+$, the maximum velocity $\bar{v}_x \in \mathbb{R}^+$, and the acceleration limit $\bar{a}_y \in \mathbb{R}^+$. The predicted occupancies of other vehicles are considered to plan a collision-free trajectory. Axis-aligned ellipses are used to approximate the vehicles' rectangular shapes conservatively. It is assumed that for each $o = 1, 2, \dots, N_{ov}$ other vehicle with $N_{ov} \in \mathbb{R}^+$ a predicted position ${}^o\mathbf{R}p_i : \mathbb{R} \mapsto \mathbb{R}$ is available. Considering the ellipse diameters of the ego vehicle $\Delta_i \in \mathbb{R}^+$ and the obstacle vehicles ${}^o\Delta_i$ the constraints

$$\frac{(\mathbf{R}p_x - {}^o\mathbf{R}p_x)^2}{(\Delta_x + {}^o\Delta_x)^2} + \frac{(\mathbf{R}p_y - {}^o\mathbf{R}p_y)^2}{(\Delta_y + {}^o\Delta_y)^2} \geq \frac{1}{4} \quad (3)$$

ensure that vehicle shapes do not overlap. A limitation of the absolute relative heading angle to $\bar{\psi} = 10 \text{ deg}$ in the equations (4) and (5) ensures the validity of the ego vehicle's ellipse with the maximum lateral position $\hat{p}_y \in \mathbb{R}^+$.

$$0 \leq \mathbf{R}v_x (1 - \hat{\kappa} \hat{p}_y) \tan \bar{\psi} - \mathbf{R}v_y \quad (4)$$

$$0 \leq \mathbf{R}v_x (1 - \hat{\kappa} \hat{p}_y) \tan \bar{\psi} + \mathbf{R}v_y \quad (5)$$

An upper bound on the velocity $\mathbf{R}v_x \leq \bar{v}_x$ prevents the vehicle from exceeding the speed limit. Additional bounds $p_{lb} - \Delta_y/2 \geq \mathbf{R}p_y$ and $p_{rb} + \Delta_y/2 \leq \mathbf{R}p_y$ on the lateral position prevent the vehicle from crossing the solid lane boundaries. The variables $p_{lb} \in \mathbb{R}$ and $p_{rb} \in \mathbb{R}$ denote the lateral position of the left and right solid lane boundaries relative to the target lane center.

4 Cost Functional and Terminal Set

Next, the cost functional and the terminal sets are derived and assessed based on the three criteria mentioned in section 1. [10] provide an overview of automated driving objectives and corresponding cost features, which are used in the following. Firstly, the motion planner shall reach a target provided by the behavior layer. The target should be reached as quickly as possible to maximize the overall progress towards the navigation goal. In structured environments, the targets in the longitudinal direction are usually a velocity or based on a lead vehicle trajectory. Furthermore, a reference path in the form of a lane center is provided as the target in the lateral direction. The cost measure progress by distance from the target, like in [4], or by the time into the target, as in [5].

Secondly, the automated vehicle must adhere to traffic rules on public roads, like [11]. This paper focuses on the rules that can be considered by the motion planner and that apply to highway-like environments. Such rules include tracking the rightmost lane, if possible, and overtaking on

the left side. A minimum time headway of 2 s must be ensured toward the leading vehicle in the longitudinal direction. Also, the vehicle shall not exceed the given velocity limit.

Finally, the passengers desire a comfortable driving experience. The literature has neither a unique definition of driving comfort nor a unique measure for the perceived driving comfort [12]. However, [13] describe effects that influence driving comfort and can be considered by a motion planning algorithm. An important factor is the force acting on the vehicle passengers due to steering and acceleration. The effect of forces is usually measured by acceleration and jerk, which should reach low values [12]. The experience of motion sickness is mainly the result of low-frequency accelerations, which may be addressed by a sufficiently continuous trajectory with low jerk [13]. Also, the automated vehicle should convey the feeling of safe operation to the passengers. Apparent safety can be achieved with, e.g., a safe distance to other vehicles and avoiding overshooting during a lane change [13]. Finally, the requirement for natural paths demands trajectories similar to human drivers. A method to approach this task is inverse reinforcement learning. Therein, the cost elements are weighted based on human driver trajectories [10].

4.1 Terminal Set

The terminal set must be chosen to ensure the convergence of the vehicle into a chosen target. Therefore, it must be reachable within the planning horizon of $T = 10$ s.

The center of the target lane is chosen as the terminal set in the lateral direction

$$\mathcal{F}_y = \{\mathbf{x}_y \in \mathbb{R}^3 \mid {}_R p_y \in \mathcal{P}_{y,lc} \wedge {}_R v_y = 0 \wedge {}_R a_y = 0\}, \quad (6)$$

where $\mathcal{P}_{y,lc}$ includes the y -positions of the lane centers. The behavioral layer is assumed to provide the target lane, considering the traffic rules regarding left-side overtaking and right-side driving. The terminal set strictly forces the vehicle to follow the lane center. Highways are constructed with continuity up to curvature in congruence with the maximum allowed velocity to ensure safe driving [14]. Thus, forces are assumed to be acceptable to the passenger. The maximum allowed velocity is chosen as the target in the longitudinal direction. Driving with maximum velocity minimizes the time along a route while it leads to zero acceleration in the x -direction. Also, the longitudinal target is considered the terminal set

$$\mathcal{F}_{x,v} = \{\mathbf{x}_x \in \mathbb{R}^3 \mid {}_R v_x = \bar{v}_x \wedge {}_R a_x = 0\}. \quad (7)$$

However, the speed limit might not be reachable in the presence of other vehicles.

The required minimum time headway of $t_{lv} = 2$ s is considered by offsetting the predicted lead vehicle position to ${}^o_R p_{x,lv} = {}^o_R p_x - t_{lv} {}^o_R v_x$, with the predicted velocity ${}^o_R v_x$, resulting in the terminal set

$$\mathcal{F}_{x,lv} = \left\{ \begin{bmatrix} {}^o_R p_{x,lv} \\ {}^o_R v_{x,lv} \\ {}^o_R a_{x,lv} \end{bmatrix} \right\}. \quad (8)$$

${}^o_R v_{x,lv}$ and ${}^o_R a_{x,lv}$ denote the offset position's first and second-order time derivative. Though other drivers and au-

tomated vehicles might have similar objectives, staying behind the lead vehicle can be uncomfortable. The behavioral layer is assumed to choose a better target in such cases.

Representing the terminal set via the target manifold simplifies the choice of the cost functional [2]. Also, planning into the target manifold leads to low-cost trajectories in closed-loop control. However, one might require a longer planning horizon and more degrees of freedom in the resulting optimization problem compared to a less strict terminal set.

4.2 Cost Functional

The cost functional consists of the running and the final cost. Since the trajectory is planned into the target manifold, the final cost is $F := 0$. The running cost must be positive definite and zero in the target manifold to ensure asymptotic stability [2]. The minimum number of trajectory features shall be included to achieve a low computational complexity while accounting for the automated driving objectives. Linear quadratic minimum-time problems, applied frequently in the automated driving context, account for the progress and comfort objectives. Moreover, they can yield an analytic solution [15].

In line with the design of the terminal set, time into the target manifold is selected as a progress metric. Traffic rules are already considered via the constraints and the terminal set selection. Thus, no features related to traffic rules are included in the cost. Commonly, acting forces are considered by minimizing acceleration and jerk-dependent cost features. Many applications choose the time integral over squared jerk and acceleration, complying with the positive definiteness requirement [10]. According to [16], the jerk also plays a prominent role in the causes of motion sickness. The distance to other vehicles is already considered in the terminal set, while the constraints prevent collisions with conservatively approximated vehicle shapes. Thus, no additional features related to perceived safety are included in the cost in favor of the computational complexity. The weights among the cost features must be appropriately chosen to achieve results similar to human-driven trajectories. The choice of weights is scenario-dependent [10] and does not deny asymptotic stability, except setting all to zero. However, the choice affects the problem complexity and solution quality, which is discussed in section 6.

As a result of the made considerations, the cost functional includes a weighted sum of time, the squared jerk, and the squared acceleration:

$$J := \sum_{i \in \{x,y\}} \int_0^{T_i} w_{T_i} + w_{a_i} {}_R a_i^2(\tau) + w_{j_i} {}_R j_i^2(\tau) d\tau. \quad (9)$$

$T_i \in \mathbb{R}^+$ denotes the time where the state $\mathbf{x}_i(T_i) \in \mathcal{F}_i$ reaches the target manifold with $\mathcal{F}_x \in \{\mathcal{F}_{x,v}, \mathcal{F}_{x,lv}\}$.

5 Trajectory Parameterization

After formulating the OCP, it must be transcribed into a finite-dimensional optimization problem. The choice of transcription method significantly influences solution qual-

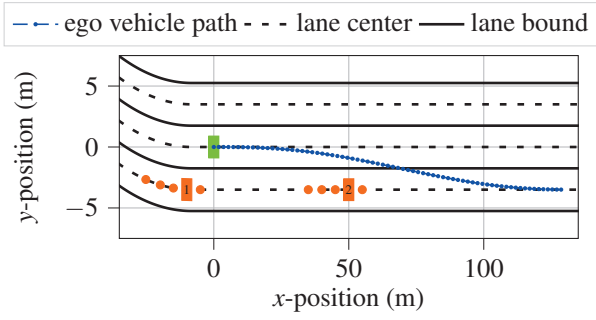


Figure 1 Scene including the ego vehicle (green) and two other vehicles (orange). The orange dots mark alternate start positions used in the runtime evaluation.

ity and problem complexity. As in [3], Direct and Indirect Methods are combined. Firstly, the Indirect Method is applied to the OCP, while neglecting the inequality constraints. If $w_{a_i} = 0$, quintic polynomials are the optimal function class [5]. Next, considering the inequality constraints, the Direct Method is used. Connecting multiple polynomials results in a spline function, which improves the solution quality for $w_{a_i} > 0$ and in active inequality constraint cases. The IPOPT nonlinear optimization algorithm [17] chooses the spline parameters and BPs optimally.

The trajectory feasibility must be ensured along the whole trajectory with finitely many constraints. One possibility is the trajectory parameterization with splines in basis form. The basis form of a polynomial spline $s: \mathbb{R} \mapsto \mathbb{R}$ is defined by its B-splines $b_{\mu, \rho, \mathbf{k}}: \mathbb{R} \mapsto [0, 1]$ and coefficients $c_{\mu} \in \mathbb{R}$ with $\mu = 0, 1, \dots, N_b - 1$ and $N_b \in \mathbb{N}^+$ [7]:

$$s = \sum_{\mu}^{N_b-1} c_{\mu} b_{\mu, \rho, \mathbf{k}} \quad (10)$$

The B-spline of order $\rho \in \mathbb{N}^+$ is defined recursively on the knots $\mathbf{k} = [u_0 \ u_0 \ \dots \ u_1 \ u_1 \ \dots \ u_{N_u-1}]^T$. The knots consist of repeated BPs $u_{\xi} \in \mathbb{R}$ with $\xi = 0, 1, \dots, N_u - 1$ and $N_u \in \mathbb{N}^+$. The spline functions are contained within their coefficients $\min\{c_{\mu}\} \leq s \leq \max\{c_{\mu}\}$. Thus, it is sufficient to check the coefficients of the constrained spline function for feasibility to ensure the trajectory feasibility. However, the approach can reduce the solution space, leading to suboptimal results in active constraint cases.

6 Evaluation

The following evaluation demonstrates the asymptotic convergence of the proposed motion planner. In addition, the influence of the cost weights on the planned and driven trajectory is analyzed. Therefore, a merge-in scenario is simulated. On the right lane, two vehicles drive with the constant deceleration of -0.5 m/s^2 , both starting at the velocity 60 km/h . The ego vehicle starts with 65 km/h and desires following the front vehicle 2 on the right lane at a constant time-gab of 2s. **Figure 1** shows the scene at the initial time step.

The behavior of the other vehicles is known precisely, so no uncertainty is induced due to prediction errors. Also, an

Table 1 Performance of the closed-loop trajectories, evaluated as the sum of distinct cost features. The improvement over the features from \mathbf{W}_j is shown in percent.

weights	$\int_{\mathbb{R}} a_x^2 + \int_{\mathbb{R}} a_y^2$		$\int_{\mathbb{R}} j_x^2 + \int_{\mathbb{R}} j_y^2$		$T_x + T_y$	
	m^2/s^3	%	m^2/s^5	%	s	%
\mathbf{W}_j	5.36	0	1.51	0	17	0
\mathbf{W}_{ja}	5.11	5	2.03	-34	16.8	1
\mathbf{W}_a	5.15	4	4.51	-199	16.7	2

exact realization of the planned motion is assumed. However, a limited localization accuracy along the lane center induces inconsistencies.

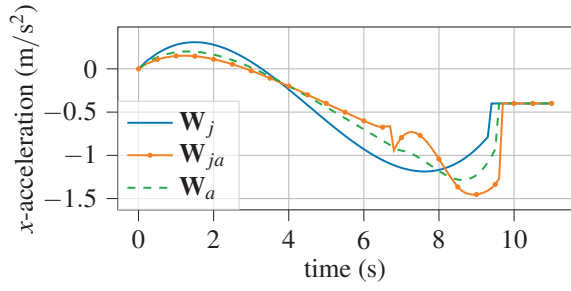
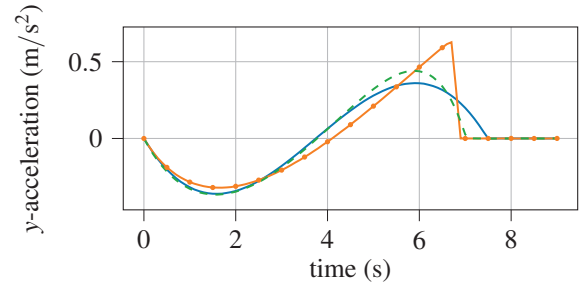
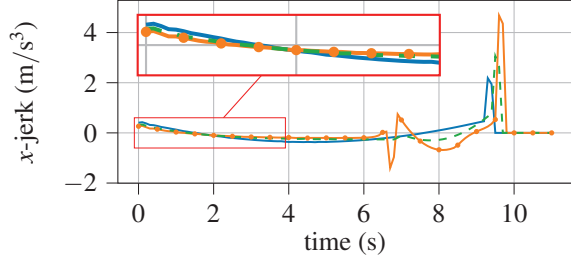
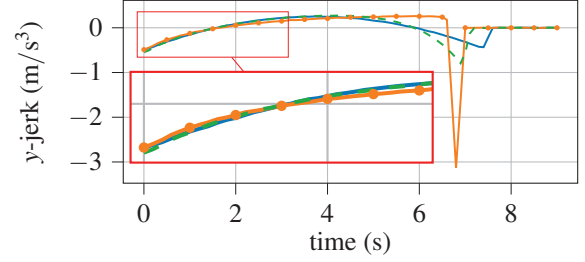
A set of cost weights is described by the tripple $\mathbf{W}_{\square} = (w_T, w_a, w_j)$, with $\square \in \{j, a, ja\}$. The weights in each direction $i \in \{x, y\}$ are chosen the same, i.e., $w_{T_i} = w_T$, $w_{a_i} = w_a$ and $w_{j_i} = w_j$. The first set $\mathbf{W}_j = (0.2, 0.0, 0.8)$ weights time and jerk, the second set $\mathbf{W}_a = (0.2, 0.8, 0.0)$ time and acceleration. In the third set, jerk and acceleration weights are non-zero $\mathbf{W}_{ja} = (0.2, 0.4, 0.4)$. Initially, the trajectory in the x -direction contains one interior BP, while in the y -direction, no interior BP is used, indicated by the tuple $\mathbf{U}_1 = (1, 0)$.

The ego vehicle state converges toward the target manifold for all weight sets. Each weight set results in a different trajectory in **Figure 2**. The trajectory of the set \mathbf{W}_j shows a continuous acceleration, except in the vicinity of the target manifold. In close vicinity to the target, the planning algorithm returns a trajectory that follows the target manifold to prevent high accelerations and jerks due to a noisy state transformation. In contrast, the \mathbf{W}_a trajectory results in higher jerk values. A discontinuity is observable in between 6s and 7s. Most of the time, absolute acceleration and jerk are lower than \mathbf{W}_j . Though close to the target manifold, the values grow larger. The trajectory resulting from \mathbf{W}_{ja} can be described as a combination of the former two trajectories, sharing properties of the former two.

The **Table 1** reports unweighted cost features resulting from the closed-loop trajectories. An increase in the acceleration weight leads to an improvement in the combined acceleration features and convergence times. However, the combined jerk features are increased by up to a factor of 2 in the case of \mathbf{W}_a . Acceleration and time improve only in a one-digit percentage range due to the suboptimal parameterization.

The solution quality is improved by introducing one additional BP in each direction, denoted by the tuple $\mathbf{U}_2 = (2, 1)$. The higher the acceleration weights, the higher the cost reduction. Since one polynomial is already jerk optimal, the results from a weight set \mathbf{W}_j do not benefit from additional BPs in this scenario.

Another effect of the suboptimal parameterization is the inconsistency between planned trajectories. Over consecutive time steps, the trajectories differ from each other. The effect results from the optimization algorithm finding a solution that is at least as good as the previous one. The difference increases as the acceleration weights increase. The effect is observable in **Figure 3**, showing the maximum po-

(a) Acceleration x -direction in the target lane center's Frenét coordinate system.(b) Acceleration y -direction in the target lane center's Frenét coordinate system.(c) Jerk x -direction in the target lane center's Frenét coordinate system.(d) Jerk y -direction in the target lane center's Frenét coordinate system.**Figure 2** Closed-loop acceleration and jerk trajectories resulting from different cost weights.**Table 2** Cost values of the closed-loop trajectories for two sets of BPs. The improvement over U_1 is shown in percent.

BPs	W_j		W_{ja}		W_a	
	cost	%	cost	%	cost	%
U_1	4.605	0	6.829	0	7.461	0
U_2	4.605	0	6.783	0.67	7.278	2.45

sition difference between two trajectories over consecutive time steps. At the beginning of the maneuver, the position inconsistency is maximum for W_a and W_{ja} . The results from W_j stay lower, while the location uncertainty influences the x -position.

The choice of the cost function influences the computational complexity of the optimization problem. The experiment is repeated with the other vehicles placed at five locations along their lane, as indicated in **Figure 1**. The jerk and acceleration cost features add new optimization variables and constraints to the problem. **Table 3** provides the maximum problem dimensionality for the different weights. In addition, statistics of the time until IPOPT converges are provided (Ubuntu 22.04, AMD Ryzen 5 3600 CPU at 3.6GHz, 16 GB RAM). The dimensionality decreases during the convergence to the target. The solution from the previous time step is used to initialize the optimization algorithm. The lowest overall time metrics are observable for the weights W_a with the lowest problem dimensionality. The set W_j results in higher time metrics. The inconsistency between planned trajectories is minimal, but the problem dimensionality increased. In addition, situations occur without a unique solution, slowing the convergence in several time steps. The reason is the redundancy

Table 3 Dimensionality of the optimization problem provided by the number of optimization variables and constraints for different weights. In addition, mean, standard deviation, and maximum computation time are reported.

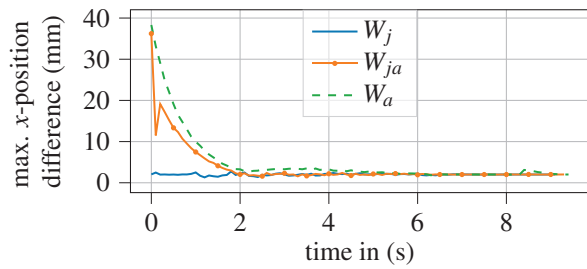
weights	vars.	constr.	time (ms)		
			mean	std.	max.
W_j	332	321	12.74	23.68	236
W_{ja}	364	353	17.59	81.35	1577
W_a	324	313	9.26	9.73	164

of BPs in inactive inequality constraint cases. Finally, the set W_{ja} results in the highest problem dimensionality and time metrics. The increased problem complexity causes a significant increase in iterations in some scenarios. In one situation, IPOPT fails to converge, though being able to return a feasible solution

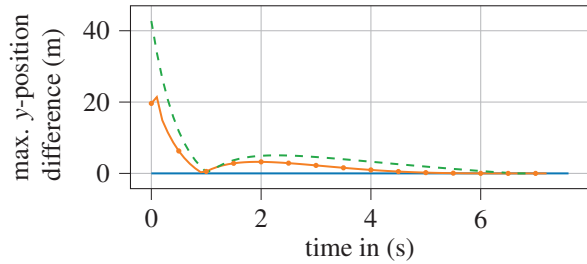
To conclude, adding the squared acceleration to the cost might not affect the behavior of the motion planner in the desired manner, as the number of BPs is small. In addition, it increases the problem's complexity due to an increased dimensionality. Thus, omitting the acceleration feature in low BP configurations might not limit the capabilities of the planning algorithm significantly. However, in the presence of multiple interior BPs, a regularization should be applied to minimum jerk cost, improving the convergence of the optimization algorithm.

7 Summary and Outlook

This contribution proposes criteria for designing an OCP in application to a planning algorithm for automated highway



(a) Maximum x -position difference between two consecutive time steps in the target lane center's Frenét coordinate system.



(b) Maximum y -position difference between two consecutive time steps in the target lane center's Frenét coordinate system.

Figure 3 Inconsistency between planned trajectories of consecutive time steps measured by the maximum position difference.

driving. Based on these criteria, an extension to an existing motion planning problem is proposed. The resulting planning algorithm is applied to a merge-in maneuver, where the effect of different cost weights and BPs on the solution quality and computational complexity is evaluated.

In future work, the experiments should be extended to additional scenarios to assess better the effect of cost features on the solution quality. Furthermore, a regularization should be added to the minimum-jerk cost so a unique solution exists in the presence of multiple interior BPs.

8 Literature

- [1] B. Paden, M. Čáp, 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, Jan. 2016.
- [2] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jan. 2000.
- [3] P. Dorpmüller, T. Schmitz, N. Bejagam, and T. Bertram, "Time-Optimal Trajectory Planning in Highway Scenarios using Basis-Spline Parameterization," *arXiv*, no. arXiv:2310.03359, Jun. 2023.
- [4] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for Bertha — A local, continuous method," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, Jun. 2014, pp. 450–457.
- [5] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 346–359, Jan. 2012.
- [6] C. Götte, M. Keller, T. Nattermann, C. Haß, K.-H. Glander, and T. Bertram, "Spline-Based Motion Planning for Automated Driving," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9114–9119, Jan. 2017.
- [7] C. de Boor, *A Practical Guide to Splines*, rev. ed., 1. hardcover print ed., ser. Applied Mathematical Sciences. New York, NY: Springer, Jan. 2001, vol. 27.
- [8] W. van Loock, G. Pipeleers, and J. Swevers, "Optimal motion planning for differentially flat systems with guaranteed constraint satisfaction," in *2015 American Control Conference (ACC)*, Jul. 2015, pp. 4245–4250.
- [9] T. Mercy, W. van Loock, and G. Pipeleers, "Real-time motion planning in the presence of moving obstacles," in *2016 European Control Conference (ECC)*, Jan. 2016, pp. 1586–1591.
- [10] M. Naumann, L. Sun, W. Zhan, and M. Tomizuka, "Analyzing the Suitability of Cost Functions for Explaining and Imitating Human Driving Behavior based on Inverse Reinforcement Learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 5481–5487.
- [11] "Verordnung zur Neufassung der Straßenverkehrs-Ordnung (StVO)," *Bundesgesetzblatt Teil I*, no. 12, p. 367, Mar. 2013.
- [12] H. Bellem, "Comfort in Automated Driving: Analysis of Driving Style Preference in Automated Driving," Ph.D. dissertation, Chemnitz, Jan. 2018.
- [13] M. Elbanhawi, M. Simic, and R. Jazar, "In the Passenger Seat: Investigating Ride Comfort Measures in Autonomous Cars," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 4–17, Jan. 2015.
- [14] M. Rohloff and Forschungsgesellschaft für Straßen- und Verkehrswesen, *Richtlinien für die Anlage von Autobahnen: RAA*, ausgabe 2008 ed., ser. FGSV. Köln: FGSV-Verlag, 2008, no. 202.
- [15] E. Verriest and F. Lewis, "On the linear quadratic minimum-time problem," *IEEE Transactions on Automatic Control*, vol. 36, no. 7, pp. 859–863, Jul. 1991.
- [16] A. Steinke and U. Konigorski, "Trajectory Planning considering Motion Sickness and Head Movements," *IFAC-PapersOnLine*, vol. 55, no. 14, pp. 113–119, 2022.
- [17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Jan. 2006.

On the Design of Interaction-Aware SCMPC for Highway Merging Scenarios

Robin Kensbock^{1,2}, Georg Schildbach¹

Abstract—This paper addresses interaction-aware decision making and motion planning for *highway merging situations* using *Scenario-based Model Predictive Control (SCMPC)*. Given tactical decision options for the autonomous vehicle (AV), a traffic prediction algorithm intends to identify the most likely evolutions from the current traffic scene, which are then evaluated by an ensemble of SCMPCs to determine the most efficient decision regarding velocity tracking cost and safety margin satisfaction. This way, we aim to leverage interaction-aware predictions to gain insights about possible target vehicle reactions to the decisions of the AV with the incentive to solve merging situations more efficiently and enhance safety by considering target vehicle intentions. We demonstrate the approach in comparison to a non-interaction-aware baseline method in a multi-vehicle simulation study.

Keywords—interaction-aware autonomous driving, highway merging, motion planning, model predictive control

I. INTRODUCTION

AUTONOMOUS driving demands the fulfillment of various algorithmic requirements for decision making and motion planning, including ensuring that the path is dynamically feasible and avoiding obstacles. However, navigating dynamic environments with multiple human actors also necessitates some understanding of traffic participants' interactions. Otherwise, misunderstandings may occur that can potentially lead to safety threats [1] or inefficient behavior of the autonomous vehicle, such as freezing [2]. According to [3], interaction in traffic situations occurs if spatio-temporal conflicts emerge among agents. One traffic situation that naturally meets this definition is highway on-ramp merging. A decreasing number of available lanes induces a demand for a change to the remaining lanes. On the other hand, vehicles have an incentive to maintain their velocity. This provokes a conflict between traffic participants, where drivers have to anticipate the intentions and cooperativeness of surrounding vehicles in order to efficiently negotiate their maneuvers. Human decisions, however, are hard to predict as they are to follow one of multiple likely driving modes, i.e., high-level maneuvers, within one traffic scene, referred to as multi-modality.

Scenario-based Model Predictive Control (SCMPC) [4], a variant of Model Predictive Control (MPC), can be a possible solution to capture this behavior. Under consideration of multiple predictions of the current situation within the constraints, SCMPC approximates the uncertainty while optimizing inputs and trajectory, ensuring feasibility for all scenarios. This makes SCMPC able to account for the multi-modality of surrounding human drivers, making it well suited for safety-critical applications. Scenarios can be generated separately using arbitrary prediction approaches.

In this work, we further advance our architecture for interaction-aware decision making and motion planning based on SCMPC [5]. The approach incorporates a model-based

interaction-aware traffic prediction method by Bahram et al. [6] to predict the behavior of surrounding vehicles. The core concept assumes that drivers are more likely to perform maneuvers that minimize the risk of the traffic scene, which is approximated by the time-to-collision between vehicles. By applying the prediction approach, we are able to identify and assess potential traffic scenarios conditioned on the tactical decision options (TDOs) of the autonomous vehicle. Formulating the associated most likely predicted scenarios within the constraints of the SCMPC, the architecture aims to make informed decisions on lane change maneuvers and acceleration, identifying the optimal TDO in terms of a given cost function. We illustrate the functionality of our approach in comparison to a non-interaction-aware baseline planner in two case studies.

A. Related Work

Although the field of interaction-aware autonomous driving is still in a nascent stage, MPC has already been applied in this context for highway merging. Game-theoretical [22]–[27] and learning-based [11], [12] approaches try to take the influence of the autonomous vehicle (AV) into account. In [30], [31], the authors combine MPC with Reinforcement Learning for on-ramp merging. In [28], [29], a differential game is solved in order to negotiate a merging maneuver. However, those approaches do not capture multi-modality.

Variants of MPC that can be able to account for this uncertainty, like Stochastic MPC [7]–[11], Scenario-based MPC [5], [13]–[18], as well as Branch MPC [19]–[21] have also been studied for interaction-aware autonomous driving. However, only a fraction covers highway merging situations, or just estimates and reacts to the intentions of surrounding vehicles in the planning process but does not consider the impact of the EVs decisions. Generally, existing papers rarely consider more than one surrounding target vehicle (TV) for highway merging situations, limiting awareness of the overall scene or do at least not account for multi-modality simultaneously.

¹Institute for Electrical Engineering in Medicine of the University of Luebeck, Germany, ({r.kensbock, georg.schildbach}@uni-luebeck.de)

²Founded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 460891204

B. Contribution

In this paper, we extend our prior work in [5] to be capable of handling highway merging situations. To this end, (i) the approach is augmented to capture the *road geometry of a highway on-ramp merging environment* with two parallel lanes. Further, the predictions are not limited to lateral maneuvers anymore but also (ii) *consider discrete decisions on acceleration and braking*, enabling interaction-aware motion predictions in the longitudinal direction. In order to demonstrate its effectiveness, (iii) we compare the interaction-aware motion planner with a *non-interaction-aware baseline planner* in a simulation study. In contrast to existing papers, the approach considers multiple surrounding target vehicles and aims to account for multi-modality at the same time.

II. BACKGROUND

A. Terminology

In this paper, we stick to the terminology used in our prior work [5]. The index $t \in \{0, \dots, T\}$ refers to the time steps over the TP horizon T with time interval Δt . All considered vehicles are denoted by i , including the AV ($i = 0$) and $i \in \{1, \dots, I - 1\}$ with I referring to the number of vehicles.

Definition 1 (Maneuver): We assume a fixed number of potential maneuvers for the AV and TVs. Generally, a maneuver can be described as a pair consisting of a longitudinal and lateral action. Each lateral maneuver $m_{t,\text{lat}}^{(i)}$ of vehicle i represents the decision to drive in the center on one of two target lanes considered from the set $\in \{0, 1\}$. A longitudinal maneuver $m_{t,\text{lon}}^{(i)} \in \{-a, 0, a\}$ corresponds to maintaining a constant amount a of acceleration/deceleration or keeping the velocity unchanged. Both longitudinal and lateral maneuvers can be combined into an overall maneuver $m_t^{(i)} := \{m_{t,\text{lon}}^{(i)}, m_{t,\text{lat}}^{(i)}\}$.

Definition 2 (Maneuver Sequence): A maneuver sequence $\pi_t^{(i)}$ for a vehicle i is defined as a tuple of t consecutive maneuvers $\{m_1^{(i)}, \dots, m_t^{(i)}\}$.

Definition 3 (Scene): The set $s_0 := \{\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(I-1)}\}$ composed by the current state of the AV $\mathbf{x}^{(0)}$ and the current states of all TVs $\mathbf{x}^{(i)}$, $i \in \{1, \dots, I - 1\}$, is called a scene where $\mathbf{x}^{(i)} := [p_{t,\text{lon}}^{(i)}, v_{t,\text{lon}}^{(i)}, p_{t,\text{lat}}^{(i)}, v_{t,\text{lat}}^{(i)}] \in \mathbb{R}^4$ with position p and velocity v in longitudinal and lateral direction.

Definition 4 (Scenario): Initialized by a scene s_0 , a scenario s_t is denoted as $s_t := \{\pi_t^{(0)}, \pi_t^{(1)}, \dots, \pi_t^{(I-1)}\}$ with an arbitrary maneuver sequence $\pi_t^{(i)}$ for each vehicle with S_t referring to the number of scenarios at time step t .

Definition 5 (Tactical Decision Option): A series of maneuvers from time step 1 to T selected as a possible candidate for the decision making module, is considered as a tactical decision option (TDO), denoted as τ_n , with n ranging from 1 to N , where N represents the total number of options.

Definition 6 (Target Vehicle Scenario): Given a probability score for a scenario $\mathbf{P}[s_t]$, $\bar{s}_{\tau_n, \kappa} \in \{s_T | \pi_T^{(0)} \in \{\tau_1, \dots, \tau_N\}\}$ is called a target vehicle scenario, where $\kappa \in \{1, \dots, K\}$

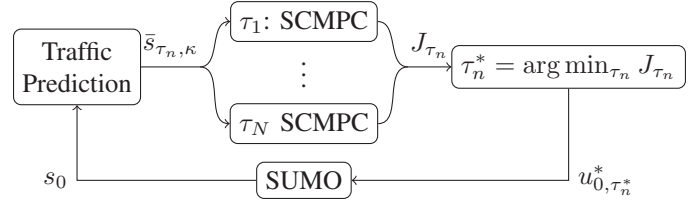


Fig. 1. Overview of the architecture for scenario-based interaction-aware decision making and planning [5] using the Simulation of Urban MObility (SUMO). For details see Section IV.

are the indices of the K most likely considered target vehicle scenarios.

B. Concept

In order to make informed decisions on the most favorable tactical decision option, the algorithm is divided into two main steps (Figure 1): traffic prediction (Section III) as well as scenario-based decision making and motion planning (Section IV). The traffic prediction, based on an approach by Bahram et al. [6] aims to identify the K most likely target vehicle scenarios for each considered tactical decision option of the autonomous vehicle by assigning a probability score. Given a reference velocity, each tactical decision option is then evaluated by the Scenario-based MPC while considering the desired front and rear safety margins for the associated K target vehicle prediction. In contrast to prior work in [5], we omit the interaction-unaware probability score in the traffic prediction and focus purely on time-to-collision and extend the approach to consider longitudinal maneuver options as well. Further, target vehicles can be considered *non- or partial-interaction-aware* by allowing only lane and velocity keeping or just longitudinal maneuvers, respectively. To decrease the number of considered scenarios, we do not permit certain maneuver sequences based on two criteria: (i) lane changes as well as (ii) changes in the direction of the longitudinal maneuver may only be made once over the horizon.

III. TRAFFIC PREDICTION

The purpose of the traffic prediction is to calculate a probability score for each considered scenario. Here, the probability score is estimated by the risk of collision, as commonly done in the automotive context via an approximation using *time-to-collision* (TTC) [32]

$$t_{\text{TTC}}^{(i,i+1)} = \frac{|\Delta p_{\text{lon}}^{(i,i+1)}|}{\max(\Delta v_{\text{lon}}^{(i,i+1)}, 0)}. \quad (1)$$

TTC is defined by the time it takes for a vehicle to closes the gap Δp_{lon} to the leading vehicle in the same lane, assuming a constant relative velocity Δv_{lon} . TTC can then be mapped to a pairwise risk using tuning parameters g_1 and g_2

$$R^{(i,i+1)} = \frac{1}{2} - \frac{1}{\pi} \arctan(g_1 (t_{\text{TTC}}^{(i,i+1)} - g_2)) \in (0, 1), \quad (2)$$

for the two considered vehicles. If there is the need to consider more than two vehicles in a scene, a comprehensive risk

$$R(s_t | s_{t-1}) = \frac{1}{I} \sum_i R^{(i,i+1)}, \quad (3)$$

can be calculated. The procedure is used in this work, for the predicted states of all vehicles given all possible scenarios as combinations of feasible maneuver sequences by mapping the computed risk to a probability score

$$\tilde{\mathbf{P}}[s_t|s_{t-1}] = 1 - R(s_t|s_{t-1}), \quad (4)$$

and normalized by the sum of all scenario probability scores

$$\mathbf{P}[s_t|s_{t-1}] = \frac{\tilde{\mathbf{P}}[s_t|s_{t-1}]}{\sum_{s_t \in S^t} \tilde{\mathbf{P}}[s_t|s_{t-1}]}. \quad (5)$$

With $\mathbf{P}[s_0] = 1$, the probability score for a possible scenario at time step T is calculated as

$$\mathbf{P}[s_T] = \prod_{t=1}^T \mathbf{P}[s_t|s_{t-1}]. \quad (6)$$

Further details can be found in [5].

IV. SCENARIO-BASED MOTION PLANNING

In order to find the optimal trajectory \mathbf{x}_t^* , SCMPC solves the finite horizon optimal control problem (FHOCPC)

$$\min_{\mathbf{u}} \mathbf{J}_t(\mathbf{x}_t, u_t) \quad (7a)$$

$$\text{s. t. } \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \quad (7b)$$

$$\mathbf{x}_0 = \mathbf{x}^{(0)}, \quad (7c)$$

$$\mathbf{x}_t \in \mathcal{Z}_{t,\tau_n}^{\text{front}}, \quad \forall t \in \{1, \dots, T\}, \quad (7d)$$

$$\mathbf{x}_t \in \mathcal{Z}_{t,\tau_n}^{\text{rear}}, \quad \forall t \in \{1, \dots, T\}, \quad (7e)$$

$$\mathbf{x}_t \in \mathcal{X}_t, \quad \forall t \in \{1, \dots, T\}, \quad (7f)$$

$$u_t \in \mathcal{U}, \quad \forall t \in \{1, \dots, T\}, \quad (7g)$$

for the optimal inputs u_t^* over the horizon T , minimizing cost $\mathbf{J}_t(\mathbf{x}_t, u_t)$ in (7a), considering system dynamics $f(\mathbf{x}_t, u_t)$ in (7b), starting at the initial/estimated state $\mathbf{x}^{(0)}$ with discretization time Δt . The constraint sets on input and states indicated by \mathcal{U} and \mathcal{X}_t are enforced in (7f)-(7g). In distinction to classical MPC, the SCMPC solution has to be feasible for all considered scenarios, i.e., for the set of most likely TV scenarios of each TDO τ_n of the AV, enforced by the sets $\mathcal{Z}_{t,\tau_n}^{\text{front}}$, $\mathcal{Z}_{t,\tau_n}^{\text{rear}}$ in (7d), (7e).

For each τ_n , the longitudinal position is constrained, such that a safe margin d_{front} has to be kept from the predicted position $p_{t,\text{lon}}^{(i)}$ of the closest TV, i.e., TV i , predicted to be in front ($p_{\text{lon},t}^{(i)} > p_{\text{lon},t}^{(0)}$) and on the same lane ($m_{t,\text{lat}}^{(i)} = m_{t,\text{lat}}^{(0)}$) as the AV according to τ_n and the associated $\bar{s}_{\tau_n,\kappa}$:

$$\mathcal{Z}_{t,\tau_n}^{\text{front}} := \left\{ p_{\text{lon},t}^{(0)} \mid p_{\text{lon},t}^{(0)} \leq p_{t,\text{lon}}^{(i)} - d_{\text{front}} + \lambda_{t,\kappa}^{\text{front}}, \right. \\ \left. i \in \bar{s}_{\tau_n,\kappa}, \forall \kappa \in \{1, \dots, K\} \right\}. \quad (8)$$

Analogously, the AV is set to stay in front of the predicted position $p_{T,\text{lon}}^{(i)} - d_{\text{rear}}$ of the last prediction step T of the closest following TV ($p_{0,\text{lon}}^{(i)} < p_{0,\text{lon}}^{(0)}$), i.e., TV i , being in the same lane ($m_{t,\text{lat}}^{(i)} = m_{t,\text{lat}}^{(0)}$), however, traveling initially in the other lane ($m_{0,\text{lat}}^{(i)} \neq m_{0,\text{lat}}^{(0)}$) minus a violation margin d_{rear} :

$$\mathcal{Z}_{t,\tau_n}^{\text{rear}} := \left\{ p_{\text{lon},t}^{(0)} \mid p_{\text{lon},t}^{(0)} \geq p_{T,\text{lon}}^{(i)} - d_{\text{rear}} - \lambda_{t,\kappa}^{\text{rear}}, \right. \\ \left. i \in \bar{s}_{\tau_n,\kappa}, \forall \kappa \in \{1, \dots, K\} \right\}. \quad (9)$$

We neglect the lateral component such that, in each time step, the leading and following vehicle is chosen as specified in the TV scenario $\bar{s}_{\tau_n,\kappa}$. To avoid infeasibility if a leading or following target vehicle merges closely in front or behind of the AV, the safety distances are implemented via soft constraints using the slack variables $0 \leq \lambda_{t,\kappa}^{\text{rear}} \leq 100$ and $0 \leq \lambda_{t,\kappa}^{\text{front}} \leq 100$, which are penalized by the same scalar factor $\xi > 0$ in the cost function.

The system dynamics in (7b) consist of a point mass model

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}, \quad (10)$$

in the form of $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}u_t$, as a satisfactory choice for motion planning [33]. The state $\mathbf{x}_t := [p_{t,\text{lon}}^{(0)}, v_{t,\text{lon}}^{(0)}] \in \mathbb{R}^2$ is composed of the longitudinal position and velocity and the input $u_t = a_{t,\text{lon}}^{(0)} \in \mathbb{R}$ of the longitudinal acceleration.

The architecture aims to find the optimal TDO τ_n^* that minimizes the cost J_{τ_n} :

$$\tau_n^* = \arg \min_{\tau_n} J_{\tau_n}. \quad (11)$$

To this end, we define (7a) as the sum over t of velocity tracking cost $\eta \|v_{t,\text{ref}} - v_{\text{lon},t}^{(0)}\|^2$, input cost $\|u_t\|_{\mu}^2$ as well as front and rear safety margin costs $\xi \|\lambda_{t,\kappa}^{\text{front}}\|^2$ and $\xi \|\lambda_{t,\kappa}^{\text{rear}}\|^2$:

$$\min_{\mathbf{u}} \sum_{t=1}^T \eta \|v_{t,\text{ref}} - v_{t,\text{lon}}^{(0)}\|^2 + \|u_t\|_{\mu}^2 + \xi \|\lambda_{t,\kappa}^{\text{rear}}\|^2 + \xi \|\lambda_{t,\kappa}^{\text{front}}\|^2. \quad (12)$$

The cost for a TDO is then defined by

$$J_{\tau_n} = \sum_{t=1}^T \eta \|v_{t,\text{ref}} - v_{t,\text{lon}}^{*(0)}\|^2 + \xi \|\lambda_{t,\kappa}^{*\text{rear}}\|^2, \quad (13)$$

using the optimal solutions $v_{\text{lon},t}^{*(0)}$ and $\lambda_{t,\kappa}^{*\text{rear}}$ from (7). We conduct this differentiation to have a planned trajectory that still satisfies other constraints such as a front safety margin or input cost which, however, are not considered as reasonable criteria for making a strategic decision. Situations where each lane is equally favorable are addressed by adding a bias to J_{τ_n} if the tactical decision foresees a lane change.

V. RESULTS & DISCUSSION

We compare the interaction-aware motion planning architecture to a baseline approach using the microscopic traffic simulator SUMO (Simulation of Urban Mobility) [34]. The baseline uses the same architecture, however assuming velocity and lane keeping predictions for all surrounding vehicles. In both cases, the input calculated by the SCMPC for the optimal TDO is applied to (10) in order to update the longitudinal position. For simplicity, the AV directly performs the lateral maneuver of the selected TDO. For simulating the behavior of traffic participants, we set up SUMO to use the Extended-Intelligent-Driver-Model (EIDM) [35] as a *state-of-the-art interaction model*. Full state measurement is assumed. The MPC is implemented in Python using the Gurobi solver [36]. One planning step takes approx. 0.7 seconds on a quad core Intel Core i5-10210U CPU @ 1.60 GHz and 16 GB RAM.

The traffic prediction is configured with tuning parameters $g_1 = 0.1$ and $g_2 = 5$, a horizon length of 6 seconds with

a step time of 2 seconds and the acceleration in longitudinal maneuvers $a = 2\text{m/s}^2$. The SCMP weights are $\eta = 100$, $\xi = 50,000$, and $\mu = 1$. The constraints are $v_{t,\text{lon}}^{(0)} \in [0, 30]\text{m/s}$, $a_{t,\text{lon}}^{(0)} \in [-4, 4]\text{m/s}^2$ and the safety distances $d_{\text{front}} = 50\text{m}$, $d_{\text{rear}} = 100\text{m}$. The sampling time for the SCMP is chosen to be 0.1 s. The desired velocity v_{ref} is set to $30\frac{\text{m}}{\text{s}}$. For each tactical decision option, the $K = 10$ target vehicle scenarios $\bar{s}_{\tau_n, \kappa}$ with the highest probability score are considered. We perform a pre-selection of AV maneuver sequences to restrict the number of TDOs τ_n to $N = 4$. Each option consists of velocity keeping and keeping the lane or changing the lane at one of the three time steps.

In order to account for the road geometry, three adjustments are made: (i) TVs driving on the upper lane are considered as partial-interaction-aware vehicles, i.e., vehicles which are considered to keep the lane in the traffic prediction, (ii) as soon as the AV switches to the upper lane, only the lane keeping TDO is possible for the EV. Lastly, (iii) the merging zone is handled analogously to a vehicle with zero velocity placed at 350 m in the lower lane.

Simulation results for two exemplary highway merging scenes are shown for three time steps. Both scenes consist of five vehicles traveling from left to right. The leading TV on the upper lane is set as non-interaction-aware vehicle. Predicted positions are represented by dots. Squares indicate the predicted positions assuming velocity keeping.

A. Case Study 1

First, the behavior of the baseline method is shown in Figure 2. The blue vehicle on the upper lane drives at a higher speed than the AV. Approaching the end of the lane, the AV chooses to slow down and keep the lower lane. A closer look at the cost of each TDO (see Figure 4) reveals that this is the preferable choice. Since the baseline assumes the blue vehicle keeps its speed, an undesirable rear safety distance cost would result from a merging maneuver. Subsequently, the AV is forced to keep its lane until a complete standstill.

In Figure 3, the interaction-aware approach is demonstrated for the same scene. Instead of keeping the lane, the AV makes a lane change decision. Paying attention to the default predicted positions, indicated by the square markers in Figure 5, it can be seen that the blue vehicle is expected to keep its velocity if the AV decides to stay in the lower lane. However, in case of a lane change decision, the interaction-aware approach expects the blue vehicle to yield in reaction, opening up a favorable merging opportunity.

B. Case Study 2

In a second scene, the roles are inverted. This time, the blue vehicle traveling in the front on the lower lane attempts to change the lane while approaching the merging zone. Since the baseline expects the TV to keep its lane it continues to keep the reference speed, leading nearly to a collision as seen in Figure 6. In contrast, the interaction-aware method anticipates the intention correctly and slows down in advance (Figure 7).

VI. CONCLUSION

We demonstrate an interaction-aware decision making and motion planning architecture for highway merging situations which aims to leverage the impact of the autonomous vehicle on surrounding road users in order to make strategic lane-change decisions to minimizing the slowdown of the autonomous vehicle and improving the overall awareness for the behavior of traffic participants. In the simulation study, we observe significant improvements in handling complex merging situations in free-flow highway traffic. The results highlight that the architecture is able to successfully anticipate the braking maneuver of the target vehicle as a reaction to changing to the left lane preventing a strong slow down for the autonomous vehicle. Further, we show that the algorithm can enhance the safety in some situations by braking proactively to increase the safety margin when recognizing that a leading target vehicle has the intention to merge.

REFERENCES

- [1] L. Fletcher, et al., ‘The MIT - Cornell collision and why it happened’, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, pp. 509-548, 2009.
- [2] Trautman, Peter and Andreas Krause (Oct. 2010). Unfreezing the robot: Navigation in dense, interacting crowds. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Taipei, Taiwan, pp. 797–803.
- [3] Wenshuo Wang; Letian Wang; Chengyuan Zhang; Changliu Liu; Lijun Sun, Social Interactions for Autonomous Driving: A Review and Perspectives, now, 2022
- [4] M.C. Campi, A. Carè, and S. Garatti, ‘The scenario approach: A tool at the service of data-driven decision-making’, *Annual Reviews in Control*, vol. 52, 2021, pp. 1-17.
- [5] Kensbock, Robin, Maryam Nezami and Georg Schildbach. ‘Scenario-Based Decision-Making, Planning and Control for Interaction-Aware Autonomous Driving on Highways.’ *2023 IEEE Intelligent Vehicles Symposium (IV)* (2023): 1-6.
- [6] M. Bahram, A. Lawitzky, J. Friedrichs, M. Aeberhard, and D. Wollherr, ‘A game-theoretic approach to replanning-aware interactive scene prediction and planning’, *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3981–3992, 2016.
- [7] Nair, Siddharth H., Vijay Govindarajan, Theresa Lin, Yan Wang, Eric H. Tseng and Francesco Borrelli. ‘Stochastic MPC with Dual Control for Autonomous Driving with Multi-Modal Interaction-Aware Predictions.’ *ArXiv abs/2208.03525* (2022): n. pag.
- [8] Nair, Siddharth H., Eric H. Tseng and Francesco Borrelli. ‘Collision Avoidance for Dynamic Obstacles with Uncertain Predictions using Model Predictive Control.’ *2022 IEEE 61st Conference on Decision and Control (CDC)* (2022): 5267-5272.
- [9] Benciolini, Tommaso, Tim Brüdigam and Marion Leibold. ‘Multistage Stochastic Model Predictive Control for Urban Automated Driving.’ *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (2021): 417-423.
- [10] Bonzanini, Angelo D., Ali Mesbah and Stefano Di Cairano. ‘Multi-stage Perception-aware Chance-constrained MPC with Applications to Automated Driving.’ *2022 American Control Conference (ACC)* (2022): 1697-1702.
- [11] Wang, Renzi, Mathijs Schuurmans and Panagiotis Patrinos. ‘Interaction-aware Model Predictive Control for Autonomous Driving.’ *2023 European Control Conference (ECC)* (2022): 1-6.
- [12] Gupta, P., David Isele, Donggun Lee and Sang Young Bae. ‘Interaction-Aware Trajectory Planning for Autonomous Vehicles with Analytic Integration of Neural Networks into Model Predictive Control.’ *2023 IEEE International Conference on Robotics and Automation (ICRA)* (2023): 7794-7800.
- [13] G. Schildbach and F. Borrelli, ‘Scenario model predictive control for lane change assistance on highways,’ *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 611-616.

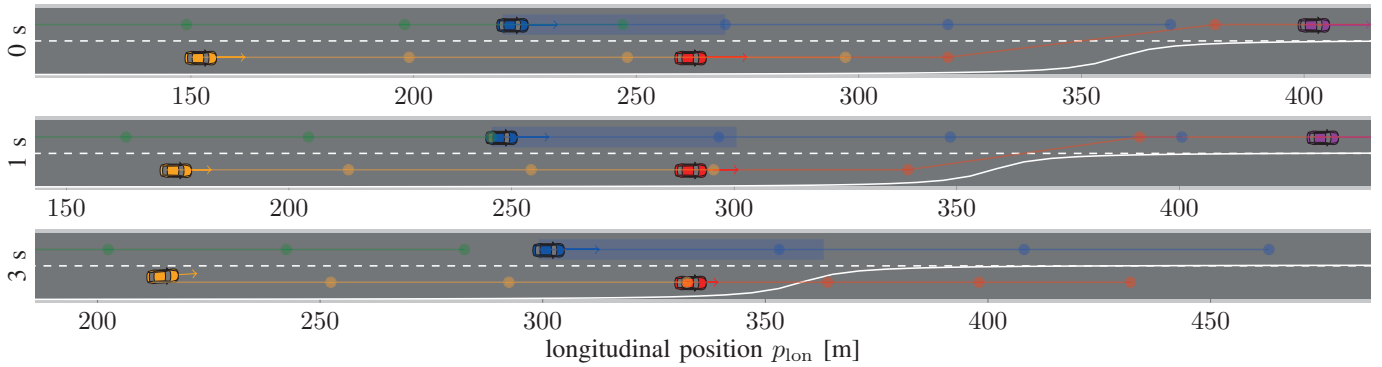


Fig. 2. Non-interaction-aware baseline: The autonomous vehicle is forced to keep the lane and slows down before reaching the end of the lane, since entering the safety margin (blue area) induces a high cost for the lane change decision option.

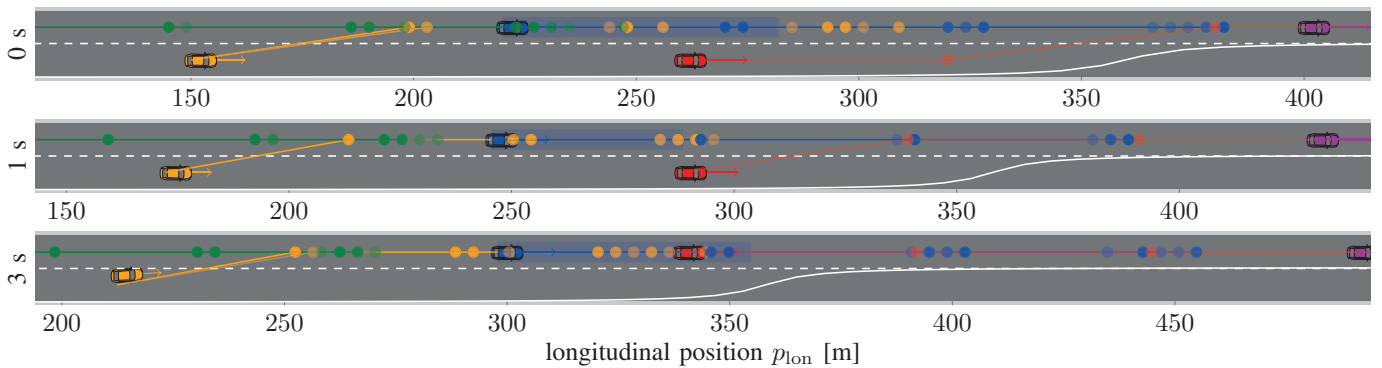


Fig. 3. Interaction-aware architecture: As the safety margin (blue area) shrinks, the autonomous vehicle (red) decides to change the lane at $t = 1$ s in front of the blue target vehicle.

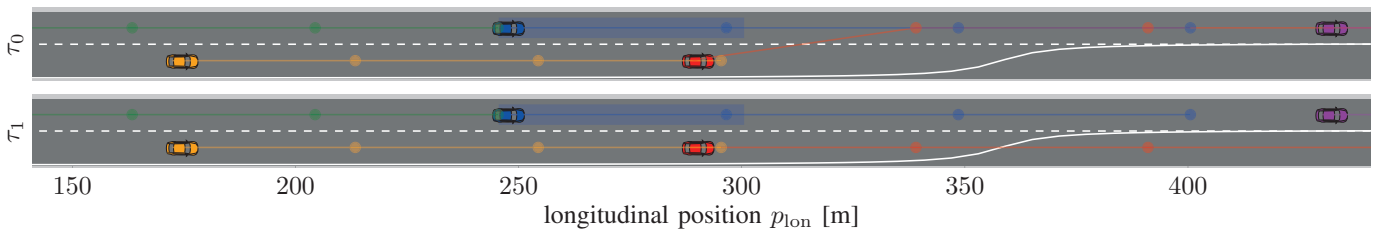


Fig. 4. Non-interaction-aware baseline: Predictions for tactical decision options τ_n of the autonomous vehicle. The blue vehicle is predicted to keep its speed regardless of the maneuver of the AV.

- [14] G. Cesari, G. Schildbach, A. Carvalho and F. Borrelli, "Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways," in *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 23-35, 2017.
- [15] C. H. Ulfssjoo and D. Axehill, "On integrating POMDP and scenario MPC for planning under uncertainty – with applications to highway driving," *IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp.
- [16] Zhou, Jian, Björn Olofsson and Erik Frisk. "Interaction-Aware Motion Planning for Autonomous Vehicles with Multi-Modal Obstacle Uncertainty Predictions." *IEEE Transactions on Intelligent Vehicles* (2022).
- [17] Batkovic, Ivo, Ugo Rosolia, Mario Zanon and Paolo Falcone. "A Robust Scenario MPC Approach for Uncertain Multi-Modal Obstacles." *IEEE Control Systems Letters* 5 (2021): 947-952.
- [18] Brüdigam, Tim, Michael Olbrich, Marion Leibold and Dirk Wollherr. "Combining Stochastic and Scenario Model Predictive Control to Handle Target Vehicle Uncertainty in an Autonomous Driving Highway Scenario." *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (2018): 1317-1324.
- [19] V. Fors, B. Olofsson and E. Frisk, "Resilient Branching MPC for Multi-Vehicle Traffic Scenarios Using Adversarial Disturbance Sequences," in *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 838-848, Dec. 2022, doi: 10.1109/TIV.2022.3168772.
- [20] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," in *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5365-5372, 2022.
- [21] Oliveira, Rui, Siddharth H. Nair and Bo Wahlberg. "Interaction and Decision Making-aware Motion Planning using Branch Model Predictive Control." *2023 IEEE Intelligent Vehicles Symposium (IV)* (2023): 1-8.
- [22] Wei, Chao, Yuanhao He, Hanqing Tian and Yanzhi Lv. "Game Theoretic Merging Behavior Control for Autonomous Vehicle at Highway On-Ramp." *IEEE Transactions on Intelligent Transportation Systems* 23 (2022): 21127-21136.
- [23] Zhang, Qingyu, Reza Langari, H. Eric Tseng, Dimitar Filev, Steven Szwabowski and Serdar Coskun. "A Game Theoretic Model Predictive Controller With Aggressiveness Estimation for Mandatory Lane Change." *IEEE Transactions on Intelligent Vehicles* 5 (2020): 75-89.
- [24] Evens, Brecht, Mathijs Schuurmans and Panagiotis Patrinos. "Learning MPC for Interaction-Aware Autonomous Driving: A Game-Theoretic Approach." *2022 European Control Conference (ECC)* (2021): 34-39.
- [25] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu and Y. Xing, "An integrated framework of decision making and motion planning for autonomous

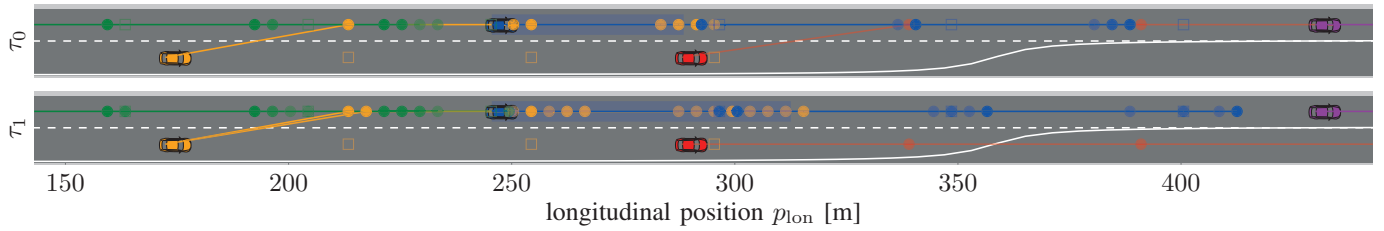


Fig. 5. Interaction-aware architecture: Predictions for two tactical decision options τ_n of the autonomous vehicle (red). The blue vehicle is predicted to slow down if the AV changes the lane, highlighted by comparing default (squares) and interaction-aware predictions (dots).

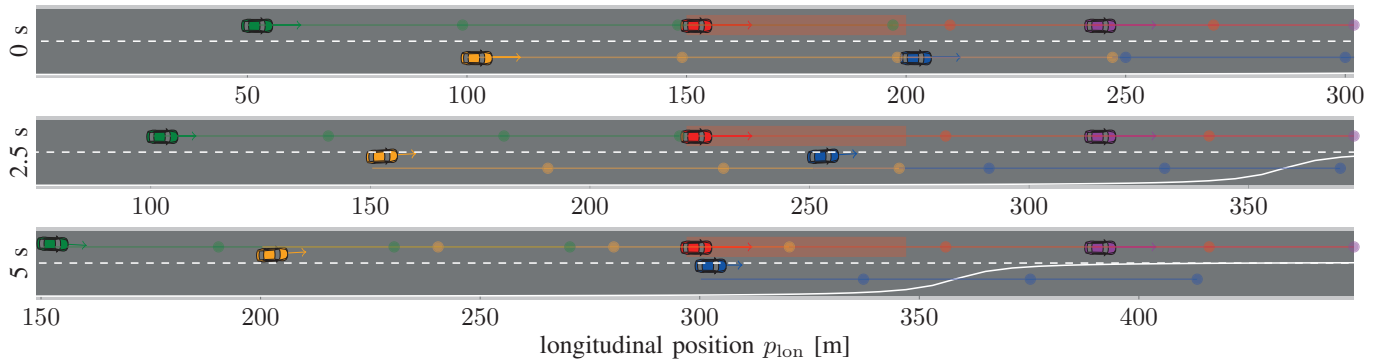


Fig. 6. Non-interaction-aware baseline: The blue target vehicle is predicted to keep the lane. A near collision follows as the autonomous vehicle continues to travel with the reference speed.

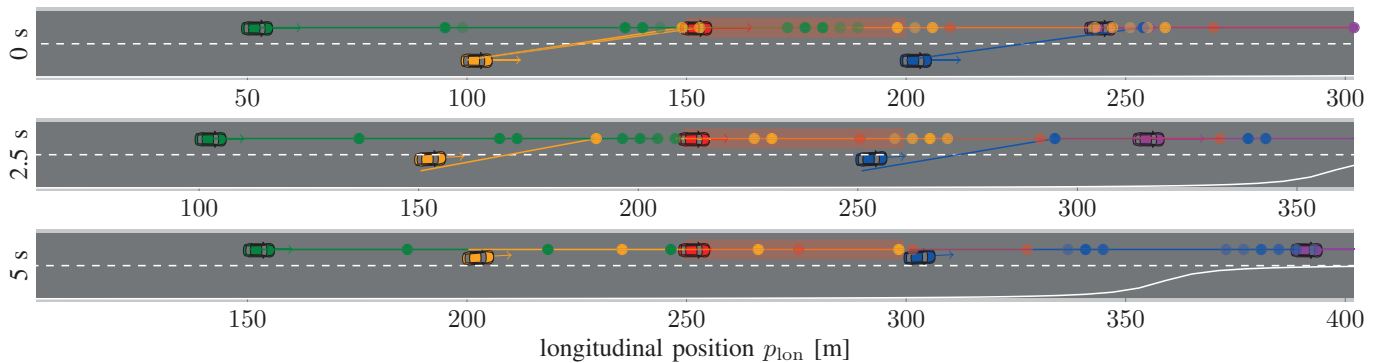


Fig. 7. Interaction-aware architecture: The lane change of the blue target vehicle is predicted. The autonomous vehicle slows down to maintain the safety margin (red area).

- vehicles considering social behaviors,” in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14458-14469, Dec. 2020.
- [26] Liu, Kaiwen, Nan I. Li, H. Eric Tseng, Ilya V. Kolmanovsky and Anouck R. Girard. “Interaction-Aware Trajectory Prediction and Planning for Autonomous Vehicles in Forced Merge Scenarios.” *IEEE Transactions on Intelligent Transportation Systems* 24 (2021): 474-488.
- [27] Bae, Sang Young, David Isele, Alireza Nakhaei, Peng Xu, Alexandre Miranda Añon, Chiho Choi, Kikuo Fujimura and Scott J. Moura. “Lane-Change in Dense Traffic With Model Predictive Control and Neural Networks.” *IEEE Transactions on Control Systems Technology* 31 (2023): 646-659.
- [28] Bouzidi, Mohamed-Khalil and Ehsan Hashemi. “Interaction-Aware Merging in Mixed Traffic with Integrated Game-theoretic Predictive Control and Inverse Differential Game.” *2023 IEEE Intelligent Vehicles Symposium (IV)* (2023): 1-6.
- [29] Le Cleac’h, Simon, Mac Schwager and Zachary Manchester. “LU-CIDGames: Online Unscented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning.” *IEEE Robotics and Automation Letters* 6 (2020): 5485-5492.
- [30] Lubars, Joseph, Harsh Gupta, A. Sivanantha Raja, Ramakrishnan Srikant, Liyun Li and Xinzhou Wu. “Combining Reinforcement Learning with Model Predictive Control for On-Ramp Merging.” *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (2020): 942-947.
- [31] Brito, Bruno Ferreira de, Achint Agarwal and Javier Alonso-Mora. “Learning Interaction-Aware Guidance for Trajectory Optimization in Dense Traffic Scenarios.” *IEEE Transactions on Intelligent Transportation Systems* 23 (2022): 18808-18821.
- [32] J. Dahl, G. R. de Campos, C. Olsson, and J. Fredriksson, ‘Collision avoidance: A literature review on threat-assessment techniques,’ *IEEE Trans. Intell. Veh.*, vol. 4, no. 1, pp. 101–113, 2019.
- [33] J. K. Subosits and J. C. Gerdes, ‘Impacts of model fidelity on trajectory optimization for autonomous vehicles in extreme maneuvers,’ *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 546–558, 2021.
- [34] P. A. Lopez et al., ‘Microscopic traffic simulation using SUMO’, in *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.
- [35] Salles, Dominik, Kaufmann, Stefan, Reuss, Hans-Christian. (2022). Extending the Intelligent Driver Model in SUMO and Verifying the Drive Off Trajectories with Aerial Measurements. SUMO Conference Proceedings. 1. 1-25. 10.52825/scp.v1i.95.
- [36] Gurobi Optimization, LLC, ‘Gurobi optimizer reference manual,’ 2024.

Coaction between Automobiles and Mobile Robots - Interoperability for Affordable Last Mile Delivery Solutions

Mayank Yogesh Khandelwal¹, Swaraj Tendulkar¹, G. Alexander Kolbai², Frank Schrödel¹

Abstract—This research paper delves into the interplay between autonomous mobile robots and automobiles, examining their interoperability, shared software stacks, and potential vehicle-to-robot communication. It underscores the critical role of ROS (Robot Operating System), simulation environments like CARLA and Autoware, while addressing the challenges faced by mobile robots in environments designed for automobiles. The study explores hurdles faced by companies in robot development, such as the absence of serial production leading to increased costs. It further emphasizes the quest for a sensor set and algorithm that balances cost efficiency with safety. Navigational challenges for outdoor robots are dissected. The research also touches on the intricate communication required with pedestrians and vehicles. The paper advocates for the integration of Global navigation satellite system (GNSS) and Depth perception technology to enhance navigation, emphasizing the need for collaborative efforts to address urban challenges. The ultimate aim is to foster collaboration between vehicles and mobile robots, ensuring interoperability for affordable last-mile delivery solutions, while also acknowledging and resolving the complex challenges faced by developers in this evolving field.

Index Terms—Robot Operating System, Vehicle-to-Robot Communication, GNSS Sensors, Mobile Robot Challenges, Last Mile Delivery Solutions, Interoperability

I. INTRODUCTION

In the dynamic urban mobility landscape, the interplay between autonomous mobile robots and vehicles unfolds a shared tapestry of challenges. Central to both entities is the formidable task of processing vast real-time data acquired from perception sensors, encompassing critical insights into traffic conditions, pedestrian dynamics, environmental shifts, and potential obstacles [16]. Efficiently navigating this data is pivotal for ensuring the safe and reliable operation of these autonomous systems. Moreover, the constrained environments of sidewalks, where these robots predominantly operate, pose unique challenges due to the diverse range of users sharing limited space [3]. Compounding these complexities are the hurdles associated with developing and deploying software stacks, requiring careful attention to compatibility, security, and reliability.

Mayank Yogesh Khandelwal¹, Swaraj Tendulkar¹, Frank Schrödel¹ is with Faculty of Mechanical Engineering, Schmalkalden University of Applied Sciences, 98574 Schmalkalden, Germany.

G. Alexander Kolbai² CEO of delivery-me, Germany, is gratefully acknowledged for his invaluable support and contributions to this research

Companies at the forefront of autonomous deliveries, including Starship Technologies, Uber, FedEx, Waymo, Amazon etc, face a myriad of challenges in achieving the efficient and widespread implementation of autonomous delivery systems[4]. Ensuring the safety of low-cost autonomous systems, particularly in pedestrian-heavy urban environments, emerges as a critical concern. The continuous technological challenge lies in reliably detecting and avoiding obstacles, ranging from pedestrians, pets and cyclists to other vehicles. Robust real-time perception and decision-making capabilities become even more complex in the face of environmental factors and intricate traffic scenarios. Even the public trust in autonomous delivery systems plays a pivotal role in their adoption, necessitating strategies to address these concerns. Furthermore, successful deployment requires robust infrastructure support, encompassing smart city technology, dedicated pickup and drop-off points, and seamless integration with existing transportation networks, government laws and policies. Adverse weather conditions and extreme temperatures add an additional layer of complexity, impacting the consistency of autonomous system performance [5].

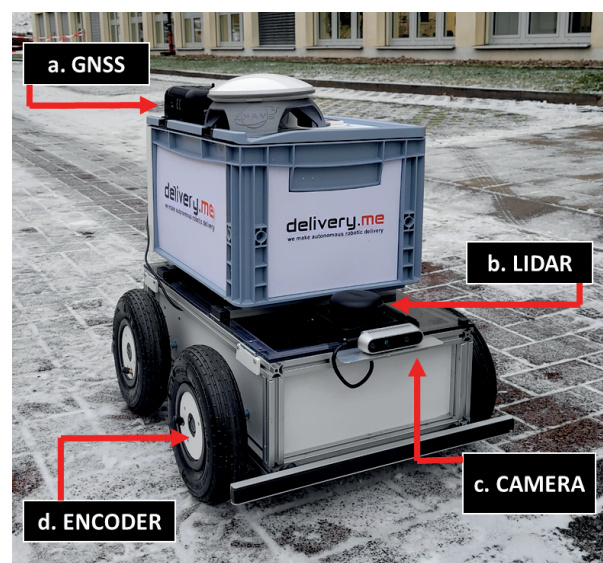


Fig. 1. Sensor Setup on delivery-me Robot

In the realm of urban logistics, low-cost delivery robots have emerged as a promising solution for last-mile delivery services. This paper explores the synergies between these robots, particularly those from delivery-me, as illustrated in figure 1, and an autonomous vehicle. Shared core principles of perception, decision-making, and control form the basis for a collaborative approach [1]. Aligning their capabilities unveils the potential for these robots to seamlessly operate in intricate urban environments, providing efficient and reliable delivery services.

The establishment of effective vehicle-to-robot communication adds another layer of intricacy, demanding seamless integration of diverse software and hardware systems. Envisioned scenarios include collaborative traffic management, real-time data sharing for dynamic route planning, and enhanced situational awareness for safer navigation [6]. Integrating robots into this ecosystem not only amplifies their capabilities but also contributes to the broader transformation of urban logistics and transportation [4].

Using the example of GNSS data, the research illustrates the need for distinct waypoints for mobile robots. While global positioning coordinates are readily available, such as those provided by widely-used mapping services, are optimal for vehicles, they fall short for mobile robots. The reliance on a High-Quality (HQ) Map specifically tailored for an operational environment, is imperative for autonomous robots. Thus, the integration of GNSS sensors for mapping and creating HQ maps separately becomes a necessity. Results from GNSS and camera lane detection showcase practical solutions, further contributing to the comprehensive understanding of these challenges in urban mobility.

The delivery-me robot stands out as a cost-effective solution in the domain of autonomous last-mile delivery [7]. Its sensor suite, as portrayed in figure 1, which includes Inertial Measurement Unit (IMU), Lidar, Stereo Depth Cameras [7], and an onboard computer, closely mirrors components used in the automotive industry. This alignment not only reduces procurement and operational costs but also underscores the reliability and robustness of these components. Scalability is a key advantage, allowing seamless adaptation to evolving delivery needs without significant additional costs. Relying on automotive-inspired components translates into lower maintenance costs, thanks to readily available parts and established service networks.

Hence this study explores the complex difficulties encountered by autonomous systems, with a specific focus on integrating low-cost delivery robots into urban mobility scenarios. Our exploration covers shared principles, technology integration, and real-world applications. Contributing to the discourse on developing efficient last-mile delivery solutions in complex urban settings. A key emphasis is placed on the importance of High-Quality (HQ) maps tailored for mobile robots, highlighting the challenges unique to their navigation compared to full-scale vehicles.

II. COMMONALITIES AND CHALLENGES: AUTONOMOUS VEHICLES VS. DELIVERY ROBOTS

Introduction to Shared Principles

In the rapidly evolving landscape of urban mobility, autonomous vehicles and low-cost delivery robots encounter a shared set of challenges rooted in their commonalities. Moreover both share essential hardware components, as presented in figure 2, that serve as the backbone of their operational capabilities. Both systems utilize advanced sensor suites, including LIDAR for accurate environmental mapping and stereo cameras for enhanced perception through depth information. The motor and encoder units play a pivotal role in translating decisions into precise physical actions, facilitating accurate navigation. Additionally, sophisticated battery management and distribution units ensure optimal power supply, addressing the energy requirements of these autonomous systems.

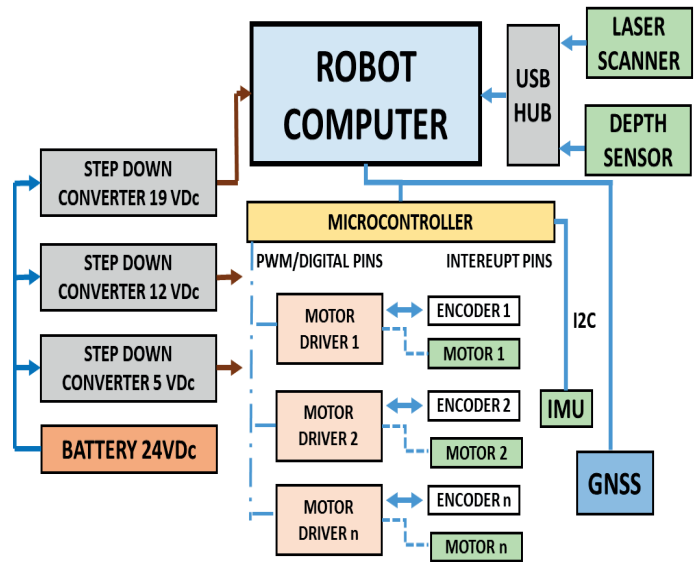


Fig. 2. Hardware Architecture [7]]

Both depend on sophisticated algorithms and well-structured software architectures to navigate, perceive and control their movements. The intricate software frameworks serve as the bedrock of their operations, facilitating tasks such as route planning, obstacle avoidance, and real-time decision-making. In the context of software architecture, both entities share commonalities in terms of perception, decision-making, and control principles. This convergence allows principles developed for one to be seamlessly adapted, refined, and shared with the other. The synergy in software architecture not only promotes interoperability but also sets the stage for collaborative advancements in autonomous urban mobility.

The need for real-time data processing also becomes a critical focal point. Both entities must efficiently interpret vast amounts of data acquired from sensors. This data encompasses crucial information on dynamic environment. The challenge lies not only in processing but also in responding to it effectively to ensure the safe and reliable operation.

Role of Robot Operating System

ROS emerges as a pivotal element in unifying the software architecture of autonomous vehicles and delivery robots. It provides a common framework that streamlines the development and integration of diverse software components. Its modular and open-source nature facilitates seamless communication between different modules, allowing for the efficient deployment of algorithms across a spectrum of robotic applications. This shared reliance on ROS not only promotes consistency in software development but also eases the challenges associated with compatibility and integration.

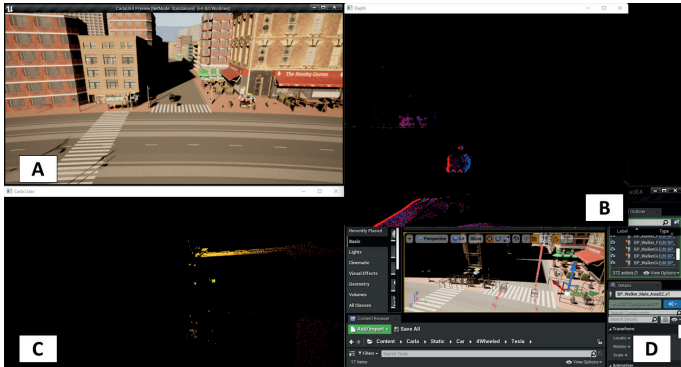


Fig. 3. CARLA simulation where A and D : Generalized Visuals B: Depth Information captured by a Camera C: Laser Scan obtained from a LIDAR

Simulation environments, such as CARLA and Autoware, play a crucial role in shaping and refining the behavior of both autonomous vehicles and delivery robots. These environments offer a controlled virtual space for testing, optimizing, and validating algorithms before real-world deployment. Simulation environments subject systems to different scenarios, aiding in identifying and addressing potential challenges. This contributes to enhancing the overall robustness of autonomous systems, as shown in Figure 3.

In this exploration of shared principles, we lay the groundwork for a comprehensive understanding of the interplay between vehicles and robots. By emphasizing their commonalities in software and hardware stacks, the role of ROS, and the utilization of simulation environments, we set the stage for a Detailed analysis of the challenges and collaborative potentials.

Infrastructure Disparity: Vehicles vs. Robots

Navigating the ever-changing landscapes of urban environments presents a shared set of challenges for both cars and robots within their Operational Design Domains (ODDs). While vehicles benefit from well-defined environments, established infrastructure, and public acceptance, robots face unique hurdles in diverse user interactions, and congested spaces. These, with their structured roadways, standardized traffic rules, and designated lanes, operate within an environment designed to accommodate them. The integration of vehicles into urban settings is streamlined, with clear guidelines, infrastructure established to support their movements and even the public acceptance is well-established.

Contrastingly, mobile robots operating on sidewalks encounter distinctive challenges posed by pedestrian-heavy environments. Negotiating through diverse user interactions becomes complex as robots share confined spaces with pedestrians. Unlike automobiles, robots must adapt to the unpredictable movements of pedestrians, varied speeds, and the intricacies of navigating crowded walkways. Confined spaces, originally designed for foot traffic, amplify the difficulty, requiring agile and adaptive systems for robots to operate efficiently.

Moreover, the established infrastructure, public acceptance, and regulatory frameworks that favor automobiles become hurdles for mobile robots. Sidewalks, primarily intended for Humans, lack predefined rules for robotic operations. Public acceptance of robots sharing walkways with pedestrians is an ongoing challenge, as societal norms and expectations are still evolving [3].

The convenience experienced by vehicles in a well-defined urban ODD contrasts sharply with the challenges faced by mobile robots navigating through diverse and unpredictable confined spaces. While vehicles seamlessly integrate into established urban infrastructure, mobile robots are forging a path through uncharted territory. Grappling with issues related to user acceptance, social norms, and the absence of clear guidelines for operation within their designated operational domains. As urban environments continue to evolve, addressing these challenges is crucial for the successful integration of mobile robots into the fabric of dynamic urban spaces.

Navigating Challenges in Constrained Environments

The execution of drive domain tasks within confined settings presents distinctive challenges for both vehicles and delivery robots. This section explores the specific hurdles faced, emphasizing the profound impact of size differences on their navigation and operational precision. Unlike vehicles, which can tolerate shifts on the road, even minor offsets for delivery robots navigating pavements can lead to potential hazards. For instance, a 20-30 cm offset in localization poses a significant risk for a robot, potentially causing it to veer off the pavement onto the road. The necessity for precise localization on sidewalks demands advanced perception systems for delivery robots, making the seamless execution of drive domain tasks a complex undertaking in constrained urban environments.

Computation Power and Sensor Limitations

Analyzing the constraints of low computation powers in delivery robots unveils a critical aspect of their operational challenges. Limited space and weight restrictions inherent to robots impact their computational capabilities, setting them apart from full-scale vehicles. Unlike cars, which can house powerful computers and robust power sources, delivery robots face restrictions that affect their range, decision-making speed, and overall operational efficiency. This section delves into the implications of these constraints on decision-making processes, emphasizing the need for advancements in computational capabilities within the confined parameters of size and weight.

Moreover, the discussion extends to the impact of limited sensors and range on the decision-making processes of delivery robots. Comparing with vehicles emphasizes a clear difference in capabilities. While cars benefit from an array of sensors and extended range, small robots on footpaths deal with sensor limitations that impact their ability to perceive and respond to dynamic urban environments. Additionally, the section sheds light on how offsets in sensor readings affect small robots navigating footpaths compared to vehicles on highways. The precision required for a robot to interpret sensor readings becomes paramount in confined spaces, where minor deviations can lead to potential hazards, contrasting with the more forgiving dynamics of vehicles on broader roadways.

Hence, the unique constraints imposed by size differences and limited computational capabilities underscore the need for innovative solutions to enhance the operational efficiency and safety of these autonomous entities.

III. CHALLENGES IN GNSS DATA FOR DELIVERY ROBOTS: A CASE STUDY

The Need for HD Maps in Last-Mile Delivery

In the dynamic landscape of autonomous mobile robots, the reliance on accurate localization is paramount for effective navigation. The complex process of autonomous localization relies on probabilistic data fusion techniques, integrating information from various sensors to continually refine the robot's position and orientation. In this context, outdoor solutions benefit from an essential resource, denied to indoor applications: the global navigation satellite system.

Mobile robots often demand the creation of High-Definition (HD) maps, also known as 3D maps [10]. These maps are meticulously detailed, providing inch-perfect accuracy and high environmental fidelity. Unlike traditional maps, HD maps contain information about the precise positions of pedestrian crossings, traffic lights/signs, barriers, and more as outlined in figure 5. This level of detail is essential for autonomous vehicles, compensating for map inaccuracies, but for delivery robots, such discrepancies pose substantial challenges.

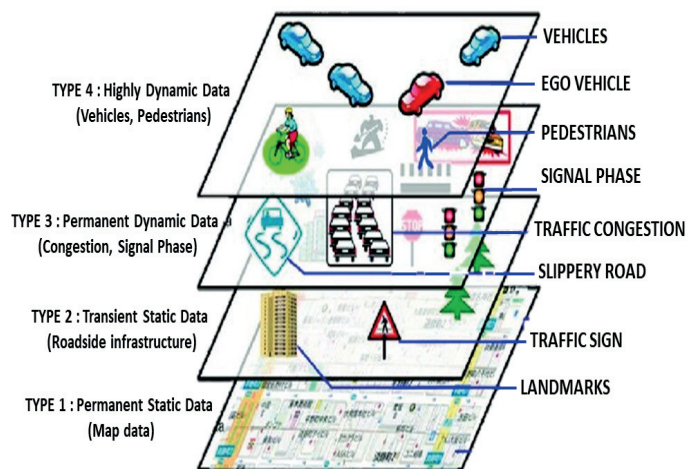


Fig. 4. Intricate details captured within a High-Definition (HD) Map [10]

Delivery robots face unique challenges, navigating pedestrian-heavy environments where conventional maps fall short. HD maps for robots encompass complex data, including roads, crossings, pavements, landmarks, dynamic obstacles and other terrain features. While autonomous vehicles can tolerate certain inaccuracies in traditional maps, delivery robots require a higher level of precision due to the intricacies of their deployment areas.

Creating these HD maps involves intricate mapping processes, ensuring that the robot can interpret and navigate the environment with precision. The goal is to provide the robot with a roadmap that aligns with the reality of footpaths, pavements, and other pedestrian-centric spaces. These maps are critical for last-mile delivery applications, where robots must navigate seamlessly in diverse and crowded urban settings.

IV. TESTING

To address the inadequacy of traditional GPS data for delivery robots, experiments were conducted to showcase the disparities between conventional maps and delivery-specific HD maps. Conventional mapping platforms like Google, Apple, and OpenStreetMap, while suitable for human decision-making, lack the precision required for robot navigation in intricate urban environments [12].



Fig. 5. Points derived from Google Maps (Blue) compared to Waypoints obtained through ANavS GNSS (Yellow)

These maps often misrepresent waypoints for pedestrians, placing them on roads instead of footpaths. As a result, relying on these waypoints directly could lead a robot astray, emphasizing the need for tailored maps specific to the robot's deployment area. Significant offset can be seen in figure 5.

Implementing the ANavS Global Navigation Satellite System module, specifically the Multi-Sensor Real-Time Kinematic Module, becomes pivotal for achieving centimeter-level accuracy in localization. ANavS is a company specializing in the development of highly accurate positioning and navigation solutions with sensor fusion and Artificial Intelligence.

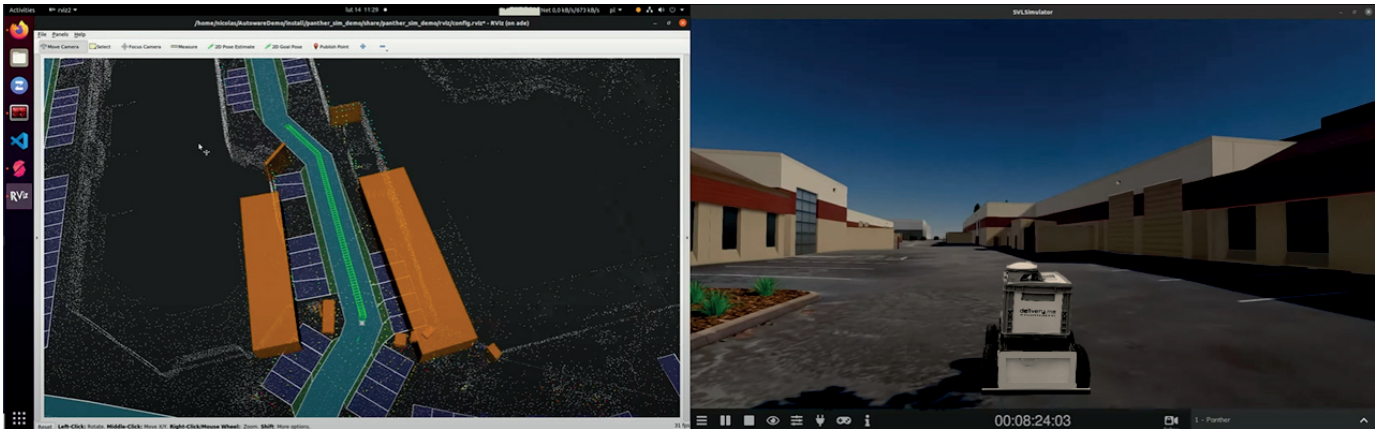


Fig. 6. AUTOWARE-ROS Simulation and Visualization of test Environment

The GNSS, with budget multi-frequency/multi-constellation receivers, allows the application of Real-Time Kinematic techniques. The ROS middle-ware, widely adopted in the robotics community, facilitates seamless integration of the RTK code.

Simulation software assists in creating and visualizing the robot's environment, offering a platform for testing and refining its behavior before actual deployment. Here, the start and end positions of the route are marked, and the robot's behavior is simulated [13]. The robot, equipped with various sensors, replicates real-world scenarios. This simulation helps refine algorithms for real-world deployment as highlighted in figure 6.

Output solution streams from the ANavS GNSS module are configured independently and forwarded through various devices, including serial port, TCP socket, NTRIP (Networked Transport of RTCM via Internet Protocol) server, or stored in a local file (ROS Bag File) [9]. All parameters related to processing options and input/output streams are stored in a configuration file at the start of the program.

The ANavS provides a command line console, reachable over a telnet connection, for process monitoring and control. This allows for human supervision of the precise positioning and calibration process, ensuring the RTK server thread's effective operation as seen in figure 7 [15]. The RTK technique stands out for its rapid convergence time and high positioning accuracy, reaching centimeter-level precision almost instantaneously with multi-frequency receivers.

In the ROS environment, nodes communicate through topics, exchanging messages to define the structure of transmitted data. Several ROS messages are necessary to accommodate the data of different output formats conveniently. The independent operation of output streams allows the combination of different devices to communicate information simultaneously [9].

We incorporate GNSS way-points, bearing vectors, and distances from the robot's camera to patch features as state vectors in our Kalman filter, demonstrating achievable positioning accuracies. The assumption of constant way-points and the robot's low-speed movement simplifies the state space model. We incorporate the robot's position change into the process noise, using a Kalman filter for GNSS. The approach involves an iterative process, considering linearization of range measurements and specific movements for waypoint convergence and robot positions.

For visual-inertial odometry, we employ a Kalman filter processing images from a camera and measurements from an inertial sensor. Built on the Robust Visual-Inertial Odometry (ROVIO) framework, it tracks the vectors and distance of each patch feature as a state parameter, alongside the position, velocity, attitude, and biases of the IMU. The process involves predicting state parameters using inertial measurements, searching for feature patch locations in new camera images, and updating the state vector based on found feature patches. Position, velocity, and attitude estimates obtained serve as measurements for the main EKF Localization as shown in figure 8. This main filter, incorporating all sensor readings, updates its state vector with all the new parameters including the carrier phase ambiguities, pseudorange multipath errors, and wheel odometry.

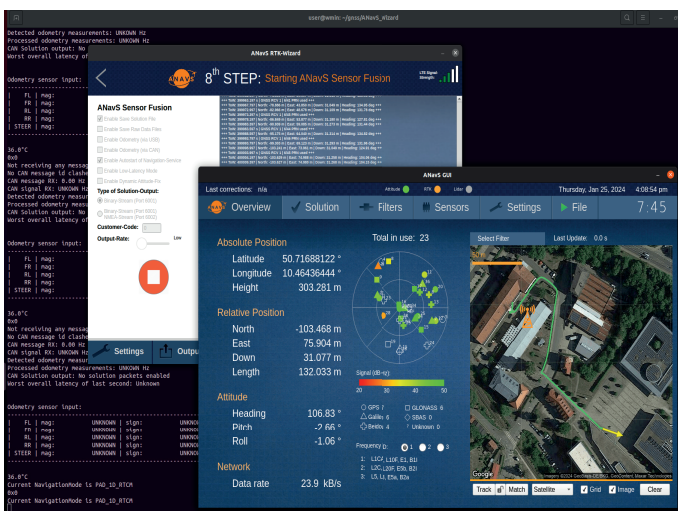


Fig. 7. Multi-Sensor Real-Time Kinematic (RTK) on the ANavS Graphical User Interface (GUI)

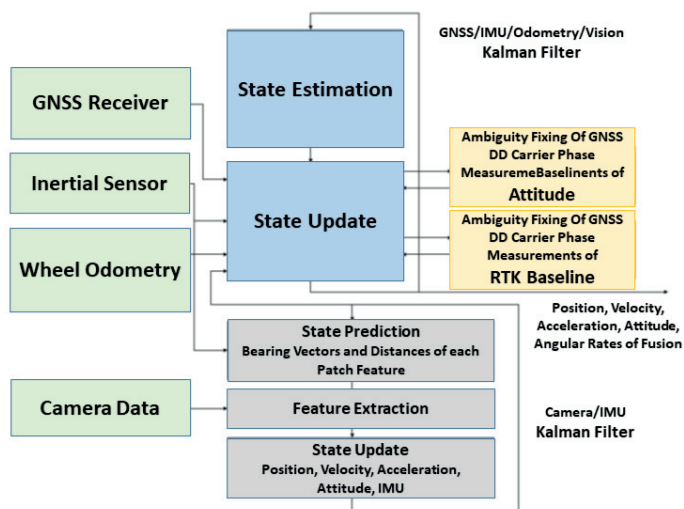


Fig. 8. Architecture for Sensor Fusion of GNSS, IMU, Wheel-Odometry and Visual Odometry in Kalman Filter [9]

In a typical scenario, the first output stream publishes the robot's position in the East North Up (ENU) reference frame. Simultaneously, the second output, independent of the first, forwards NMEA output format through a TCP socket. This flexibility in communication ensures the efficient utilization of output data for various purposes[11].

The comprehensive case study explores the intricate challenges faced by robots in the context of localization. From the need for tailored HD maps to the implementation of advanced GNSS, each aspect plays a crucial role in enhancing the precision and reliability of delivery robots. Further integrating it with ROS as middleware and simulation environments, demonstrates a comprehensive method. to addressing the unique challenges posed by dynamic urban environments.

CONCLUSION

In conclusion, this paper sheds light on the intricate challenges faced by delivery robots in navigating dynamic urban landscapes, with a particular focus on GNSS data and localization. The significance of accurate localization for the effective functioning of autonomous mobile robots is a recurring theme, emphasizing the complexity inherent in the perception-driven process of autonomous localization. The distinction between traditional and High-Definition maps underscores the critical need for precision in last-mile delivery applications, where delivery robots traverse intricate pedestrian-centric spaces.

The integration of the GNSS module, specifically the MSRTK Module, emerges as a pivotal solution to achieve centimeter-level accuracy in localization [9]. The collaborative approach of integrating this module with the widely adopted Robot Operating System and simulation environments exemplifies a comprehensive strategy to address the unique challenges faced by delivery robots. Experiments showcasing disparities between conventional maps and delivery-specific HD maps highlight the inadequacy of traditional GNSS data for robots operating in urban environments.

Looking into the future, the scope of this approach extends beyond GNSS data. The inclusion of visuals obtained from depth cameras, point clouds, and other sensor data offers opportunities for sensor fusion approaches. By leveraging extended Kalman filters, the fusion of multiple automotive sensors promises precise localization and object detection during the robot's journey. This advanced sensor fusion approach aims to refine the capabilities of delivery robots, enabling seamless navigation and interaction in diverse and evolving urban landscapes.

Moreover, the future envisions a collaborative ecosystem where robots and vehicles work in tandem, leveraging shared principles in software and hardware stacks. The need for interoperability becomes crucial for the development and deployment of affordable solutions. As vehicles and robots share common challenges and environments, establishing seamless communication is paramount. Envisioning a future where autonomous systems can communicate effectively fosters a collaborative environment, contributing to the broader transformation of urban logistics and transportation. This interoperability not only enhances the capabilities of individual robots and vehicles but also lays the foundation for a more efficient, adaptive, and integrated urban mobility system.

REFERENCES

- [1] Kawser, M. T., Fahad, M. S., Ahmed, S., Sajjad, S., The Perspective of Vehicle-to-Everything (V2X) Communication towards 5G. International Journal of Computer Science and Network Security, 19(4) (2019)
- [2] Tević, S., Krunić, M., Dragojević, M., Kapročki, N., Development of ADAS perception applications in ROS and 'Software-In-the-Loop' validation with CARLA simulator. Telfor Journal, 12(1), 40-45 (2020)
- [3] Zug, S., Seyffer, N., Plank, M., Pfleging, B., Schrödel, F., Identification of Potential Conflict Zones Between Pedestrians and Mobile Robots in Urban Situations. 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA) 1-4) IEEE (2023)
- [4] Turning Cars into Robot Traffic Managers. Berkeley College of Engineering (2018)
- [5] Autonomous Delivery Robots: The Precursor to Self-Driving Cars, Forbes (2022, October 10)
- [6] Car 2 Car Communication Consortium "White Paper Connected And Cooperative Automated Driving (CCAD)" (2023 May)
- [7] delivery-me. delivery me - Robotic Delivery Services. <https://www.delivery-me.eu/robotics-1/deliver-robot/>
- [8] IntelRealSense. SLAM with D435i. <https://github.com/IntelRealSense/realsense-ros/wiki/SLAM-with-D435i>
- [9] Henkel, P., Sperl, A., Mittmann, U., Bensch, R., Farber, P., Gunther, C., Precise Positioning of Robots with Fusion of GNSS, INS, Odometry, Barometer, Local Positioning System and Visual Localization (February 2022)
- [10] Soorchaei, B., Mahdi, R., Romero, R., Raftari, A., Fallah, Y., High-Definition Map Representation Techniques for Automated Vehicles, Advance Technologies of Navigation for Intelligent Vehicles, (October 2022)
- [11] A Ferreira, B Matias, J Almeida, E Silva "Real-time GNSS precise positioning: RTKLIB for ROS" International Journal of Advanced Robotic Systems, (May 2020)
- [12] Dempsey, D., 11 Myths About GPS for Autonomous Vehicles, Electronic Design, (May 2019)
- [13] Duc, N., Husarion, Autoware.Auto Simulation Demo, husarion.com (February 2022)
- [14] D Dempsey, 11 Myths About GPS for Autonomous Vehicles, Electronic Design, (May 2019)
- [15] ANavS GmbH, Datasheet of ANavS Multi-Sensor Fusion RTK Module, v.3.5.5, Oct. 2018.
- [16] Amazon Abandons Autonomous Home Delivery Robot in Latest Cut. (2022, October 06). Bloomberg.

AI-Based Localization and Classification of Visual Anomalies on Semiconductor Devices

Minh Khai Le
Elmos Semiconductor SE
Dortmund, Germany
Minh-Khai.Le@elmos.com

Jason Zi Jie Chia
Elmos Semiconductor SE
Dortmund, Germany
jason-zi-jie.chia@elmos.com

Dennis Peskes
Elmos Semiconductor SE
Dortmund, Germany
dennis.peskes@elmos.com

Abstract—This paper presents an AI-based system for automated visual inspection of semiconductor components, aimed at improving the Zero-Defect strategy in their manufacturing process. The system leverages unsupervised learning using Variational Autoencoder to learn and compare images of undamaged components to identify anomalies. An anomaly score is devised to enable detection of even minor flaws on the edges of components and decision rules are evaluated using appropriate metrics. The proposed system surpasses the current tape machine in detecting anomalies, hence contributing to achieving the Zero-Defect strategy in semiconductor manufacturing.

I. INTRODUCTION

The semiconductor industry has experienced tremendous growth in recent years, and with this growth comes the need to optimize manufacturing processes to achieve maximum productivity. In particular, ensuring the functionality and reliability of semiconductors is critical to the performance and safety of devices and systems in various industries such as automotive, mobile phones, televisions, and computers [1].

To achieve high-quality standards, a Zero-Defect strategy must be adopted to ensure that only fully functional integrated circuits (ICs) are delivered to customers. This strategy requires strict monitoring of the manufacturing process through various inspection procedures, including physical measurements and visual inspections of the ICs [2].

However, the current tape machine's defect detection process is not completely error-free, resulting in wrongly identified functional ICs as defective and undetected defective ICs. Hence, there is a need to develop an AI-based system to improve the classification of ICs in the Tape & Reel, contributing to the Zero-Defect strategy.

II. CURRENT STATE OF SEMICONDUCTOR INSPECTION IN THE TAPE & REEL

The final visual inspection of semiconductors is performed by tape machines in the Tape & Reel stage of the manufacturing process. This stage involves the packaging of ICs in tape and reel for easy transportation and storage. The sorting of ICs by tape machines has a direct impact on yield, which is a crucial factor in assessing production efficiency. Yield is defined as the ratio of deliverable ICs to the total number of produced ICs [3]. Therefore, it is essential to sort out only defective components during inspections to maintain the highest possible yield.

A. Utilizing Tape Machines for Final Visual Inspection

The current classification mechanism of these tape machines uses simple image processing functions to inspect the ICs. An IC is set as a template and the inspection is based on the number of white pixels that deviate from the template. However, this method has limitations and can lead to false negatives, where defective ICs are not detected.

To improve the classification of ICs in the Tape & Reel, more advanced image processing techniques can be employed. These techniques can detect even the smallest defects that may be missed by the current tape machine. Implementing these techniques can maximize yield and improve production efficiency. Therefore, it is necessary to explore new methods to improve the inspection process and achieve the Zero-Defect strategy.

B. Defining the Requirements for an AI-Based Anomaly Detection System for ICs

Semiconductor manufacturing is a complex process that involves several stages of production, including cutting wafers into individual ICs. However, such a process can cause damage to the edges of the components, which can negatively impact their functionality. To ensure that defective ICs are sorted out while minimizing the overall yield, we develop an appropriate AI-based system for anomaly detection on the IC surface.

To begin with, the desired model must be placed after the tape machines. These machines perform a preliminary filtering by carrying out their usual classification of the ICs. The developed model should then recheck the images on which the tape machine does not detect any damage. If the model detects any damage, the corresponding IC will be sorted out.

To optimize the detection of defective ICs, anomalies located at the edges of the ICs should be sorted out since errors at the edge have a greater impact on functionality compared to errors in the middle of the IC. Therefore, it is crucial for the model to detect these types of anomalies at all times.

It is also essential to determine the types of damage to ICs the model should detect. For instance, scratches caused by marking are mainly superficial and do not impair the functionality of the ICs. Conversely, damage to the edges of the ICs, which can be severe, can significantly impact the

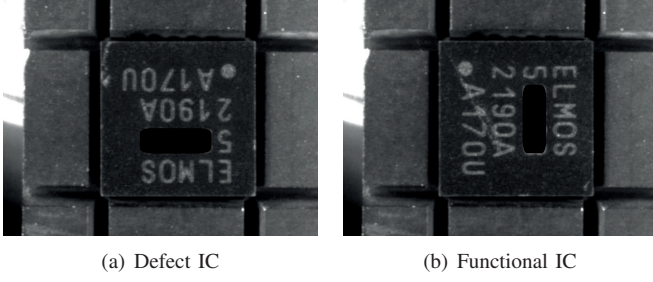


Fig. 1. Representation of two ICs with anomalies on the surface, where (a) is defective and (b) is intact.

functionality of the ICs. As such, the model should specialize in detecting such anomalies.

For comparison, fig. 1(a) shows a corresponding IC with damage at the upper left edge, which can be assumed to have completely broken off a significant portion of the component at the corresponding edge. Fig. 1(b) shows a IC with a scratch in the middle of the marking, which is assumed to have no functional impairment.

III. METHODS

The development of an AI-based system that can classify integrated circuits (ICs) based on photos taken by tape machines requires a modification of a Variational Autoencoder (VAE). This chapter focuses on the optimization of a VAE to accurately reconstruct images without anomalies. The VAE is trained to model the data distribution of images of ICs without existing anomalies, allowing it to generate images without anomalies and compare them with input images. The difference between the images can be used for anomaly localization and identification.

A. The Concept of Autoencoder

Autoencoders (AEs) are neural networks that are trained through unsupervised learning to learn reconstructions that resemble the original input. They have a wide range of potential uses, including clustering tasks and anomaly detection [4]. An undercomplete AE, which is the minimum configuration for an AE, is presented in fig. 2 [5]. The encoder's task is to compress the input vector using a lower-dimensional latent variable in the latent space, while the decoder's task is to reconstruct the input vector using the latent variable as input [6]. The encoder and decoder of

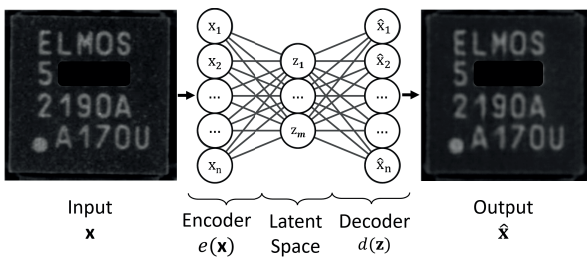


Fig. 2. Structure of an *undercomplete* Autoencoder network [6].

an AE are trained to minimize the loss function given by equation

$$\arg \min_{e,d} \mathbb{E}[V(\mathbf{x}, d(e(\mathbf{x})))], \quad (1)$$

where the chosen loss function V is typically the Mean-Squared-Error (MSE) [6].

AEs can also be used for anomaly detection, particularly in industrial applications, where they are trained on normal images and subsequently tested on anomalous images. However, due to the high complexity of images, AEs may overfit to the training data and perform poorly on anomalous images [8].

To address this limitation, the VAE adds a regularization term to the loss function, ensuring that the latent space is sufficiently regulated so that small variations within the space have only a small impact on the decoded output. This prevents overfitting and improves the accuracy of anomaly detection. The concept of an AE is then expanded to a VAE, which introduces the ability to generate input images from a predefined latent space [9].

B. Structure and Loss Function of the Variational Autoencoder

The VAE is an extension of the AE, which enables the generation of new images similar to the training data. The VAE consists of two neural networks: the inference network (encoder) and the generative network (decoder). The encoder maps the input image \mathbf{x} to a distribution $q_{\phi}(\mathbf{z}|\mathbf{x})$, where \mathbf{z} is a latent variable with a lower dimensionality than \mathbf{x} . The decoder maps \mathbf{z} to the reconstructed image $\hat{\mathbf{x}}$ using a distribution $p_{\theta}(\mathbf{x}|\mathbf{z})$. The encoder and decoder are trained jointly to minimize the reconstruction error between the input image and the reconstructed image [10]. The architecture of the VAE can be seen in fig. 3.

To ensure that the latent variable \mathbf{z} follows a prior distribution, typically a standard normal distribution $\mathcal{N}(0, 1)$, a regularization term \mathbb{KL} called Kullback-Leibler Divergence is added to the loss function. It measures the divergence between the inferred distribution $q_{\phi}(\mathbf{z}|\mathbf{x})$ and the prior distribution. The loss function of the VAE is the sum of the reconstruction error and the regularization term, defined as:

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \underbrace{-\mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction error}} + \underbrace{\beta \cdot \mathbb{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || \mathcal{N}(0, 1))}_{\text{regularization term}} \quad (2)$$

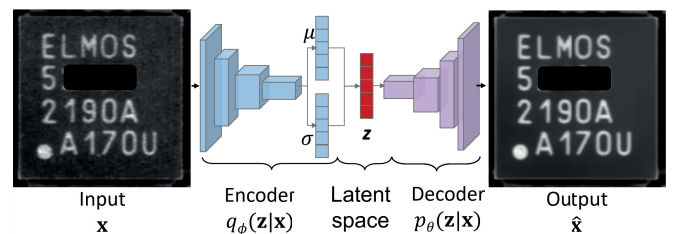


Fig. 3. Architecture of a Variations Autoencoder [10].

where \mathbb{E} denotes the expectation over the latent variable \mathbf{z} and $\beta > 0$ serves as a weighting parameter. The reconstruction error measures the difference between the input image and the reconstructed image, while the regularization term encourages the latent variable to follow a prior distribution. The parameters θ and ϕ are learned by backpropagation through the loss function. By incorporating a weighting factor, $\beta > 1$, in the loss function, the regularization term's strength is increased, leading to a more stable and robust disentanglement [7].

The VAE can generate new images by sampling from the prior distribution and using the decoder to generate a new image. The quality of the generated images depends on the representation capabilities of the encoder and decoder networks and the choice of the prior distribution [8].

IV. AUTOMATIC CLASSIFICATION OF SEMICONDUCTOR ANOMALIES

The focus of this chapter is on the detection of anomalies at the edges of ICs, which can be very small and when optimally reconstructed, may have only a low score. The challenge here is to detect such anomalies with a low MSE and distinguish them from the input images that have a score due to simple inaccuracies in the reconstruction.

A. Improved Anomaly Detection by Focusing on Relevant Areas

In order to detect fine anomalies on ICs, it is important to use an appropriate score for decision-making. The Mean-Squared-Error (MSE) is defined as

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2, \quad (3)$$

where x_i represents a pixel in the input image \mathbf{X} and \hat{x}_i represents the corresponding pixel in the reconstructed image $\hat{\mathbf{X}}$ and commonly used for evaluating the reconstruction quality. However, since there can also be differences between input and reconstruction images in the absence of anomalies, it is necessary to narrow down the relevant areas that are likely to contain anomalies. To demonstrate the effectiveness of focusing on relevant areas, we consider the example shown in fig. 4(a). An anomaly is present at the lower left corner of the IC and its detection is particularly challenging due to its small size. Fig. 4(b) shows the reconstruction image generated by the Variational Autoencoder, which is

very similar to the input image. The MSE of both anomaly maps is 0.0006 and not able to distinguish them by setting an threshold. To improve the anomaly detection, we only consider the pixels with high difference values. In this study we use 1% of the highest difference pixel. This way the MSE of fig. 4(a) changes to 0.0121 and fig. 4(b) to 0.0058 and easily set a threshold.

B. Local Anomaly Weighting

In addition to considering that anomalies on the edge should be detected more often than those in the center, as described in section II-B, errors occurring on the edge should have a stronger influence on the MSE calculation and thus on the anomaly score. To achieve this, a stronger weighting of the edge pixels is used. Since the location of an IC can vary in each image, the range of possible coordinates is limited. Therefore, the range of the edge area that needs to be weighted can be defined as the outer 10% of pixels in the image. The edge pixels are then weighted by a factor of 20 and incorporated into the MSE calculation for the anomaly score.

The weighted MSE is defined as follows:

$$AS = \frac{1}{\frac{1}{100} \cdot n} \sum_{i=1}^{\frac{1}{100} \cdot n} a_{(i)} w(i), \quad a_i = (x_i - \hat{x}_i)^2 \quad (4)$$

where AS is the anomaly score, n is the total number of pixels in the image, $a_{(i)}$ is sorted in descending order with x_i as the pixel intensity value of the input image, \hat{x}_i is the pixel intensity value of the reconstructed image and $w(i)$ is a function that weights the pixels based on their position.

The function w is defined as follows:

$$w(i) = \begin{cases} 1, & (\sqrt{n} \cdot l < i < \sqrt{n} \cdot u) \wedge (l < i \bmod \sqrt{n} \leq u) \\ 20, & \text{otherwise,} \end{cases}$$

where $l = \lfloor 0.1 \cdot \sqrt{n} \rfloor$ and $u = \lfloor 0.9 \cdot \sqrt{n} \rfloor$. The first condition of the logical operator removes the lower and upper 10% of edge of the image matrix. The second condition of the logical operator removes the left and right edge of the image. Thus the first condition of $w(i)$ indicate the center of the image and the second condition applies a factor of 20 to those pixels that are on the edge of the image. Fig. 5(a) shows the anomaly maps without weighting the edge and an anomaly in the center appears. By using the weighted MSE of the formula 4 the anomaly disappears in fig. 5(b).

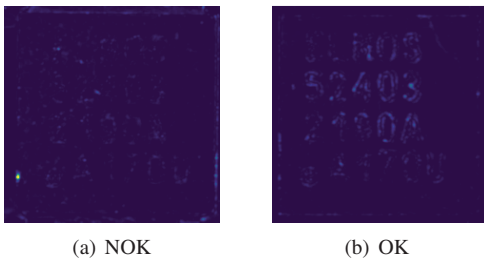


Fig. 4. Difference Map with MSE: $0.6 \cdot 10^{-3}$

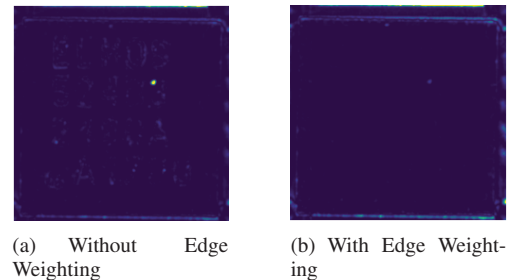


Fig. 5. Illustration the effect of Local Anomaly Weighting

C. Z-Score Method as the Decision Rule

Detecting anomalies is an essential task in several fields, including pattern recognition, computer vision, and image processing. The z-score technique is a commonly used method for anomaly detection. It postulates that anomaly score values are normally distributed with unknown parameters, i.e., mean and variance. According to this method, anomaly score values that contain anomalies come from a distinct distribution and are significantly different from the average anomaly score values of all other images. The z-score can be obtained from the anomaly score values by computing the difference between the mean and dividing it by the standard deviation [11].

However, the z-score has the disadvantage of being a parametric statistical method that relies on assumptions. It assumes a normal distribution, so a modified version of the z-score should be used if the training dataset is not normally distributed. The modified z-score replaces the mean with the median and is calculated using the formula:

$$z_{mod} = \frac{AS - AS_{med}}{c \cdot MAD}$$

where AS_{med} is the median of the anomaly score values from the training dataset, MAD is the median of the absolute deviations from the median, and c is a correction factor. The constant $c = 1.4826$ is required if MAD is used as the standard deviation of a distribution, as it corresponds to the standard deviation of a normal distribution [12].

A threshold can be set using z-scores, where any z-score above or below this threshold is considered an outlier and corresponds to an anomaly semiconductor device. The rule is that all values that are three standard deviations away from the mean should be marked as outliers [11].

V. RESULTS

To evaluate the developed AI-based system and the tape machine, a test dataset is first constructed. To obtain a representative sample of the available data through the tape machine, the sample size of the test dataset \mathcal{T} is set to 1000. This is to ensure that there are enough components with anomalies present in the evaluation. This dataset is assessed by experts at Elmos Semiconductor SE to evaluate the presence of anomalies for the purpose of this work. The test dataset \mathcal{T}_E consists of 77 images of ICs with anomalies present and 923 images without indications of anomalies. The index E is used to identify \mathcal{T} as having classifications made by experts at Elmos Semiconductor SE.

After constructing a test dataset, the next step is to select an appropriate metric to evaluate the performance of the developed AI-based system and the tape machine. *Precision* and *Recall* are two important metrics used in machine learning to evaluate the performance of models in detecting anomalies. Precision measures the proportion of predicted positive cases that are actually positive, while Recall measures the proportion of actual positive cases that are correctly identified by the model. The F_β -Score is a

metric that combines these two metrics into a single value. It is calculated as the harmonic mean of Precision and Recall with the equation

$$F_\beta = \frac{1 + \beta^2}{\frac{\beta^2}{Re} + \frac{1}{Pr}} = \frac{(\beta^2 + 1) \cdot TP}{(\beta^2 + 1) \cdot TP + \beta^2 \cdot FN + FP}, \quad (5)$$

where β^2 is a weighting parameter that determines the relative importance of Precision and Recall. A value of $\beta = 1$ indicates equal weighting, while a higher value of β places more emphasis on Recall and a lower value places more emphasis on Precision. The F_β -Score ranges from 0 to 1, with higher values indicating better performance of the model in detecting positive cases while minimizing false positives. The semiconductor industry follows the Zero-Defect strategy that aims to deliver only defect-free ICs to customers. Therefore the Recall will be significantly weighted compared to Precision with $\beta = 10$.

A. Evaluating a Tape Machine

The performance of the tape machine's anomaly detection on dataset T was evaluated using a confusion matrix, as shown in table I. Out of the 1000 images in the T , the tape machine correctly classified 969 images, resulting in a true positive (TP) count of 896 and a true negative (TN) count of 73. However, the tape machine misclassified 27 images as having an anomaly on the ICs, which were actually not present. Furthermore, for 4 images where an anomaly was present on the corresponding ICs, the tape machine failed to detect the anomaly, resulting in false negatives (FN).

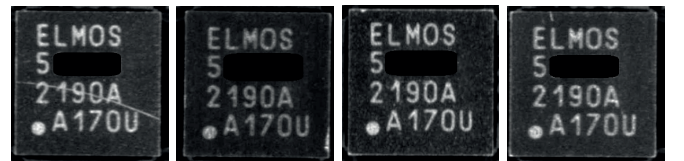
Predicted Class	True Class		Σ
	Anomaly	No Anomaly	
Anomaly	73	27	100
No Anomaly	4	896	900
Σ	77	923	1000

TABLE I

CONFUSION MATRIX OF THE TAPE MACHINE.

To better understand the performance of the tape machine's anomaly detection, the 4 images where the tape machine failed to detect the anomalies is presented in fig. 6. These images highlight the type of defects that were missed by the tape machine's detection system.

The F_{10} -Score for the tape machine is 0.94526. This value serves as the benchmark for the VAE-based system.



(a) Scratch (b) Bottom Right (c) Top Left (d) Top Middle

Fig. 6. Four Anomalous IC Surface Images Missed by Tape Machine Anomaly Detection

B. Evaluating the VAE-Based System

The modified z-score method was implemented in a VAE-based system to detect anomalies in ICs and the system's performance results are summarized in table II. The model predicted anomalies in 112 ICs, but upon further analysis, it was discovered that 35 of these ICs were anomaly-free. However, the model accurately detected all ICs that had actual anomalies, demonstrating the effectiveness of the modified z-score method in detecting anomalies in ICs.

To evaluate the improvement in performance, the F_{10} -Score was calculated, which considers both precision and recall. The obtained F_{10} -Score was 0.99552. This represents a significant improvement over the tape machine, with a 5.32% increase in the F_{10} -Score.

Predicted Class	True Class		Σ
	Anomaly	No Anomaly	
Anomaly	77	35	112
No Anomaly	0	888	888
Σ	77	923	1000

TABLE II

CONFUSION MATRIX OF THE VAE WITH THE MODIFIED Z-SCORE METHOD.

In order to illustrate the detection accuracy, a difference map was generated to visualize the anomalies that were detected by the modified z-score method but not detected by the tape machine. Fig. 7 shows the difference maps of the 4 ICs that were missed by the tape machine.

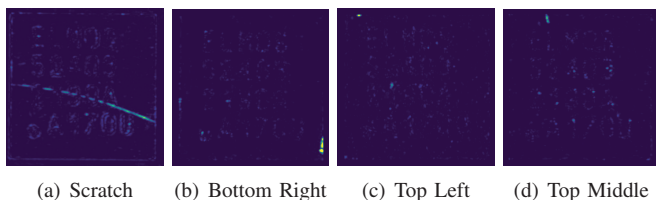


Fig. 7. Anomaly Localization in IC Surfaces: Difference Map of The 4 Undetected Anomalies by Tape Machine

VI. CONCLUSIONS

The manufacturing process of integrated circuits (ICs) is complex and susceptible to errors. Elmos Semiconductor SE follows a Zero-Defect strategy to ensure that only fully functional ICs are delivered to customers. A visual inspection is conducted before packaging and shipping, where a tape machine captures images and performs an integrated automated image processing system to check for any defects that may impair the functionality of the ICs. However, the machine is not perfect and some defects may go unnoticed. The aim of this study is to develop an AI-based system for an automated visual inspection of the ICs using the Variational Autoencoder (VAE) from unsupervised learning. The VAE is trained on images captured by the tape machine that are considered defect-free. Using the VAE, the differences between the reconstructed images and the actual images are

compared and a score is developed to identify any anomalies. This score is designed to meet the requirements of the semiconductor industry and can detect small defects that may affect the functionality of the ICs. The evaluation of the AI-based system shows that it can identify anomalies that were not detected by the tape machine in the initial inspection. The developed AI-based system can serve as an additional quality control measure and helps to improve the Zero-Defect strategy.

REFERENCES

- [1] wko. (2022). Die Halbleiter-Lieferkette: Bedeutung für die Wirtschaft. Global Situation Reports. Retrieved from <https://www.wko.at/site/global-situation-report/halbleiter-lieferkette-1-3-2022.html>
- [2] Minkov, E., Wang, R., Tomasic, A., & Cohen, W. (2006). NER systems that suit user's preferences: adjusting the recall-precision trade-off for entity extraction. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers (pp. 93-96).
- [3] Nishi, Y., & Doering, R. (2017). Handbook of Semiconductor Manufacturing Technology. CRC Press.
- [4] Bank, D., Koenigstein, N., & Giryas, R. (2020). Autoencoders. arXiv preprint arXiv:2003.05991. Retrieved January 15, 2023, from <https://arxiv.org/abs/2003.05991>
- [5] Ackley, D. H., Hinton, G., & Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. Cognitive science, 9(1), 147-169.
- [6] Swamynathan, M. (2019). Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python. Apress.
- [7] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In International Conference on Learning Representations.
- [8] Tao, X., Gong, X., Zhang, X., Yan, S., & Adak, C. (2022). Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey. IEEE Transactions on Instrumentation and Measurement, 71, 1-21. <https://doi.org/10.1109/tim.2022.3196436>
- [9] Mou, L., & Jin, Z. (2018). Tree-Based Convolutional Neural Networks: Principles and Applications. Springer Singapore. SpringerBriefs in Computer Science, 10-12.
- [10] Kingma, D. & Welling, M. (2019). An Introduction to Variational Autoencoders. Foundations and Trends® in Machine Learning, 12(4), 307-392. <https://doi.org/10.1561/22000000056>
- [11] Barnett, V., & Lewis, T. (1994). Outliers in Statistical Data. Wiley.
- [12] Rousseeuw, P., & Croux, C. (1993). Alternatives to the Median Absolute Deviation. Journal of the American Statistical Association, 88(424), 1273-1283. doi: 10.1080/01621459.1993.10476408.

Machine learning for improving the trustworthiness of sensors

Ghazanfer Hussain, Levin G. Thekkumthala, Peter A. William
Faculty IV, ECE
University of Siegen
 Siegen, Germany
 ghazanfer.hussain | levin.thekkumthala | peter.william
 @student.uni-siegen.de

Michael G. Wahl, IEEE Senior Member
Faculty IV, ECE | Electronic Systems and Automation
University of Siegen | Digital University Kerala
 Siegen, Germany | Thiruvananthapuram, India
 michael.wahl@uni-siegen.de

Abstract—Artificial Intelligence / Machine Learning (AI/ML) is a technique with two well-known applications: image analysis and text analysis. In this paper, we will describe an application of AI/ML to a simple 6D acceleration sensor, illustrating the challenges of technical applications and sketching the path for using AI/ML in this environment, where the target is to identify faulty patterns in the data set.

Index Terms—AI, Machine Learning, Sensor data processing, small scale architectures.

I. INTRODUCTION

The challenge which we are facing is the inherent limited reliability of technical systems. A 'convenient' fault which can occur is a catastrophic failure. Here, the faulty behavior becomes visible immediately. For increasing the reliability, it is also necessary to trace variations in the acquired data, which means to detect a change in the data pattern. This is a precondition for predicting failures, where the predictions can be used to optimize the maintenance schedule.

Artificial Intelligence is, among other areas, well known for image analysis and text mining. Both domains have a common property: the data which form the learning basis are well known and available in large quantities. Everybody knows what an image is, and the de-facto standard for images is JPG. In addition, the results can be easily verified. For text mining, the language differs, but the syntax is clear, whereas the semantics is somehow vague. Nevertheless, tools like ChatGPT show clearly that AI/ML is a powerful tool.

In the technical domain, the situation is quite different. The first challenge is the uniformity of data, which is not guaranteed in data from technical devices such as sensors. Technical data can have outliers which are generated by unknown effects and there can be missing data, with both effects disturbing the learning/analysis process in the AI/ML, and the amount of data per time stamp is quite small. A second challenge is the kind of data which differs from sensor to sensor. There are fast sensors with a high data rate, e. g. acceleration sensors, whereas temperature sensors have possibly the same data rate, but a much smaller change rate. Due to the high variability and the small user community, the path to a generic AI/ML development environment is difficult.

In addition, the hardware used for the interpretation of the data is in a quite different range: on the one end huge data centers, on the other end small processors with minimal resources. This difference leads to the necessity to explore the space of small scale AI/ML. There exist many different sensors and small scale applications, but all these applications are very specific. In this paper, we will have a look at different aspects which need to be taken into account when designing a technical AI/ML system, focusing on a specific 6D sensor with the intention to pave the road towards a more global concept.

- Data pre-processing to transform the raw data into a form suitable for AI/ML algorithms
- Algorithms which are suitable for the analysis of technical data with the intention of
- Hardware architectures which can be implemented as part of the sensor system

II. AI/ML DEVELOPMENT

For many common cases in AI/ML development, standardised data sets are available which are shared in the scientific community. This does not hold for sensor data, because there is a multitude of manufacturers with a multitude of different sensors. In addition, the community working on a specific sensor is quite small. For being able to perform a detailed study on sensor data, we had to acquire data on our own account, in spite of a close cooperation with a large sensor manufacturer. Obviously, all sensors that are being manufactured, seem to work perfectly until the end of life of operation. Therefore, we decided to acquire data on our own account, which also solved the potential problems of using protected data.

A. Hardware setup and data acquisition

The environment for acquiring the data was deliberately kept quite simple. We used the Grove 6D accelerometer and gyroscope which is equipped with a sensor ST LSM6DS3, see figure 2. This sensor was connected to an Arduino via I²C which had the task of acquiring the sensor data and sending them with the timestamp via USB to a laptop as an Excel/CSV format. There, the data could be displayed and stored, as shown in figure 1.



Fig. 1: Concept of the data acquisition path

For getting different types of data, we acquired data in the following scenarios: motorway, road, and city. The data from the motorway obviously showed little variations, whereas the data acquired in the city were much diverse due to the frequent breaking and accelerating.

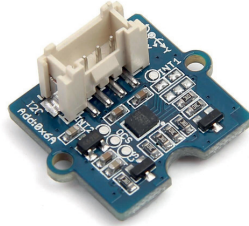


Fig. 2: Data acquisition and sensor module

Besides the three values for each acceleration and gyro rate, the sensor also provided temperature values as Fahrenheit or Celsius.

B. Data Pre-processing

Pre-processing is essential because technical data are usually not well formed. Examples are inadequate data, where data points are missing, bias and human mistake, manual labelling and annotation of data sets, and finally privacy and compliance: the majority of potential data providers do not divulge their data due to national security, personality protection (medicine...), and because non-perfect data are treated as company secret.

1) *Data Quality*: Almost any data set has a variety of anomalies and underlying issues: *Mismatched data types*: Data can have different forms and ranges when getting information from different sources, *mixed data values*: different sources use different feature names making it difficult to combine them, *Data outliers*: Outliers are basically values which are out of the possible physical range in the data set, *Missing Values*: A value in a time series is e.g. zero. Because AI algorithms cannot deal with this, missing values need to be either filled, e.g. with the average of the neighborhood values or the overall average value, or the complete entry will be discarded.

2) *Data Reduction*: Excessive data resulting from characterising phenomena in multiple ways or data unrelated to a specific ML, AI, or analytic goal are common. This reduces the amount of data stored while both simplifying and improving the accuracy of the analysis.

- Attribute selection helps to fit your data into smaller pools. It basically merges characteristics or tags.
- Dimension Reduction means using less data to support analysis and other processes that come afterward. Pattern recognition is used by algorithms such as K-nearest

neighbours to group together similar data and make it easier to handle.

3) *Data conversion*: Here, data will be converted into the format(s) required for analysis and other steps that come after.

Normalisation: To enable more reliable comparisons, normalisation scales your data into a regularised range. For example, you will need to scale the results within a given range, such as -1.0 to 1.0 or 0.0 to 1.0.

Feature selection: The process of determining which variables – features, characteristics, etc. – are most crucial to your analysis. Using more features will lengthen the training process and occasionally result in less accurate outcomes.

Data Validation: The data has now been divided into two sets. A model is trained using the first set of data. The testing data, which is the second set, is used to assess the final model's resilience and correctness. This second phase aids in locating any issues with the hypothesis that was applied to the data's cleansing and feature engineering.

C. Data Analysis

A summary of the acquired sensor data is shown in figure 3. Obviously, the thermometer values (Fahrenheit and Celsius) values do not change significantly over the measurement period, which reflects the short acquisition time.

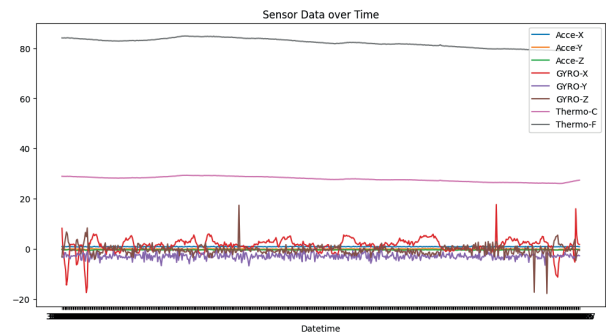


Fig. 3: Sensor data analysis of all coordinates

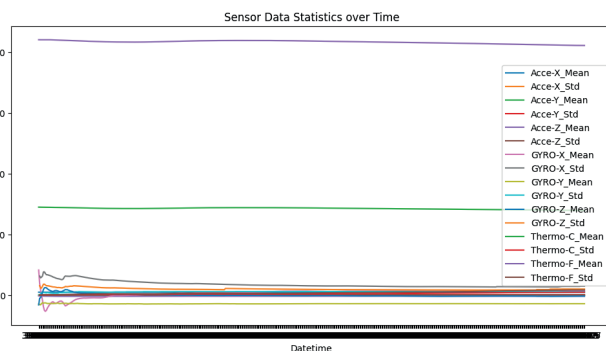


Fig. 4: Mean value and standard deviation over time

The acceleration (AcceX) and gyroscope (GyroX) values obviously show many changes throughout the experiment. Figure 4 shows the sensor data analysis of all coordinates

with mean value and standard deviation. This information is essential for the improvement of the data quality.

The numeric values of mean and standard derivation are shown in figure 5.

Mean values:		Standard Deviations:	
Acce-X	0.934216	Acce-X	0.069396
Acce-Y	-0.036533	Acce-Y	0.091258
Acce-Z	-0.348697	Acce-Z	0.060665
GYRO-X	1.425976	GYRO-X	2.939549
GYRO-Y	-2.822195	GYRO-Y	1.104726
GYRO-Z	-0.319268	GYRO-Z	1.999299
Thermo-C	27.890439	Thermo-C	0.952483
Thermo-F	82.201463	Thermo-F	1.713087
dtype: float64		dtype: float64	

Fig. 5: Numeric mean and standard derivation values

Figure 6 shows a list of outliers which were identified more or less manually. These were removed from the data sets, because they were obviously wrong. Luckily, the number of outliers was not so big that they change the overall picture of the data. At the end of this process, we had now two data sets: the original acquired values and the values without outliers.

Outliers:								
	date	timeStamp	Acce-X	Acce-Y	Acce-Z	GYRO-X	GYRO-Y	GYRO-Z
19	02/06/2023	06:17.9	0.7579	-0.0439	-0.5114	-1.75	-1.19	0.28
196	02/06/2023	09:35.2	0.7754	-0.0615	-0.2406	0.98	-2.73	0.21
303	02/06/2023	11:34.4	0.6847	0.0766	-0.5729	-8.54	-1.47	5.88
311	02/06/2023	11:43.3	0.5993	0.0996	-0.2948	5.74	-0.42	-2.59
316	02/06/2023	11:48.9	0.7491	-0.0576	-0.4299	-0.49	1.54	2.94
320	02/06/2023	11:53.4	0.6910	-0.1064	-0.3060	-2.52	-11.83	3.50
327	02/06/2023	12:01.2	0.7759	0.0537	-0.3070	6.79	0.07	-0.98
332	02/06/2023	12:06.7	1.1419	-0.1366	-0.0703	8.40	-2.52	-2.87
341	02/06/2023	12:16.8	1.1122	0.1479	-0.0561	26.32	-6.09	-8.75
354	02/06/2023	12:31.3	1.1224	-0.0434	-0.2416	3.43	-1.12	0.00
427	02/06/2023	13:52.6	0.6500	0.0229	-0.4055	-0.28	0.14	-5.88
428	02/06/2023	13:53.7	0.7281	-0.1527	-0.5929	1.26	-1.61	-3.22
429	02/06/2023	13:54.9	0.7549	-0.1864	-0.3011	1.61	-4.62	3.50
444	02/06/2023	14:11.6	0.7452	-0.0059	-0.6080	2.03	-8.26	-3.08
463	02/06/2023	14:32.7	1.0995	-0.1474	-0.4002	1.47	-0.21	0.56
464	02/06/2023	14:33.8	1.0887	-0.0986	-0.1708	2.59	-1.26	-17.71
471	02/06/2023	14:41.7	0.7315	-0.0454	-0.4685	-0.84	-10.22	-2.66
473	02/06/2023	14:43.9	0.7627	-0.1913	-0.3196	-3.99	4.06	-6.58
504	02/06/2023	15:18.4	0.7706	-0.0210	-0.5397	2.17	0.14	-3.15
534	02/06/2023	15:51.9	1.2834	0.0078	-0.3216	3.78	-5.25	-1.75
549	02/06/2023	16:08.6	1.1043	-0.0181	-0.2181	3.43	-2.59	3.36
550	02/06/2023	16:09.7	1.0882	-0.0303	-0.5134	1.05	-4.34	-2.31
552	02/06/2023	16:11.9	0.7676	-0.0361	-0.2001	2.17	-10.01	2.24
568	02/06/2023	16:29.7	1.0848	-0.2396	-0.1025	0.07	0.42	-2.52
708	02/06/2023	19:05.8	1.1014	-0.0956	-0.3699	2.03	-4.62	-3.50
718	02/06/2023	19:16.9	0.7662	-0.0830	-0.3655	1.05	-1.05	-0.07
721	02/06/2023	19:20.3	1.1527	0.0776	-0.4168	2.31	8.89	3.92
762	02/06/2023	20:06.0	0.7662	0.0498	-0.3928	4.55	-1.61	-5.25
845	02/06/2023	21:38.4	1.0843	-0.1791	-0.1698	8.33	-2.80	-0.28
865	02/06/2023	22:00.7	1.0878	0.0039	-0.2474	10.29	-5.32	-0.07
868	02/06/2023	22:04.1	0.7696	0.1728	-0.2045	-2.59	-2.52	3.15

Fig. 6: Outlier values of each coordinates of 6d sensor

These data sets without outliers were used later on in the hardware analysis.

D. AI/ML algorithms

Sensor Fusion algorithms are used to combine data sets from accelerometer and gyroscope for better orientation. Madwick filter can be more computationally efficient variant, while Mahony filter further reduces computational cost. Power

consumption is less for this filter than for Kalman filters. Kalman filters use the data of the Gyroscope to orient data of the accelerometer and hence more accuracy and robustness.

Time Series compression These algorithm can preserve the information which are essential, but also it will reduce the amount of data. Gyroscope and accelerometer data are usually collected as time series, where each data point represents a measurement of angular velocity (in the case of gyroscope) or acceleration (in the case of accelerometer) at a specific time. We can apply *Piecewise Aggregate Approximation* (PAA) [10]) to gyroscope and accelerometer data in a similar manner as it is applied to time series data in general. First, the time series are segmented into small intervals, where the mean value, the median, or another aggregate function is applied to the data points in this segment. Then the original data points can be replaced by the representative values from the aggregate function. By this process, we can effectively reduce the dimensionality of the accelerometer and gyroscope data while preserving essential features and trends. This is beneficial for tasks such as activity recognition and motion analysis. PAA also can help in reducing computational complexity and memory usage when dealt with large volumes of sensor data collected over extended periods.

Incorporating PAA into the sensor fusion algorithm can effectively reduce the computational complexity of processing time series data from multiple sensors while retaining important information for fusion. This in turn will get us to more efficient and robust fusion system. It will improve decision-making in various real-world applications such as robotics, autonomous vehicles, wearable devices, healthcare monitoring, and industrial automation. Reducing the data sets using the approaches described above is a significant step towards small scale AI/ML which is intended to run on the hardware as lined out in the next section.

III. HARDWARE ARCHITECTURES

In the previous sections of the paper, we have discussed the aspects of sensor data acquisition, pre-processing and analysis. For this part, we had ample resources at hand. Bringing AI/ML to the sensor means that the hardware available for processing the data is rather limited.

A. State of the art

One approach are Quantum Dot Cellular Automata (QCA) [5] which are a potential solution for building energy-efficient and rapid accelerators needed for AI applications like computer vision and robotics. The traditional CMOS technology has drawbacks [1], such as severe process variation, GHz frequency limit, increased leakages, reduced control over the gate, and high power densities. QCA technology employs electrostatic force and Coulombic repulsion force to propagate signals between electrons and adjacent cells, resulting in QCA wires, inverters, and majority gates. QCA clocking uses reversible logic and four phases to conserve energy and maintain cell logic, enabling information flow in the forward

direction. The USE clocking scheme is a set of design standards that ensure simplicity in manufacturing, scalability, and adaptable routing. The Nand-Nor-Inverter (NNI) universal gate are introduced which adheres to the USE clocking scheme and has a smaller footprint to enable NAND and NOR operations. QCA Designer 2.0.3 was used to develop the QCA design simulation, and the MAC unit introduces an 8-bit PIPO register using D flip-flops and presents two designs. In further studies, also other hardware implementations need to be analysed.

Other approaches to achieve smaller hardware are approximate computing [4], which meets quite well the requirements of physicals. Memristors are new hardware components which can be used in NN implementations for low-energy MAC calculations [1]. In the approach Processing in the Emerging Memory (PIEM) [2], a MAC operation is the output current of the sum of the memristor's conductance and the reference voltage. AdderNet [7] is as an alternative which replaces the original convolution with adder kernel-only additions, resulting in lower energy consumption. There are other components that can be used in neural networks, such as the analog memristor network and the XNOR logic operation kernel. The AdderNet kernel uses bit-level operations and analog circuits, resulting in lower power consumption. The LeNet-5 on Zynq-7020, where all computation and weights are stored on board, resulted in a significant drop in energy consumption and logic resource utilization. Vector symbolic architecture (VSA) [3] is a computational approach inspired by the brain that can operate on both symbolic and numeric methods. VSA uses three operations: bundling, binding, and permuting to perform computations. Other approaches are followed in medical applications [8].

B. Methodology

The layer denotes the total number of hidden layers the neural network will have during training, and each layer comprises of several neurons. Depending on the type of network structure required, different numbers of layers and neurons might be included. The following describes the way how layers and neurons function.

1) *Neuron*: Figure 7 shows the working principle of a neuron and layer structure and how it is connected to the next layer. Equation 1 illustrates how a neural network operates, and here w stands for the weight and x for the network's input. The weights are stored in weight memory and are initialised via a ROM. The input value x is multiplied by the corresponding weight value w . This result is added to the bias. The operation takes place in a multiplier/accumulator (MAC). The result is saved in the variable sum . This process is repeated until all inputs have been multiplied by weight. The activation function needs to be applied to the final value following the multiplication of the weight values by the input values and the added bias value. Without the activation function, the network would be a collection of linear layers layered on top of one another, making it impossible for it to learn from the variety of input data. It is crucial to keep in mind that weights are shared on the common bus, while numbers are allocated to

distinguish between layers and neurons. For instance, if the layer contains 50 neurons, it will also have a weight of 50 and bias values associated with it.

$$y = wx + b \quad (1)$$

2) *Layer*: In fully connected network requires the connection of every neuron from the previous layer. Data from each layer is initially kept in the shift register between layers, and it is transferred to the following layer once every clock cycle, as shown in graphic 7(b).

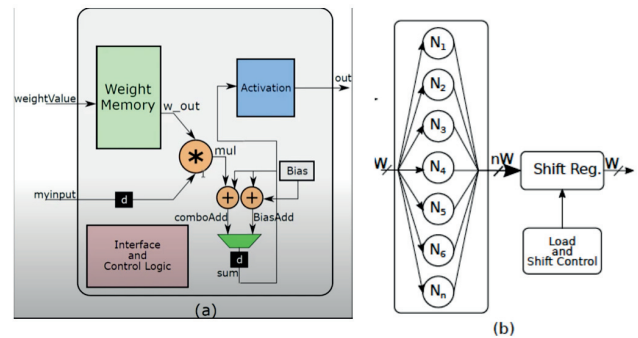


Fig. 7: (a) Neuron Architecture, (b) Layer Architecture [6]

C. Implementation of neural networks

To study the hardware requirements for small scale AI/ML, a well known data set was used. This removed the dependency of our work from the availability of correct and faulty sensor data. The MNIST [11] data set, a widely used data set for machine learning tasks, is composed of 10,000 test images and 60,000 handwritten images for training, and every digit from 0 to 9 is saved as a 28 by 28 pixel greyscale image. Considering all of the sensor inaccuracy, the MNIST data set is rather clean in comparison to the 6D sensor data; however, data cleaning is still necessary before it can be utilised for machine learning tasks.

1) *TensorFlow Neural Network Implementation*: Figure 8 follows a tensorflow-based implementation of a deep neural network; the network's prerequisite libraries are included in its initial few lines. After that, it downloads the collection of 60,000 handwritten MNIST digit images and separates them into training sets and test pictures. This separation of the training and testing sets ensures that the network generalises to unseen data sets, as it would otherwise overfit to the training data set. The network itself has six layers, five of which are hidden layers and made up of 50, 50, 50, 50, and 10 neurons with sigmoid activation. The last layer consists of 10 neurons that suggest which neuron is giving the maximum value. The model is trained on 25 epochs with Adam optimizer and sparse categorical cross-entropy loss. The final couple of lines of the script gather the weight and bias values from the second layer to the last layer — not from the first layer, because the first layer is initially a fully connected layer of the data set

— and stores them in a text file. The weight and bias files are obtained using built-in methods in the tensorflow library, while it is probable that there are further methods [9] to get weight and bias values.

```
import tensorflow as tf
import json
import time
mnist = tf.keras.datasets.mnist
(x_train,y_train),(x_test,y_test) = mnist.load_data()
x_train=tf.keras.utils.normalize(x_train,axis=1)
x_test =tf.keras.utils.normalize(x_test,axis=1)

mdl = tf.keras.models.Sequential()

mdl.add(tf.keras.layers.Flatten())
mdl.add(tf.keras.layers.Dense(50,activation=tf.nn.sigmoid))
mdl.add(tf.keras.layers.Dense(50,activation=tf.nn.sigmoid))
mdl.add(tf.keras.layers.Dense(50,activation=tf.nn.sigmoid))
mdl.add(tf.keras.layers.Dense(50,activation=tf.nn.sigmoid))
mdl.add(tf.keras.layers.Dense(10,activation=tf.nn.sigmoid))
mdl.add(tf.keras.layers.Dense(10,activation=tf.nn.sigmoid))
mdl.compile(optimizer='adam',
loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
start_time = time.time()
mdl.fit(x_train,y_train,epochs=25)
(val_loss,val_accuracy) = mdl.evaluate(x_test,y_test)
end_time = time.time()

inference_time = end_time - start_time
print(f"Inference time: {inference_time} seconds")

num_params=mdl.count_params()
print(f"Number of parameters: {num_params}")
weightLst =[]
biasLst =[]

for i in range(1,len(mdl.layers)):
    weights = mdl.layers[i].get_weights()[0]
    weightLst.append((weights.T).tolist())
    bias = [[float(b)] for b in mdl.layers[i].get_weights()[1]]
    biasLst.append(bias)
data ={"weights":weightLst,"biases":biasLst}
f = open('weightsandbiases.txt','w')
json.dump(data,f)
f.close()
```

Fig. 8: Tensorflow neural network

2) *The Zynet [6] Neural Network Implementation:* The tensorflow-based implementation figure 8 of neural networks and the Zynet implementation figure 9 of deep neural networks are comparable. Installing a necessary Zynet module is the first step, followed by defining the layer structure. The network comprises six layers; the first five hidden layers are made up of 50, 50, 50, 50, and 10 neurons with sigmoid activation function, while the final layer is made up of 10 neurons with hardmax activation. The hardmax activation function evaluates the output of each neuron and determines which neuron is providing the highest value. Since tensorflow-based and Zynet are completely linked networks, every neuron in the previous layer is entirely connected to the subsequent layer. For hardware implementation, we needed the weight and bias values we obtained from the tensorflow implementation, hence we now use the appropriate .txt file. We provide the file as input to the generateArray method for weight and bias values required for the hardware. Every layer, along with the number of bias values and neuron weights, is saved in a memory initialization file.

Zynet specifies the number of bits used for the integer part

```
from zynet import zynet
from zynet import utils
import numpy as np
def MnistZynet(dataWidth,sigmoidSize,weightIntSize,inputIntSize):

    mdl = zynet.model()

    mdl.add(zynet.layer("flatten",784))
    mdl.add(zynet.layer("Dense",50,"sigmoid"))
    mdl.add(zynet.layer("Dense",50,"sigmoid"))
    mdl.add(zynet.layer("Dense",50,"sigmoid"))
    mdl.add(zynet.layer("Dense",50,"sigmoid"))
    mdl.add(zynet.layer("Dense",10,"sigmoid"))
    mdl.add(zynet.layer("Dense",10,"hardmax"))

    weightArray =utils.genWeightArray('WeightsAndBiases.txt')
    biasArray = utils.genBiasArray('WeightsAndBiases.txt')

    mdl.compile(pretrained='Yes',
weights =weightArray,
biases=biasArray,
dataWidth=dataWidth,
weightIntSize=weightIntSize,
inputIntSize=inputIntSize,
sigmoidSize=sigmoidSize
)

    zynet.makeXilinxProject('ai','xc7z020c1g400-1')
    zynet.makeIP('ai')
    zynet.makeSystem('ai','block_design')
if __name__=="__main__":
    MnistZynet(dataWidth=8,sigmoidSize=10,weightIntSize=3,inputIntSize=1)
```

Fig. 9: Zynet Neural network

of input and weight values using fixed point representation, inputIntSize, and weightIntSize. WeightIntSize specifies bits as the integer portion; the other bits are fractional portions. The depth of the activation function, which corresponds to the sigmoid size, is implemented as the Look-up table (LUT) specified depth 10, which is 1024 values, and when depth is 5, values are 32 (e.g. $2^{\text{address-bits}}$). The zynet.model() method creates a deep neural network object, and model.add() adds a new layer to the network. The method zynet.makeXilinxProject() generates Xilinx project with deep neural network (DNN) as top module. It takes two parameters; first parameter is project name, second is FPGA board number which you need to program, and zynet.makeIP() package the DNN in IP-XACT format. The zynet.makeSystem() makes block design for a specified IP block, IO peripherals, AXI interface, Zynq processor system, and DMA controller. When we execute neural networks using the given parameter, the hardware footprint is depicted in figure 11. Following the execution of the Zynet implementation figure 9, a project with the name "ai" will be created in Vivado, the Xilinx synthesis tool. The script assumes that Vivado is present in the computer environment variable because it would otherwise produce an error message. The number of layers in software implementation matching hardware implementation is another crucial factor. If the count of each layer varies between the hardware and software layers, the output will be flawed because it will not have weight and bias values for that particular layer. The axi-stream interface facilitates communication between various system components, such as the zynq processing, peripheral, memory while the axi-lite interface helps in data set initialization. After the data set was cleaned, we created hardware designs using the same procedure section III-C; the hardware design is shown in the figures 12 and 13.

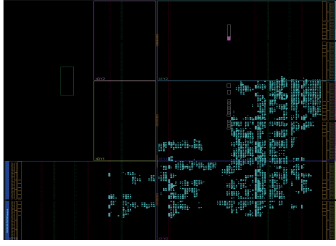


Fig. 10: Footprint for MNIST neural network 0

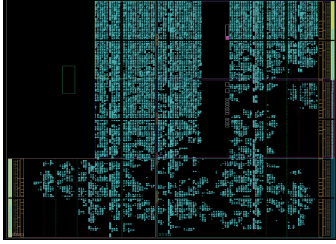


Fig. 11: Footprint for MNIST neural network 3

Table I provides an overview for different implementations on the Xilinx FPGA Zynq-7000 for different data sets from the reference as well as our own sensor data. The size of the data set that is provided is directly correlated with the number of parameters. The MNIST data set consists of 60,000 training samples, and the sensor data set only has 1400 data points. The NN 3, the largest network with a size of 47520 parameters, consumes power 193 while utilizing 24711 Look-Up Table (LUT), 12727 Flip-Flops (FF), 77 Block RAM (BRAM), and 1 global clock buffer (BUFG). NN 0 with sensor data has 581 parameters, and consumes power only 35, using 2379 LUT

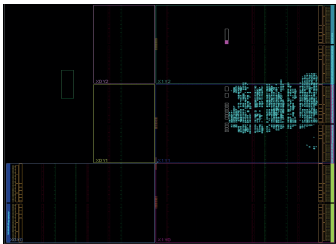


Fig. 12: Footprint for 6D Sensor neural network 0

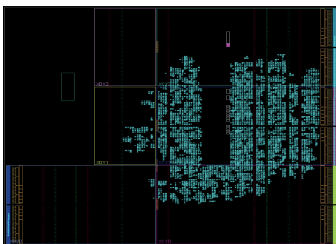


Fig. 13: Footprint for 6D Sensor neural network 1

and 1631 FF.

Arch.	Data	Param.	Power	Resources
NN 0	MNIST	16020	60	LUT 6111, FF 3275, BRAM 22.50, IO 97, BUFG 1
NN 1	MNIST	24490	79	LUT 7186, FF 3799, BRAM 30, IO 97, BUFG 1
NN 2	MNIST	33460	107	LUT 11938, FF 6079, BRAM 44.5, IO 97, BUFG 1
NN 3	MNIST	47520	193	LUT 24711, FF 12727, BRAM 77.0, IO 97, BUFG 1
NN 0	Sensor	581	35	LUT 2379, FF 1631, IO 97, BUFG 1
NN 1	Sensor	4461	106	LUT 10333, FF 6526, IO 97, BUFG 1

TABLE I: Resources utilization of different dataset

The result of this task shows the path towards an implementation of small scale AI/ML on an FPGA, a precondition for a future implementation on silicon.

IV. SUMMARY AND OUTLOOK

In this paper we described the different preconditions and tools which are necessary to approach a sensor-based AI/ML. Nevertheless, the key problem is not yet solved: the availability of data of correct and faulty sensors, even whilst cooperating with industry. Thus, a project proposal was submitted to XECS, where the generation of data is one of the key issues.

REFERENCES

- [1] I. Hussain, A. Singh, S. Caudhury: "A review on the effects of technology on CMOS and CPL logic style on performance, speed and power dissipation". In 2018 IEEE Electron Devices Kolkata Conference (EDKCON), pages 332–336. IEEE, 2018.
- [2] A. P. James: "A hybrid memristor–cmos chip for AI". *Nature Electronics*, 2(7):268–269, 2019.
- [3] D. Kleyko, M. Davies, E. P. Frady, P. Kanerva, S. J. Kent, B. A. Olshausen, E. Osipov, J. M. Rabaey, D. A. Rachkovskij, A. Rahimi, et al.: "Vector symbolic architectures as a computing framework for nanoscale hardware". arXiv preprint arXiv:2106.05268, 2021.
- [4] V. Leon, K. Pekmestzi, D. Soudris: "Exploiting the potential of approximate arithmetic in DSP and AI hardware accelerators". In 2021 31st International Conference on Field-Programmable Logic and Applications (FPL), pages 263–264. IEEE, 2021.
- [5] A. Mamdouh, M. Mjema, G. Yemiscioglu, S. Kondo, A. Muhtaroglu: "Design of efficient AI accelerator building blocks in quantum-dot cellular automata (qca)". *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(3):703–712, 2022.
- [6] K. Vipin: "Zynet: Automating deep neural network implementation on low-cost reconfigurable edge computing platforms". In 2019 International Conference on Field- Programmable Technology (ICFPT), pages 323–326, 2019.
- [7] Y. Wang, M. Huang, K. Han, H. Chen, W. Zhang, C. Xu, D. Tao: "Addernet and its minimalist hardware design for energy-efficient artificial intelligence". arXiv preprint arXiv:2101.10015, 2021.
- [8] Y. Wei, J. Zhou, Y. Wang, Y. Liu, Q. Liu, J. Luo, C. Wang, F. Ren, L. Huang. "A review of algorithm and hardware design for ai-based biomedical applications". *IEEE transactions on biomedical circuits and systems*, 14(2):145–163, 2020.
- [9] M. Nielsen: "Neural networks and deeplearning on MNIST dataset". <http://neuralnetworksanddeeplearning.com/chap1.html>, 2019
- [10] M. Ohsaki, H. Abe, T. Yamaguchi. "Numerical Time-Series Pattern Extraction Based on Irregular Piecewise Aggregate Approximation and Gradient Specification". *New Gener. Comput.* 25, 213–222 (2007). <https://doi.org/10.1007/s00354-007-0013-9>
- [11] C. J. B. Burges. "THE MNIST DATABASE of handwritten digits". Yann LeCun, Courant Institute, NYU Corinna Cortes, Google Labs, New York

Vision-based Autonomous Trajectory Drifting using Deep Reinforcement Learning

Fabian Domberg^{1*}, Ben Barkow^{1*}, Georg Schilbach¹

¹Institute for Electrical Engineering in Medicine, Universität zu Lübeck, Germany

Abstract—We present a method for training an autonomous vehicle to drift along different trajectories. Unlike previous work, we use only visual input from a front-facing camera. Because it is difficult to train end-to-end policies to achieve such complex behavior with only image input, we train different parts of our neural network separately. We use supervised learning to train a CNN-based feature estimator to predict the vehicle state, which is then fed into a fully connected network trained using Reinforcement Learning. This training method allows for better interpretability and reliable convergence compared to end-to-end approaches that attempt to optimize over the much larger combined search space. While our final trained controller shows some undesirable oscillations during drift and overly cautious driving, it generally proves the feasibility of drifting using only visual inputs.

Keywords—Image-based, Racing, Drifting, Reinforcement Learning, Supervised Learning

I. INTRODUCTION

AUTONOMOUS driving remains a challenging area of robotics and has evolved from a research topic to being developed and implemented in applications including personal mobility, logistical systems, and motorsport. Racing especially has established itself as an interesting research field pushing autonomous vehicles and their underlying algorithms to their limits. These unforgiving environments often force engineers to come up with innovative ideas. This work focuses on one such scenario, which is autonomous drifting. Whilst this topic has been examined in some prior works, they often assume near-perfect state information and accurate, high-frequency measurements. As this may not always be available, we seek to answer the question of whether autonomous drifting is possible given *only* visual inputs from a vehicle-mounted camera. The following chapter will elaborate on previous research in this field and outline this paper’s contributions.

II. RELATED WORK

Previous work in the domain of autonomous drifting often relies on traditional control paradigms to synthesize robust controllers [1, 2]. With recent advances in machine learning, neural networks have emerged as a viable alternative for many, especially difficult, nonlinear control problems. Reinforcement Learning (RL) in particular has proven effective for such complex and difficult-to-model tasks [3]. This is due to its ability to learn from interactions with an actual system, rather than relying on an abstract mathematical model of it. Thus, in the context of vehicle drifting, RL has been used in several recent publications. Cutler et al. [4] use it to refine an initial controller derived from a simplified model in order to improve its real-world performance. They choose this initialization step because the otherwise prohibitively large search space for continuous-control RL. More recently, Cai and Mei et al. [5] used the model-free Soft-Actor-Critic (SAC) RL algorithm to learn a drifting policy from scratch. Changing vehicles and track layouts during training even allowed their controller to

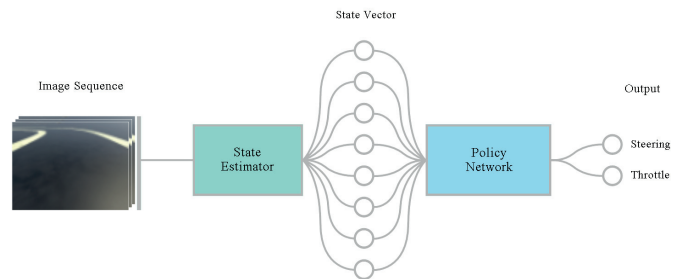


Fig. 1: Our proposed network architecture, allowing for independent training and interchanging of the two sub-networks.

generalize to unseen circumstances during deployment. Building on this, Domberg et al. [6] presented a controller capable of robustly drifting along arbitrary trajectories, validating their findings on a scale RC car. While these works demonstrate the power of learning-based methods, they all rely on accurate, ground-truth measurements about the vehicle’s state and its surroundings. Jaritz et al. [7] instead investigate the feasibility of using only visual inputs from a front-facing camera on the vehicle, and present a learned controller capable of performing in a rally racing scenario. This suggests that highly dynamic vehicle states can be estimated using vision alone.

In this work, we aim to show that the task of autonomous drifting is solvable using only visual inputs. We therefore propose the use of a state estimation network, which precedes the regular decision-making policy network, as shown in Figure 1. This split network design, as opposed to a singular large neural network, offers several advantages. First, following the principle of divide-and-conquer, it allows both the state estimation and policy network to be developed and trained independently, resulting in faster convergence and easier adaptability. Second, it allows either part to be exchanged or modified without changes to the other. This, for example, enables the same state estimation component to be used in conjunction with different policy networks, e.g. for regular driving and drifting. Or, the other way around, enables using the same policy network with different state estimators, e.g. for different visual settings or even zero-shot sim-to-real transfer.

* Equal Contribution

III. NETWORK ARCHITECTURE

The first subcomponent of our architecture is the state estimation network block. It is responsible for converting a sequence of three RGB images captured by the vehicle's front-facing camera into a low-dimensional state vector. The second subcomponent is a Fully Connected Neural Network also referred to as Multi-Layer Perceptron (MLP). It takes in the current state vector and computes the control inputs, i.e. steering and throttle, of the vehicle.

It allows for separate training and fine-tuning of the CNN and MLP while providing insight into the otherwise hidden output layer of the CNN. This has huge advantages in testing the effectiveness and accuracy of the feature extractor.

A. State Estimator Network

The state estimator network is a modified version of the VGG 16 CNN [8]. Since this network was originally designed for a classification task, we slightly modified its architecture. The final structure of the CNN is visualized in Figure 2.

The CNN receives three downsampled RGB images from the front-facing camera, which are stacked into a single 9-channel 80x60 pixel image. The reason for choosing three consecutive frames is so that the network can accurately capture the motion information contained between them. By providing multiple frames, changes in position, scale, or rotation of objects can be mapped to the vehicle's ego-motion.

In general, the number of layers in the network is reduced from the original 16 to 8, to account for the reduced input image size. Also, each sequential layer combination of convolutional and pooling layers only uses a single convolutional layer, as opposed to the original two or three in VGG 16. However, the total number of these layer-sets however is only slightly reduced from 5 to 4. The softmax of the last fully connected layer is replaced by a sigmoid activation to enable numeric outputs instead of probabilities. The dimensionality of the four fully connected layers $fc5$, $fc6$, $fc7$, $fc8$ at the end of the network is gradually reduced so that the final number of outputs is eight, which is the number of state variables to estimate.

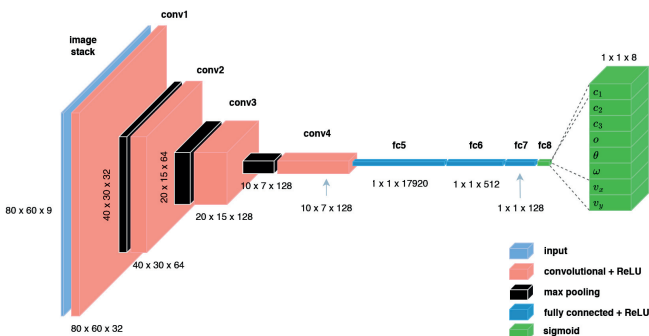


Fig. 2: Architecture of the feature extractor network.

B. State Vector

The state vector is used as the only input to the policy network and thus needs to represent the full vehicle state, as well as information about the road ahead, as precisely as possible while maintaining reasonable dimensionality. Similar to [6], we define the state vector as $F = [c_1, c_2, c_3, o, \theta, \omega, v_{xCar}, v_{yCar}]$. Its individual variables are defined as follows.

Curvatures: c_1, c_2, c_3

Given nine evenly spaced waypoints $W = [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9]$, each 2.5 meters from the previous starting from the vehicle's current position meandering along the desired trajectory, we calculate the three curvatures c_1, c_2, c_3 . Splitting W into $W_1 = [w_1, w_2, w_3]$, $W_2 = [w_4, w_5, w_6]$, and $W_3 = [w_7, w_8, w_9]$, we calculate three distinct curvatures at different distances along the road ahead of the car.

Offset from road center: o

The value of o is calculated as the distance between the position of the vehicle and the nearest point u on the track.

Rotation relative to path tangent: θ

The angle θ is calculated between the vehicle's forward orientation and the tangent of the path at its current position u on the track.

Velocities: v_x, v_y, ω

The translational velocities v_x and v_y transformed to be in the vehicle's local coordinate system, i.e. ego-centric. Similarly, ω is the vehicle's yaw rate around its z axis.

For better convergence and stability in training, we normalize each state value by its theoretical (or, in the case of velocities, empirically determined) maximum value. The setup and kinematic model used to calculate the desired values can be examined in Figure 4.

C. Policy Network

The decision making MLP part of our overall network architecture directly takes in the (estimated) state vector F , as output by the state estimation network. Internally, it is comprised of three hidden layers with 256 nodes each. At the end, the final layer outputs a throttle and steering command. This structure is visualized in Figure 3.

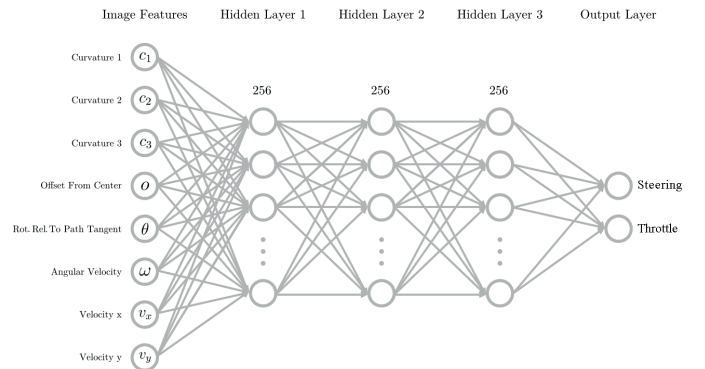


Fig. 3: Visualization of the policy network.

IV. REWARD FUNCTION

In the following, we describe our reward function. For modularity and easier reproducibility, we introduce its different parts one by one, building upon each other.

A. Regular Driving

This part of the reward function is responsible for regular driving, i.e. following the desired trajectory as quickly as possible. It is purely based on the speed of the vehicle and its distance from the center of the path. It is defined as:

$$f_{\text{drive}}(o, v_x, v_y) = f_{\text{offset}}(o) f_{\text{vel}}(v_x, v_y) \quad (1)$$

where

$f_{\text{offset}}(o)$: Reward based on the distance of the vehicle from the path centre.

$f_{\text{vel}}(v_x, v_y)$: Reward based on the vehicle's speed in the direction of the path.

1) *Center Distance Component*: The center distance component is included in the reward function to help the agent learn that driving off the road is not a desired behavior. This guidance is theoretically achieved by penalizing a distance $|o| > o_{\text{max}}$ with a fixed reward of $f_{\text{dist}}(o) = 0$. However, since RL algorithms favor continuous, i.e. differentiable, reward functions, the reward for $|o| < o_{\text{max}}$ is implemented as the following exponentially decaying function:

$$f_{\text{offset}}(o) = 1 - \left(\frac{|o|}{\frac{w_{\text{track}}}{2}} \right)^4 \quad (2)$$

The rapid decrease of the overall reward at the outer boundaries enables the agent to learn to avoid these critical regions before actually driving off the track. This behavior can be altered to occur earlier or later by changing the exponent.

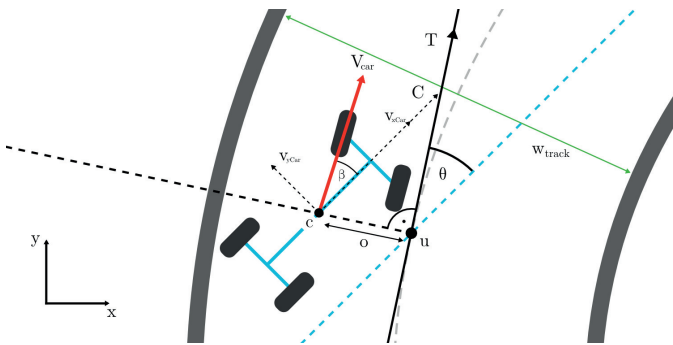


Fig. 4: Kinematic visualization of a car in a curve, currently in a drifting state.

2) *Velocity Component*: Apart from staying on the road, a key concern of the agent should be to move around the racetrack as fast as possible. This could be achieved simply by rewarding higher vehicle velocities. However, to avoid suboptimal behaviors such as driving in circles, we instead use the vehicle's velocity along the current path tangent:

$$\begin{aligned} \vec{T} &= \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \\ \theta_T &= \arctan 2(t_y, t_x), \\ \vec{V}_T &= \begin{bmatrix} \cos(-\theta_T) & -\sin(-\theta_T) \\ \sin(-\theta_T) & \cos(-\theta_T) \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} v_{xT} \\ v_{yT} \end{bmatrix}, \\ f_{\text{speed}}(v_{xT}, v_{yT}) &= \frac{|v_{xT}|}{v_{xT_{\text{max}}}}. \end{aligned} \quad (3)$$

where

\vec{V}_T : Car's velocity relative to tangent T ,

θ_T : Angle of tangent T with global x-axis,

v_{xT} : Velocity component along tangent T ,

v_{yT} : Velocity component perpendicular to T

$v_{xT_{\text{max}}}$: Empirically determined maximum speed of the car.

B. Drifting

To additionally encourage drifting behavior, we add another component to equation (1) and introduce the weighting factor $\lambda \in [0, 1]$. The overall reward function becomes:

$$f_{\text{drift}}(o, v_x, v_y, \beta) = \lambda f_{\text{drive}}(o, v_x, v_y) + (1 - \lambda) f_{\text{slip}}(\beta). \quad (4)$$

The drift reward uses the *slip angle* β as the main measurement for drifting. It represents the angle between the vehicle's forward vector and its current velocity vector. Since during regular driving, slip angles of up to 25° may occur due to steering, and slip angles over 90° are considered undesirable, we utilize the following bell-curve shaped reward function from [6]:

$$f_{\text{slip}}(\beta) = \left(1 + \left| \frac{\beta - c}{a} \right|^{2b} \right)^{-1}. \quad (5)$$

where

$$\beta = \arctan \left(\frac{v_y}{v_x} \right),$$

a, b, c : Parameters for altering the bell-curve shape.

Akin to [6], we choose $a = 20$, $b = 2.5$ and $c = 45$. In this configuration, slip angles around 45° are rewarded most, while there is a steep decline in reward towards 25° and 90° respectively.

V. EXPERIMENTAL SETUP

In the following, we introduce the simulation environment and detail the training process of both the state estimator and policy network.

A. Simulation Environment

Analogous to [6], we use a *gym* environment exported from the Unity Game Engine’s *ML-Agents* toolkit. For accurately modeling the vehicle dynamics within the simulation, the *vehicle-physics-pro* plugin is used.

B. State Estimator Network

We first use Supervised Learning to train the state estimator network to predict the state F given a sequence of three images. For this, we manually drive the vehicle in the simulation to collect image sequence and state F pairs. Apart from utilizing different driving styles to ensure a wide coverage of the state space, we also employ randomization of the environment. Upon an environment reset, the car is initialized with a random offset from the road center o_{init} , a random rotation relative to the path tangent θ_{init} , a random angular velocity ω_{init} , and a random velocity v_x and v_y . To have as much variance in the road curvature as possible, the vehicle is additionally placed at a random position on one of the three Formula 1 courses used during training. These are designed to contain sections with a wide range of curvatures ranging from straight sections to sharp *hairpin* turns. To increase robustness and generalization beyond the training data, we also utilize the following domain randomizations. These include altering the z-height of the camera, which is randomly shifted up or down by a value within $[-0.1, 0.1]$. Also, visual alterations such as changing the brightness of the camera image and rotating the main light source (sun) are used. The resulting dataset of 225000 image sequences to state F pairs is split into a training set with 180000 datapoints and two additional sets for validation and testing with 22500 each. We implement the network in PyTorch and train it until convergence.

C. Policy Network

During training, the policy network is given the ground-truth state F , which allows it to be trained independently of the state estimator. We thereby employ the same environment randomizations as described above, e.g. placing the vehicle at random locations around the track. The network is also implemented in PyTorch and trained using *stable-baselines3*’s implementation of Proximal Policy Optimization (PPO). Note that Action and Value networks share the same architecture. We train the network for maximum of 3.5 million steps, or until convergence.

VI. RESULTS

To gain insight into the state estimator network performance, we first evaluate its standalone performance before validating it in combination with the policy network.

A. State Estimator Network

Table I shows the overall Mean-Squared-Error (MSE) for estimating the state vector F given only a stack of three images, as well as the individual MSEs for each state variable, which allows some interesting insights. Firstly, they show that for the curvatures c_1 , c_2 , c_3 , which roughly describe the curvature of the road ahead in *near*, *medium* and *far* distance, estimates get worse with distance. It can also be seen that variables that require temporal information, such as velocities, generally have a higher error. This is especially true for the translational velocities v_x and v_y , which we hypothesize to be due to a lack of difference between consecutive images on straight road sections. With an overall MSE of 0.067, however, the state estimator reaches adequate accuracy.

	MSE
c_1	0.028
c_2	0.030
c_3	0.040
o	0.031
θ	0.045
ω	0.056
v_x	0.172
v_y	0.139
Overall	0.067

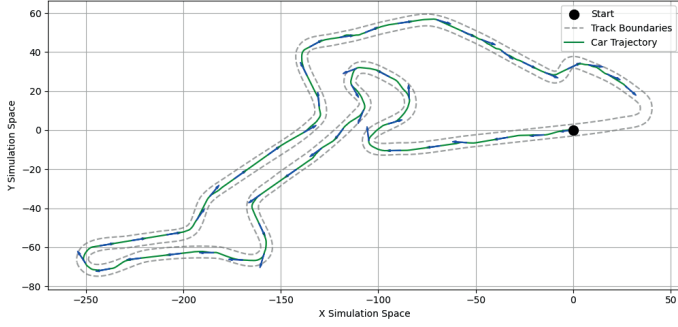
TABLE I: Mean-Squared-Error (MSE) of the state estimator network on its validation dataset.

B. Regular Driving Policy

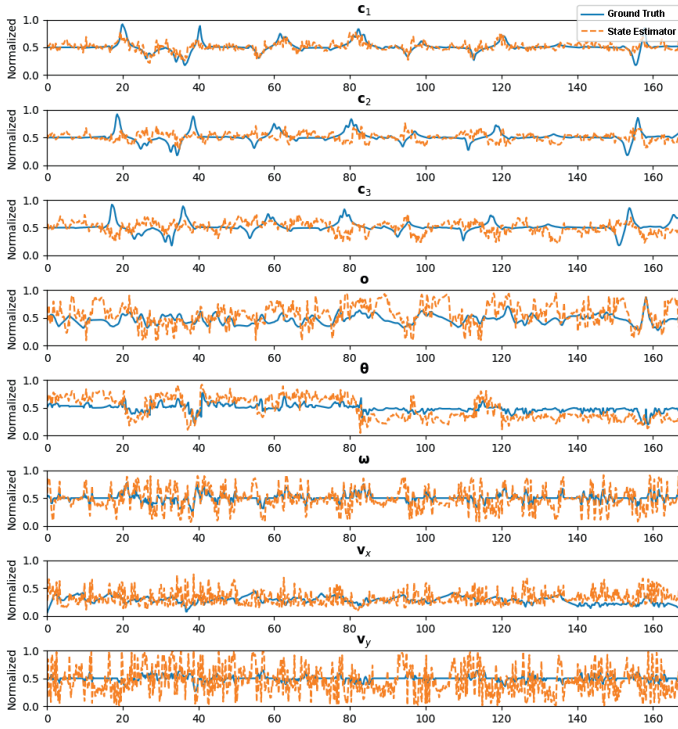
To assess the interaction of state estimator and policy network components, we let the combined network drive a lap on the *Nürburgring* track. The results can be seen in Figure 5. Note that in this case, the policy network was trained only for regular driving. Looking at Figure 5(b), it can be seen that despite a certain noise level, the state predictions generally follow the ground truth. While the vehicle remains rather careful, e.g. utilizing only around 30% of its maximum velocity, it does manage to complete almost an entire lap before the episode ends at 170 seconds.

C. Drifting Policy

To judge the drifting performance using only visual observations, we evaluate the combined network on a circular trajectory. Figure 6 shows the results. While the state estimator performs well, the resulting behavior does not depict a clean drift. The vehicle does only roughly manage to follow the desired trajectory. Rather than maintaining a smooth drifting motion, it rapidly enters and exits a drifting state. These oscillations are particularly noticeable from in 6(b).



(a) Vehicle path and reference.



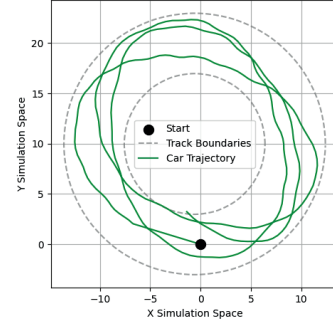
(b) State estimate and ground-truth.

Fig. 5: Regular driving policy network on the Nürburgring track, including state estimator predictions.

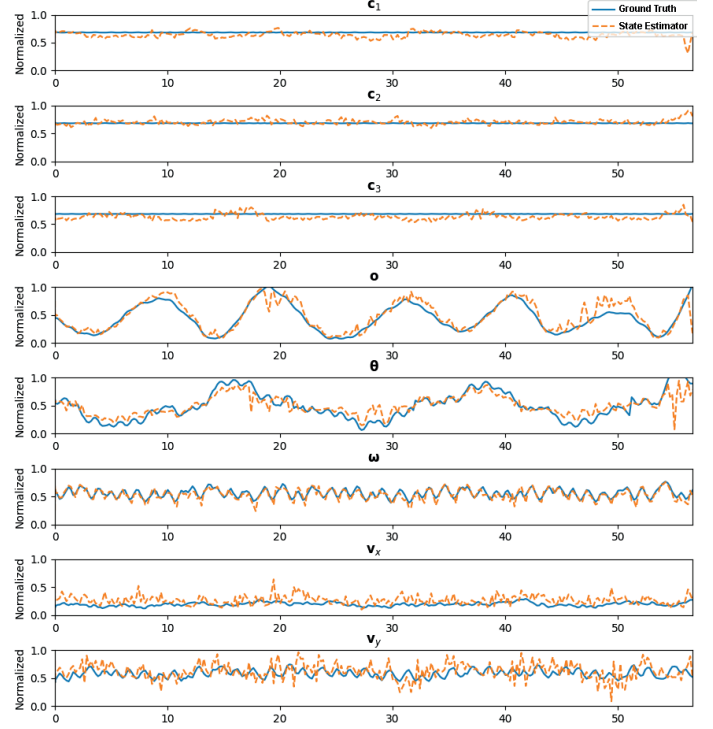
Finally, we examine the combined network’s behavior on a track section from the *Formula Drift*, which contains curves of various radii. The results, provided in 7, look similar to those on the circular trajectory in Figure 6(b). Looking at 7(b), again the state estimation is accurate and the vehicle manages to move along the desired trajectory, but rather than smoothly drifting it oscillates heavily.

VII. CONCLUSION

Our experiments show that using Reinforcement Learning to train a drifting controller using only visual input is generally possible, however not as straightforward as feeding in state measurements directly, as done by [6]. We successfully split the network into a feature estimator and a policy network. This allows for independent training and evaluation, which in turn leads to better interpretability and (faster) training convergence compared to single-network approaches.



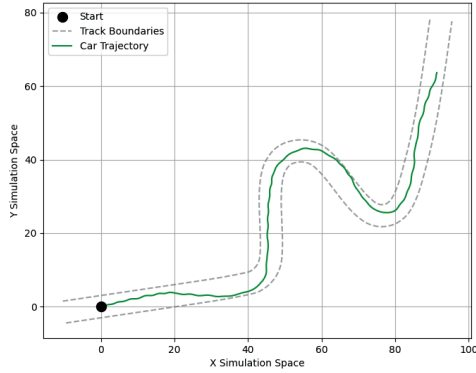
(a) Vehicle path and reference.



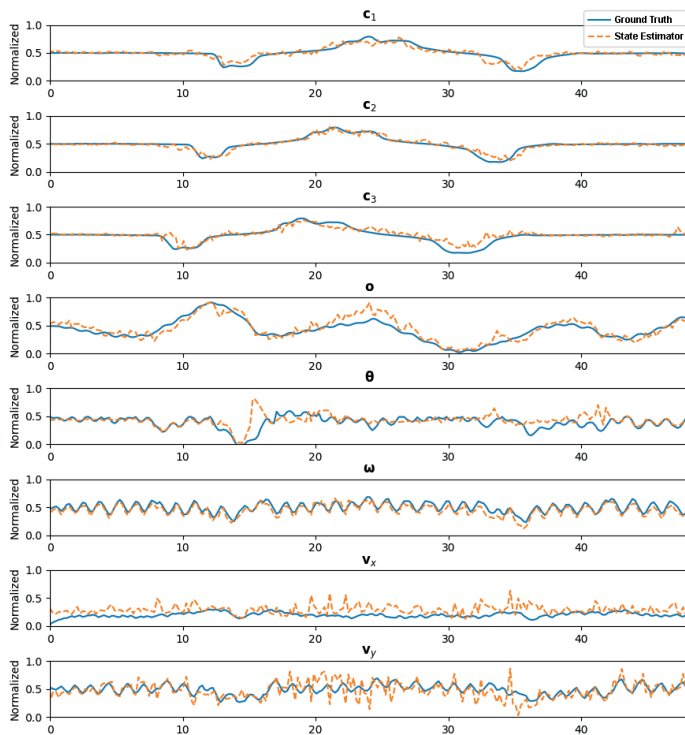
(b) State estimate and ground-truth.

Fig. 6: Drifting policy network on a circular track, along with the state estimator’s predictions.

While our feature estimator subnetwork achieves good overall accuracy, its error for velocities is especially high. We suspect this to be because of the lack of distinct visual features in the environment to infer speed from. Using a photorealistic simulation, particularly with a more irregular road texture, will likely overcome this. Our decision-making policy network also performs well, however, some issues, such as oscillating drifts and overly careful driving, remain. In future work, these shortcomings could be addressed by jointly continuing the training after individually (pre-)training feature extractor and policy network and combining them into one. This joint post-combination finetuning may also be done in a different simulation or even the real world, where training from scratch is not feasible. Pre-training the policy network in simulation, while the feature extractor is initialized with human-collected datapoints from the real target vehicle, may allow for zero-shot sim-to-real transfer.



(a) Vehicle path and reference.



(b) State estimate and ground-truth.

Fig. 7: Drifting policy on a section of a Formula Drift track and the state estimator's predictions.

REFERENCES

- [1] R. Y. Hindiyeh and J. Christian Gerdes, "A controller framework for autonomous drifting: Design, stability, and experimental validation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 136, no. 5, p. 051015, 2014.
- [2] J. Y. Goh, T. Goel, and J. Christian Gerdes, "Toward automated vehicle control beyond the stability limits: drifting along a general path," *Journal of Dynamic Systems, Measurement, and Control*, vol. 142, no. 2, p. 021004, 2020.
- [3] J. Baek, C. Lee, Y. S. Lee, S. Jeon, and S. Han, "Reinforcement learning to achieve real-time control of triple inverted pendulum," *Engineering Applications of Artificial Intelligence*, vol. 128, p. 107518, 2024.
- [4] M. Cutler and J. P. How, "Autonomous drifting using simulation-aided reinforcement learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5442–5448, IEEE, 2016.
- [5] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1247–1254, 2020.
- [6] F. Domberg, C. C. Wemmers, H. Patel, and G. Schildbach, "Deep drifting: Autonomous drifting of arbitrary trajectories using deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7753–7759, IEEE, 2022.
- [7] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 2070–2075, IEEE, 2018.
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

Challenges of Infrastructures for autonomous Buses in Cities: A review

Dr. Alessandro Becciu¹, Prof. Dr. Edwin N. Kamau²

¹Nuraxys GmbH, Im Brückfeld 33, 51491 Overath, alessandro.becciu@nuraxys.com, corresponding author

²University of Applied Sciences Cologne, Betzdorfer Str. 2 50679 Köln, Edwin.kamau@th-koeln.de

Keywords: Autonomous driving, autonomous shuttles, city infrastructures

Abstract

Autonomous shuttles are rapidly emerging as key technology to optimize urban vehicle traffic. Several cities worldwide are currently investigating this system of transport; however, there are still several open issues. This paper reviews the main critical challenges faced by municipalities in integrating autonomous shuttles into their infrastructure. It discusses routes, charging stations, communication nets, public perception and legal frameworks. In particular a proposal for the German regulations will be discussed.

1 Introduction

Nowadays, there is a constant request to find innovative methodologies aimed at improving the road transportation. Public transportation has the goal to bring down the overall traffic load by reducing the number of traffic jams generated by cars as well as by decreasing CO₂ emission levels. Public transportation at the same time faces challenges related to strict timetable difficult to adhere, overcrowded buses as well as lack of finding personnel. Autonomous shuttles or buses have reached level 4 of autonomy, where the driver can take hands off during a defined use case (see standard SAE J3016 [31]) and are rapidly emerging as key solutions to optimize urban vehicle traffic. They present a number of sensors and functionalities to extract the information from the environment in order to detect other traffic participants like vehicles, pedestrians, and bicycles, to avoid collision and to select a proper path for the driving destination. These autonomous vehicles promise to enhance mobility, reduce congestion [1], increase traffic safety [2], enhance travel speed and contribute to a more sustainable urban environment [3].

There are several cities worldwide that are testing autonomous shuttle services. For example, in the United States San Francisco [4] has started autonomous shuttle service on a fixed route, which connects residential neighborhoods with stores and community centers. In the recent past years in Europe more and more cities like Paris [5], Rome, and Turin [6] have delivered first autonomous shuttles services on a selected route of the city. In Germany several pilot projects with automated shuttles have already been carried out. According to the Association of German Transport Companies (Verband Deutscher Verkehrsunternehmen - VDV) over 40 municipalities have pilot projects listed so far [7] - see overview in Table 1. The municipalities plan to make consistent investments in order to integrate this new type of vehicles with the existing public transportation. For example, the city of Hamburg is planning to deploy above 10,000 autonomous shuttles by the 2030 [8]. However due to the radical change that the autonomous

shuttles introduce to the system of transportation and infrastructures, the municipalities face several challenges. For example, municipalities can decide to designate a certain route to the autonomous shuttles or modify existing ones. This has a strong impact to the traffic management system, because such a decision requests to make investments to create new lanes, as well as traffic signs and traffic lights. Moreover, such autonomous vehicles typically run on electric power, therefore municipalities need to install charging infrastructures along the route. Without considering the requirements imposed by autonomous shuttles, the system of transportation would perform worse.

In this paper we discuss the main challenges cities and municipalities face in order to integrate existing infrastructure with this new type of transportation. Section 2 will discuss the route, section 3 will describe the charging stations and communication network, section 4 will debate the people acceptance and finally section 5 will tackle the regulatory aspects especially related to Germany.

2 Route

To navigate through complex traffic autonomously, shuttles must acquire and learn from route information. The shuttle acquires information from the environment through a variety of sensors, including cameras, radars, lidars, IMU, GNSS, and ultrasound. Such information is used to create a map, where the shuttle is able to localize itself. The perception algorithms of the shuttle evaluate if the detected objects from the environment are static or dynamic and classify if they are obstacles or road users such as pedestrians or vehicles. Based on the distance and the speed of the detected objects, an evaluation about the risk of collision between the shuttle and the detected objects is performed. This information is then used in order to define an optimal trajectory for the shuttle to continue the travel without collisions.

The machine learning algorithms of the autonomous shuttle improve their knowledge of the route as well as the capability to travel every time the shuttle carries out a drive. The learning phase depends on the route complexity. The

Aachen - Interreg I-AT	Gera - EMMA	Mannheim – ShuttleME
Aachen - Marktliner	Hamburg - HEAT	Mannheim – RABus
Aachen - Urban Move	Hamburg – ReallabHH	Monheim am Rhein – Monheim -Shuttle
Bad Birnbach - Linie 7015	Hof - SMO	München - EasyRide
Bad Birnbach – Autonome Linien und Bedarfsverkehre	Iserlohn – a-BUS Iserlohn	München – TEMPUS
Bad Essen - HubChain	Karlsruhe – EVA shuttle	Neubäu am See - AutBus
Bad Soden – Salmünster- EASY	Keitum (Sylt) – NAF-Bus	Neustadt an der Weinstraße – Hambach-Shuttle
Berlin – See-Meile	Kelheim – KELRIDE	Oberhausen – Centro-Shuttle
Berlin – Stimulate	Kelheim - SUE	Osnabrück – HubChain
Berlin – Pole Position	Kronach - SMO	Regensburg – Emilia
Berlin – Shuttles & Co	Lahr – Lahr Shuttle	Rehau - SMO
Berlin – First Mover	Lauenburg an der Elbe – TaBuLa	Soest – Ride4all
Darmstadt Lincoln - Shuttle	Lauenburg an der Elbe – TaBuLa LOG	Stolberg - AS-NaSA
Drolshagen -SAM	Leipzig-ABSOLUT	Stuttgart – DiaMANT
Düsseldorf - KoMoD	Lennestadt Altenhudem- SAM	Stuttgart – U-Shift MAD
Eltville am Rhein - EASY	Ludwigsburg – DiaMANT	Überherrn - Terminal
Enge – Sande - Emil	Lunden – NAF-Bus	Waiblingen – AMAISE
Frankfurt am Main - EASY	Magdeburg – AS-UrbanÖPNV	Wiesbaden – DIGI-S
Frankfurt am Main - CUBe	Mainz – EMMA Shuttle	Wiesbaden – EASY
Friedrichshafen – RABus	Mainz – EMMA ² 2.0	Wusterhausen – AutoNV_ OPR
Gemeinde Rackwitz - FLASH	Mannheim - Rob-Shuttle	

duration of the process might take from a few days to a few weeks [9].

Autonomous shuttles are currently used in regular operation especially in closed environments, test-tracks as well as first-mile/last-mile transport to connect the passengers from and main public transport hub to his final destination and the other way around.

There are several challenges and critical factors for designing the route that need to be considered. A list is provided below [9, 10, 11, 12, 13, 14, 15,16].

Speed: In several studies the shuttles have been deployed on streets, where the allowed traveling speed was much

higher than the one reached by the shuttle itself. The shuttle's operational speed is in general between 15- 25 km/h. To communicate the presence of the shuttle to other traffic participants, some municipalities have installed traffic signs that highlight "slow vehicle", a speed limit of 30 km/h, and displays that show the actual speed along the route.

In project FABULOS in Helsinki a separate lane was created and the shuttle could reach up to 40 km/h [10].

Narrow and curvy streets: The dimensions (width and height) of the vehicle and the roadway play an important role in order to maintain a safety distance between the vehicle and the other road users. Depending on the shuttle's dimension, the minimum lane width must be derived.

Vehicles parked on narrow and curvy street can be perceived as an obstacle for the autonomous shuttle. This might cause an abrupt braking. In order to overcome such an issue, municipalities can install the traffic sign Clearway along the road [11]. Moreover, the vehicles driving behind the shuttle are also encouraged not to overtake, because the shuttle's abrupt braking can lead to safety risks [14].

Vegetation along the road: Branches that are too low above the street as well as bushes along the road might become an obstacle for the shuttle. A measure that can be taken is to trim the trees, remove the overhangs on plant islands and remove the roadside weed.

Road signs and lane markings: Autonomous shuttles detect the information of road signs and lane marking. In defining a route, it must be made sure that the signs and road markings are clearly recognizable by the sensors or cameras on the shuttle buses. If this is not the case, they must be adapted accordingly or replaced.

Interaction with other road users: several projects are carried out on open routes, where the autonomous shuttle needs to interact with other road users. Typical critical situations are:

- Pedestrians at crosswalks, at intersections, and walking along the road
- Cyclists riding along the road
- Vehicles at intersections in various conditions

Understanding how autonomous buses interact with other vehicles and human-driven vehicles is crucial. Lessons learned emphasize the importance of considering human behavior in decision-making algorithms of the shuttle.

Currently the safety operator, a safety trained person by the autonomous shuttle provider, who monitors the vehicle performance, checks if the intersection is safe and, in such a case, he presses a proceed type button [17]. In such a situation a reduction of the shuttle's speed could be also recommended.

3 Electric stations and communication network

3.1 Electric Powered Route

For several types of shuttles, the power of a battery pack ranges from 33 kWh to 110 kWh. Such batteries should

ensure operating time above 10 hours [20, 21, 22]. See examples on Table 2

Model	Battery Capacity
Holon Mover	110 kWh
EasyMile EZ10	38.4 kWh
Navya	33 kWh

However, the operating time of the shuttle is influenced by the topological conditions of the route. The length, the steepness of the route as well as the usage of heating and air conditioning have an impact on the shuttle's energy consumption. In a project on Lennestadt it was communicated that the shuttle was not able to properly cope with the steepness of the route and in the end, it was decided to choose a different route [19].

Since several buses run on electric power, there is the necessity to plan and install dedicated charging stations along the route. In general, the charging stations are defined according to the following standards: The SAEJ772 [23, 26] defines the charging levels used to classify the power, voltage and rated current of the charging station. The IEC 61851-1 [24, 26] defines the charging modes used to classify power supply, protection and communication control of the charging station. The IEC 62196-2 [25, 26] defines the charger types used to classify the different type of sockets used to supply power to the autonomous shuttle.

Charging time can last several hours. For example, EasyMile EZ10 can be charged for 7 hours on average [21]. According to best practice, it is suggested to carry out small maintenance of the shuttle close to the charging infrastructure, this because at the same time the shuttle can be charged [15].

3.2 Communication network

Autonomous shuttles utilize various sensor systems such as cameras, radars, lidars, and ultrasonic devices to detect environmental objects. At the same time the shuttles might need to interact with other devices on the infrastructures, such as traffic lights, traffic signs and cameras at the intersections as well as the other road users or other autonomous shuttles. This requires the installation of high-speed communication networks to ensure reliable connectivity. A typical example of communication between the shuttles is for example platooning. A platoon is an autonomous and cooperative group of shuttle buses that travel together while maintaining a minimum following distance. The first shuttle is typically the train conductor and uses shuttle-to-shuttle (vehicle-to-vehicle) communication technologies to determine the distance between shuttles, travel speed and train size.

The development directions pursued for the direct wireless information exchange of vehicle-to-everything (V2X) are currently the "Direct short-range communications (DSRC)" and "Cellular V2X (C-V2X)".

The DSRC [27] is an adapted variant of the WLAN standard IEEE 802.11p, which has been optimized for data exchange between vehicles and other road users. It entered in the automotive market already in 2015. The communication range exceeds 1 km and it works effectively at road user speeds up to 500 km/h. In comparison to the C-V2X, the DSRC transmits messages with very low data volume, whereas the C-V2X has the data volume of 4000GB per day. Moreover, the DSRC latency time is only 0.4 ms. On the other side the C-V2X [28] is a standard for V2X applications, which uses the Long-Term Evolution (LTE) technology present in the cellular devices. Therefore, the usage of the already existing infrastructure for cellular devices creates an advantage with respect to the DSRC in terms of installation and maintenance. Moreover, since the 5G networks are also becoming more established, such a technology allows a data exchange with a small latency time (1 ms) and high data rate (up to 10 Gbit/s).

The vast amount of data generated by the communication system poses a challenge to the municipalities, which need to establish data management and storage facilities in order to analyze the data and safely store it.

Finally, DSRC and C-V2X devices cannot currently communicate with each other, therefore this might be a challenge that needs to be considered in planning the proper infrastructure [29].

4 Public Awareness

Even if there is more and more information about the new advances on the autonomous driving vehicles, the public still perceives this new technology with skepticisms. Municipalities therefore face the challenge to introduce campaign to educate the population by highlighting the benefits of such technology and assuring about the safety aspects. Several projects [12, 13, 15] have indeed shown that feelings of unease or skepticism disappear during the trip.

The overall test was described as pleasant, exiting and the autonomous shuttle technology was perceived as useful.

Concerning the safety, the test participants expressed positive feeling in general. However, in the Ride4all project, where the shuttles were tested with people with disabilities, the passengers associated the positive feeling with the presence of the safety operator and that the seat belts were required. Lack of source of information from the driver might be an issue, if a passenger needs to travel alone. Moreover, the shuttles stop several times detecting bags and branches as obstacles and generating sudden stops. Besides the improvement of the obstacle detection algorithms, it is recommended to provide a smoother stopping motion, which would increase the perceived safety.

Another challenge that the shuttles are still facing is the speed. As previously stated, the shuttles move at a slow pace. The test passengers suggested working to increase the average speed while maintaining a sufficient level of safety. This would improve the service quality.

5 Regulatory aspects

Several municipalities have already adopted autonomous shuttles transportation, however defining regulations for this new type of transportation is still challenging due to the liability in case of accidents and involvement of insurance.

In the United States the legislation on the autonomous shuttles and autonomous vehicles differs from one state to the other. Several states require to have a safety operator on board during the drive, while other states allow drives without the safety operator. On the other hand, in China the government authorities have provided regulations that permit the testing of autonomous shuttles since 2017. The first autonomous shuttle Baidu was tested in Beijing in 2018.

In Europe, regulations vary by country and there is no clear standard. In Germany, the VDV has proposed a legal framework for fully automated and driverless Level 4 operation in public transport. See suggestions below.

Stepwise operating license: According to the framework the autonomous shuttles are allowed to drive, if an authorization for a general operating license is released by the federal government, as well as an approval by the federal state (Land), where the driving route has been defined. Tests on the field have shown that the approval on the autonomous vehicles depend on the environmental conditions. Therefore, it has been decided to use a stepwise approval approach, comparable to the approval of special vehicles, such as fire trucks or mobile cranes: vehicle approval and vehicle approval for the environment. The federal government is responsible for issuing the operating license, while the federal states decide where and under which circumstances the autonomous shuttles can be operated.

Prerequisites for driverless operation: Currently the autonomous shuttle can be operated, if a safety operator is present. In order to allow driverless operation, the VDV suggests to create the prerequisites to allow driverless operation. For instance, the shuttle shall be able to be deactivated at any time also externally if a risk is faced.

Special status for test vehicles: It suggested to enable remote control/teleoperation for test vehicles that currently do not comply with all the requirements

Operational-Technical-Supervisor: The loss of the driver needs to be compensated by a new legal figure, which has operational technical supervision according to VDV. This figure monitors and approves vehicle-related maneuvers and determines the deactivation of the vehicle. The operational and technical supervision is suggested to be allowed to be carried out from “outside” – for example via an external control center or in the local area/on site. Finally, it is suggested that operational and technical supervision is not performed for only one vehicle, it would be too expensive, but rather for the vehicles in the operational area.

6 Conclusions

Autonomous shuttles are being tested in various cities worldwide to integrate them with other transportation systems. Currently there are several challenges for the infrastructure to be resolved, such as the geometry of the route and its infrastructure, the slow speed of the shuttle as well as the fact that there is still the need for the presence of human operator also inside the vehicle. However, the tests show a positive feedback from the passengers and this creates the right environment to continue the development and overcome the challenges.

7 Literature

- [1] Vander Laan, Z., Sadabadi, K.F., 2017. Operational performance of a congested corridor with lanes dedicated to autonomous vehicle traffic. *International Journal of Transportation Science and Technology* 6, 42-52
- [2] Fernandes, P., Nunes, U., 2012. Platooning with ivc-enabled autonomous vehicles: Strategies to mitigate-communication delays, improve safety and traffic flow. *IEEE Transactions on Intelligent Transportation Systems* 13, 91-106
- [3] Greenblatt, J.B., Saxena, S., 2015. Autonomous taxis could greatly reduce greenhouse-gas emissions of us light-duty vehicles. *Nature Climate Change* 5, 860.
- [4] AP News: <https://apnews.com/article/autonomous-driverless-buses-robotaxi-san-francisco-802c39fd57adccaea604c7ee13a128> (accessed on 08 January 2024)
- [5] Smart Cities World: <https://www.smartcities-world.net/news/news/driverless-passenger-shuttle-launched-in-paris-ile-de-france-6262> (accessed on 08 January 2024)
- [6] Infra: <https://www.infrajournal.com/en/w/self-driving-buses-and-shuttles-the-5-most-advanced-cities-in-italy> (accessed on 08 January 2024)
- [7] <https://www.vdv.de/liste-autonome-shuttle-bus-projekte.aspx> (accessed on 23 January 2024)
- [8] <https://www.hamburg-news.hamburg/en/innovation-science/eur-26-million-autonomous-ride-pooling-hamburg> (accessed on 23 January 2024)
- [9] <https://www.camo.nrw/portfolio/camo-guide-automatisierte-shuttlebusse/> (accessed on 25 January 2024)
- [10] <https://fabulos.eu/helsinki-pilot/> (accessed on 26 January 2024)
- [11] https://ride4all.nrw/wp-content/uploads/2022/03/Praesentation_Abschluss_R4A.pdf (accessed on 25 January 2024)
- [12] https://ride4all.nrw/wp-content/uploads/2022/03/Projekt_Ride4All_Konzept_zur_Barrierefreiheit.pdf (accessed on 25 January 2024)
- [13] <https://pendelmobility.com/wp-content/uploads/2022/10/Deploying-Your-First-Autonomous-Vehicle.pdf> (accessed on 25 January 2024)
- [14] Anna Anund, Ricker Ludovic, Brunella Caroleo, Hugo Hardestam, Anna Dahlman, Ingrid Skogsmo,

Mathieu Nicaise, Maurizio Arnone, Lessons learned from setting up a demonstration site with autonomous shuttle operation – based on experience from three cities in Europe, *Journal of Urban Mobility*, Volume 2, 2022

- [15] https://www.probefahrt-zu-kunft.de/EASY_Brosch.pdf (accessed on 25 January 2024)
- [16] <https://www.hochbahn.de/de/projekte/das-projekt-heat> (accessed on 25 January 2024)
- [17] Li, X.; Guvenc, L.; Aksun-Guvenc, B. Autonomous Vehicle Decision-Making with Policy Prediction for Handling a Round Intersection. *Electronics* **2023**, *12*, 4670. <https://doi.org/10.3390/electronics12224670>
- [18] Iclodean, C.; Cordos, N.; Varga, B.O. Autonomous Shuttle Bus for Public Transportation: A Review. *Energies* **2020**, *13*,2917.
- [19] <https://www.lokalplus.nrw/lennestadt/gruenen-fraktion-zu-sam-peinlichkeit-hoch-drei-42460> (accessed on 28 January 2024)
- [20] <https://ridebeep.com/solutions/vehicles/holon> (accessed on 28 January 2024)
- [21] https://en.wikipedia.org/wiki/EasyMile_EZ10 (accessed on 28 January 2024)
- [22] https://navya.tech/wp-content/uploads/documents/Brochure_Shuttle_EN.pdf (accessed on 28 January 2024)
- [23] https://www.sae.org/standards/content/j1772_202401/ (accessed on 28 January 2024)
- [24] <https://webstore.iec.ch/publication/33644> (accessed on 28 January 2024)
- [25] <https://webstore.iec.ch/publication/64364> (accessed on 28 January 2024)
- [26] <https://www.emobilitysimplified.com/2019/10/ev-charging-levels-modes-types-explained.html> (accessed on 28 January 2024)
- [27] https://en.wikipedia.org/wiki/Dedicated_short-range_communications (accessed on 28 January 2024)
- [28] <https://5gaa.org/c-v2x-explained/> (accessed on 28 January 2024)
- [28] <https://www.gttwireless.com/dsrc-vs-c-v2x-comparing-the-connected-vehicles-technologies/#:~:text=Its%20use%20of%20WLAN%20is,to%201mS%20for%20C%2DV2X.> (accessed on 28 January 2024)
- [29] <https://www.vdv.de/eckpunkte-zum-rechtsrahmen-fuer-das-autonome-fahren-im-oev.aspx> (accessed on 28 January 2024)
- [30] Norton Rose Fulbright Automotive Vehicle. Available online: <https://www.automotive-iq.com/content-auto-download/5bcecf6ed1d92e3b5f3ec797> (accessed on 28 January 2024).
- [31] https://www.sae.org/standards/content/j3016_202104/ (accessed on 28 January 2024)

Automated failure and tolerance analysis as a combined consideration for the proof of safety of electronic systems

Roman Müller-Hainbach, Levent Ergün, and Prof. Stefan Butzmann
University of Wuppertal, Germany

Abstract

Several industries, especially the automotive industry, are moving towards higher and higher portions of electronics in their products. And they become increasingly complex and hard to analyze. This presents a challenging burden when it comes to the required proof of functional safety manufacturers have to provide for their products. Not only random faults, but also component tolerances can lead to unexpected safety hazards. Today's methods struggle to keep up with these challenges. This paper identifies some key challenges with today's methods and presents a new one, that uses computer automation and a model-based approach to improve this process. It is outlined how the new method does the same things better while also adding new capabilities.

Introduction

In this paper we want to look at methods to produce proof of functional safety of electronic systems. First we identify challenges with established methods, that are being used today. We will then present our new approach, that is heavily computer-aided and comes with new ways to perform both wider and deeper analyses on these systems. A new workflow using this approach will be explained after that. An example for demonstrational purpose is given in this paper as well. Our conclusion summarizes our findings and results and provides our thoughts on possible future developments.

1 Challenges of today's methods

Functional safety analysis poses significant challenges to engineers, who are facing increasing difficulties in dealing with ever more complex systems. Some of these difficulties are linked to deficiencies of the traditional methods, that are being used today.

1.1 Scope

High degrees of dynamic coupling between subsystems of different domains mean that functional safety analysis comes with increasingly broader scope in need of consideration. These functional interactions necessitate collaborative effort. The need for involvement from more engineers increases overall work load for this task.

1.2 Manual work

Classical methods of safety analysis, such as failure modes, effects, and diagnostic analysis (FMEDA) and fault tree analysis (FTA), rely heavily on manual work from (often dedicated) safety engineers. They consist in large parts of hypothetical reasoning and communication. Assistance

from computers often only comes in form of spreadsheets to hold the results of such labor. Model-based approaches to development exist ([1, 2, 3, 4]), but are rarely used extensively for safety analysis due to a lack of sophisticated methods of automation. These deficiencies lead to higher demands in work hours and inconsistencies.

1.3 Problem complexity and effort

An obvious challenge is to maintain the ability to handle increasing problem sizes. Especially the problem of combinatorial explosion in dual-point fault (DPF) analysis (quadratic in this case¹) quickly becomes insurmountable without measures to reduce complexity.

1.4 Prediction quality

Humans make mistakes. Even experienced experts are problematic as accurate predictors for system responses to fault events. Humans lack the consistent concentration to maintain a high level of accuracy. Another issue with human prediction is the reliance on intuition to solve complex problems. At some point guessing becomes inevitable.

Summary: Goals to achieve

We derive the following goals from the aforementioned challenges:

- Reduce human effort
- Ensure completeness
- Increase prediction quality
- Create a consistent tool-chain

We aim to achieve these goals with our computer-aided approach to functional safety analysis.

¹ $N = N_{SPF} + N_{DPF} = n \cdot M + \binom{n}{2} \cdot M^2 = n \cdot M + \frac{1}{2} \cdot n(n-1) \cdot M^2$

2 Computer-Aided Functional Safety Analysis

The general idea of computer-aided functional safety analysis is to reduce the required manual work as much as possible. Several ideas are presented in this paper that aim to shift the responsibilities of safety engineers towards modeling, decision and declaration work. Every task, that can be automated, benefits from the precision and reliability of computer algorithms.

2.1 Model-driven

A key aspect of enabling computer-aided functional safety is the adoption of model-based design. Only then can a computer program be tasked to work on any system. These models are the basis of communication between the engineers and any automation algorithms. The domain of modeling can vary. We have gathered experience with modeling in MATLAB/Simulink² and SPICE³, for which our approach has worked very well.

2.2 Library-driven

To accelerate the process of defining failure modes for each component in a given system, we chose to create a library system in which failure modes can be organized in component types. These libraries can grow over time and be reused with multiple systems. Applying these libraries to a new system is as simple as assigning each component in the model one of the defined component types. To further reduce the effort of this necessary assignment, we allow the definition of type specifications to help find components of each type. We find that most components can be assigned automatically using this approach.

2.3 Automated simulation

The main work of automation comes from the simulation of all fault scenarios, that can be generated from the declaration of components and failure modes. The preparation of the model, the simulation (using various simulators) itself and the gathering of results in a standardized format can be fully automated. This task in particular is also well suited for massive parallelization, offering great opportunities for time savings, given appropriate resources are allocated to compute equipment.

2.4 Fault injection and tolerance analysis

Our approach to safety analysis includes failure modes and tolerance effects analysis. We are able to inject failure modes through component parameter manipulation or substitution with a new failure model in-place.

We have implemented delayed failure injection (including temporal permutation of DPFs). This is especially important for DPF analysis. It needs to be emphasized, that a DPF event is not the occurrence of two faults at the same time. A single-point fault (SPF) scenario turns into a DPF

scenario by the advent of a second fault while the first fault is already present.

Tolerance effects can be applied to either component parameters or to a components' output signal amplitude (black box approach). Values are drawn from standard or custom distribution functions. The combination of failure modes effect and tolerance analysis is implemented on the technical side of things, but research on evaluation of this data has not yet concluded.

2.5 Analysis algorithms

We aim to automate the process of evaluating simulation results in terms of safety. Our proposed method is to use user-defined mathematical and logical test expressions to categorize all scenarios as either *safe*, *critical* or *diagnosed*. The test expressions used for this are divided into criticality and diagnosis criteria, which together form the safety criteria of a system. The decision process of this categorization is displayed in **Figure 1**. Based on these evaluations, latent faults (LFs) can be identified automatically. The calculation of safety metrics also becomes possible.

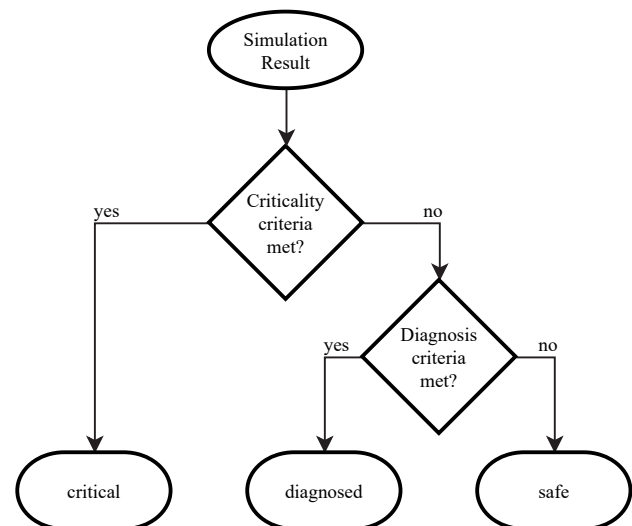


Figure 1 Decision tree for determining a scenarios' safety class using safety criteria

3 Workflow using new methods

There are several improvements to the workflow during development when using our new approach.

3.1 V-model development

In accordance with the industry standard ISO 26262[5] for functional safety in the automotive industry we seek to enable the development process according to the V-model. That is to begin with requirements and abstract, behavioral modeling. Our goal is to allow for functional safety analysis even with preliminary models, that don't include actual implementation details on a component level. These models are still very valuable to analyze using assumed black box failure behavior. We enable to work with such failure

²MATLAB/Simulink is a product of The MathWorks

³ngspice in particular

modes and provide preliminary results based on that. This approach allows for the detection of structural design flaws early on in the development process. For the later stages of development we also implemented methods to validate actual hardware prototypes using hardware-in-the-loop (HiL) simulators. This proved especially useful for software validation.

3.2 Complexity reduction through hierarchical modeling

Our approach to tackling the combinatorial explosion problem is reduce the number of simulations needed to cover all possible fault combinations. The key observation to exploit here is the fact, that complex systems can be broken down into smaller subsystems, that interface with each other. And that many faults result in a similar symptom at the module boundaries. It is therefore redundant to consider such faults individually while analyzing the effect of these faults on the overall system.

The simplest example to show this is a resistive divider network, consisting of two resistors of the same value. We consider the four standard failure modes of a resistor: open-circuit, short-circuit, resistance drift to 50 % of the original value and 200 %. Applying each failure mode to both resistors yields eight SPFs. Simulating these faults, sweeping the input voltage and observing the output voltage, shows far less distinct output curves than the eight actual curves there are. It is possible to describe the failure behavior of this subsystem under all component failures with just four failure symptoms. System simulation on the next higher level can work with these symptoms as new subsystem failure modes.

3.3 Iterative process

During the development process, the design is going to change many times. As is intended and provoked by the safety analysis results of early stage models. Because of this safety analysis becomes an iterative process, where found flaws can be mitigated as soon as they are discovered with reduced effort of redesigning because it is far more likely with this method, that obscure issues can be identified earlier than with traditional methods. The hierarchical modeling method allows for subsystem implementations to be added or changed over time with only the higher system level needing to be re-simulated, but not any other subsystem. This reduces the impact of changes made and makes the development process more agile.

4 Example of combined analysis

The following voltage monitoring circuit is an interesting example, as it passes both component tolerance, as well as fault injection tests. Our approach of combining the two proves, that these analyses individually leave enough room for critical scenarios to remain undetected under normal circumstances.

4.1 Example circuit

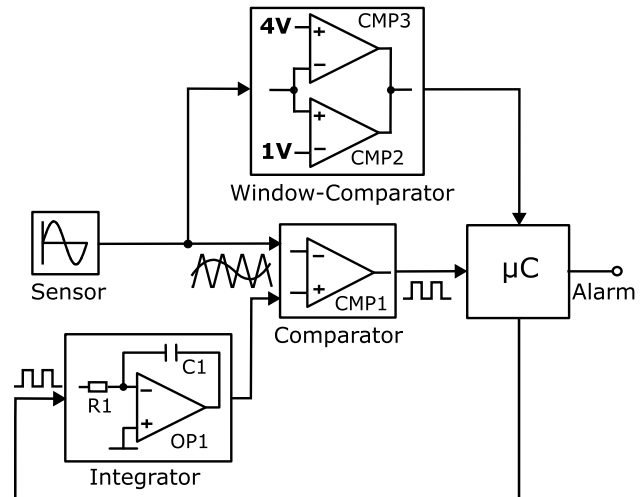


Figure 2 Example: voltage monitoring circuit

The circuit, as shown in **Figure 2**, monitors a sensor signal and produces an alarm signal whenever the sensor signal goes outside of an amplitude window of 1 V to 4 V. To achieve SPf functional safety the system is already designed in a redundant way: It consists of one static window comparator using comparators and reference voltages and a dynamic circuit, that creates a triangle signal using an operational amplifier integrator circuit and square wave signal generated in the microcontroller to compare against the sensor signal. This comparison produces a pulse width modulation (PWM) signal at the output of the comparator, which the microcontroller reads in as a digital signal. The duty-cycle of this signal correlates linearly with the signal amplitude. The microcontroller processes the logical output from the window comparator and measures the duty cycle of the PWM signal to deduce whether the sensor signal lies within the accepted amplitude window. It then uses a logical OR operation to combine both values into the actual alarm signal the circuit sends out.

4.2 Simulation setup and safety criteria

The simulation is done with a ramp signal to sweep all possible sensor signal amplitudes. This is important to verify the safety conditions under all possible circumstances. The circuit behaves quasi-static if the ramp signal is slow enough compared to the frequency used for the dynamic branch of the circuit.

The circuit is considered safe if the alarm signal activates for all input voltages outside the defined as safe amplitude window, with safety gaps applied. The expected output signal for this circuit is shown in **Figure 3**. Critical operating areas are annotated. The following analyses use this input signal and safety condition.

4.3 Tolerance analysis

This analysis investigates the influence of component parameter variations due to tolerances in manufacturing or operational conditions. We applied a tolerance of $\pm 5\%$ on

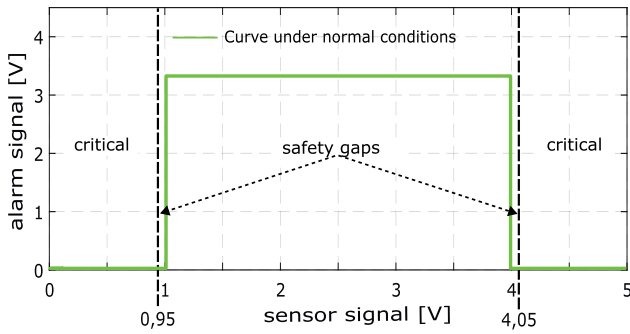


Figure 3 Alarm signal under normal conditions

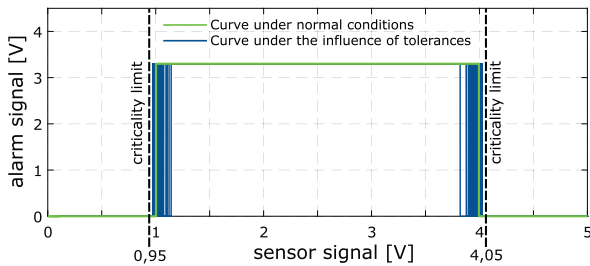


Figure 4 Tolerance simulation result

the resistance values and $\pm 10\%$ on the capacitance values. The comparators and the operational amplifier get tested with input offset voltages of ± 5 mV.

The results shown in **Figure 4** indicate, that the redundant measuring design and selected component accuracy achieve the goal of satisfying the safety condition under the ranges of possible component values due to tolerances. The circuit is considered functionally safe by the tolerance analysis.

4.4 Fault injection analysis

This analysis investigates the influence of component failures, which are expected to occur during a products' lifetime. We injected various standard component failures like *open-circuit*, *short-circuit* and *value drifts* of 50% and 200% for every resistor and capacitor. For the comparators and operational amplifier we assumed all possible *stuck* failures for every input and output.

Like with the tolerance analysis, **Figure 5** shows, that all SPF are covered by the redundant design. The circuit is considered functionally safe by the single-point fault injection analysis.

4.5 Combined analysis

The combined analysis does fault injection and tolerance application at the same time. For every SPF injected, a number of runs with different tolerance values for all other components is generated and simulated. **Figure 6** shows how the system is able to become critical under the influence of some SPFs, when other components' tolerances are considered as well. This is because the measure of redundancy can fail when one path is out of order because the SPF, that happens in it, while the other path goes out of the safe operating area due to the tolerances on its' com-

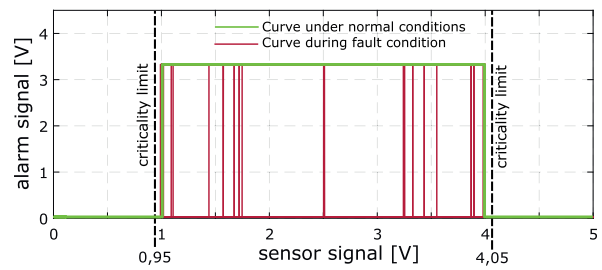


Figure 5 Fault injection simulation result

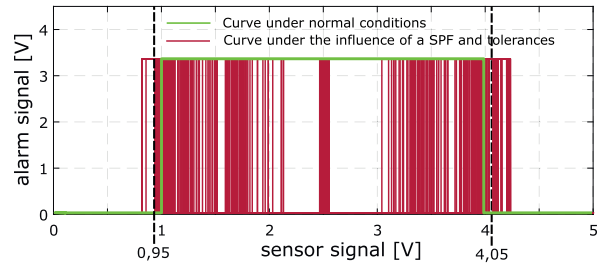


Figure 6 Combined simulation result

ponents. This shows how even a system with perfect diagnostic coverage for SPFs to rule out any DPF critical conditions can become critical under the influence of tolerances. Only a combined analysis approach is able to check for such cases.

5 Conclusion and future

In this work, we looked at the challenges of today's methods of functional safety proof and derived goals for a better approach to this problem. We presented such an approach, that heavily relies on computer aid through modeling and a high degree of automation in the process. The effectiveness of this new approach has been demonstrated with a real-world example.

5.1 Goals achievable

The goals for a modern method for proving the functional safety of electronic systems have been shown to be achievable. The task of predicting the *what-ifs* can be reduced to modeling work and declaring components and (reusable) failure modes. Database keeping enables users to ensure completeness of their analysis. By the shift in focus for developers from thinking through the behavior of a system case-by-case to just providing detailed enough models, the achievable prediction quality is increased by leveraging the speed and reliability of machines, computer simulation in this case. The consistency of the results of the carried out analysis can be managed by a program.

5.2 Benefits of our computer-aided methods

It has been shown how computer-aided methods enable developers to handle very complex systems by reduction in complexity and effort. A modular modeling approach allows for safety analysis to begin in an early stage of de-

velopment through the use of abstract black-box models, that can be switched out for concrete implementations later on. A digital workflow and consistent tool-chain allows for efficient use of resources, such as re-usable failure mode libraries and system modules. Relying on computer simulation gives rise to the scaling possibilities of hardware resources over human resources. The data obtained is available for powerful analysis algorithms, that can give valuable insights and find *the needle in the haystack*.

5.3 Future possibilities

There are many ways in which modeling capabilities can be improved to create more accurate prediction results. Such capabilities could include thermal modeling to conduct safe-operating-area (SOA) analysis, so called *smoke* analysis, during fault simulations. Results of which can be used to implement what we call a *fault sequence analysis*, where the incurring damage from operation outside of the SOA is considered by the model and correctly predicts the probable chain reaction of faults.

Another opportunity is fault modeling, which can be enhanced with stochastic variations. This takes on the idea of combining fault analysis with tolerance analysis and applies it to the failure models as well.

We also plan on using the results from tolerance analysis to highlight sensitivity hot spots in the design. Such insights could be valuable for optimizing component selection and therefor cost reduction in the final stages of development.

6 Literature

- [1] I. Pill, I. Rubil, F. Wotawa and M. Nica, "SIMULTATE: A Toolset for Fault Injection and Mutation Testing of Simulink Models," 2016 IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Chicago, IL, USA, 2016, pp. 168-173, doi: 10.1109/ICSTW.2016.21
- [2] I. Fabarisov, I. Mamaev, A. Morozov, K. Janschek: Model-based Fault Injection Experiments for the Safety Analysis of Exoskeleton System, IEEE Ninth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), IEEE, arXiv 2021, doi: 10.48550/ARXIV.2101.01283
- [3] Ezio Bartocci, Leonardo Mariani, Dejan Ničković, and Drishti Yadav. 2022. FIM: fault injection and mutation for Simulink. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). Association for Computing Machinery, New York, NY, USA, 1716–1720. <https://doi.org/10.1145/3540250.3558932>
- [4] Saraoğlu, Mustafa & Morozov, Andrey & Söylemez, Mehmet & Janschek, Klaus. (2017). ErrorSim: A Tool for Error Propagation Analysis of Simulink Models. 245-254. 10.1007/978-3-319-66266-4_16.
- [5] ISO 26262: Road vehicles – Functional safety, ISO, 2011

CSAM anomaly detection with AI

Jason Chia Zi Jie, Dr. Roland Krumm, Elmos Semiconductor SE, Dortmund, Germany
jason-zi-jie.chia@elmos.com; roland.krumm@elmos.com

Abstract

This paper presents an industrial application of unsupervised autoencoders for the task of anomaly detection within CSAM (C-type Scanning Acoustic Microscopy) images to detect potential defects in the semiconductor manufacturing process as part of the work done within the ECSEL project iRel4.0.

The approach presented leverages the high-yield nature of the semiconductor industry to generate a functional unsupervised autoencoder model with significantly shorter training time than supervised approaches to achieve comparable results. This approach has been validated with both synthetic wafers with manufactured defects as well as real production wafers. The approach extends current CSAM inspection methods by adding die-level resolution to potential defects as well as surpasses current manual optical inspection methods contributing to achieving Zero-Defect strategy in semiconductor manufacturing.

1 Introduction

As part of the ECSEL iRel4.0 project, this work presents an AI application toward improving reliability as part of the objectives within the project [1]. The work presented demonstrates the capability of using unsupervised autoencoders for anomaly detection in a high-yield and high-density application which can allow for yield recovery as well as improved failure detection contributing to Zero-Defect strategy.

1.1 Device and CSAM wafer image

The device utilized for the work is a burrowed cavity pressure sensor that requires a complete seal for functional performance. One of the quality control steps at Elmos include an optical inspection of CSAM imaging to detect defects on the manufactured wafer as in **Figure 1**. The type of defects can vary, but the key failures that are being detected are voids and scratches on the surface that will interfere with seal of the device which lead to reliability issues and potentially failure of the device.

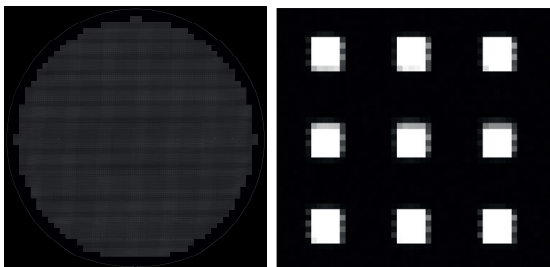


Figure 1 CSAM wafer image (left); zoomed in (right)

Typically wafers are rejected upon inspection if large defects are observed. This would then typically result in a complete scrap of the wafer. The presented method offers an automated approach to detect such defects as well as potentially allow for yield recovery on problematic wafers.

1.2 Current state of CSAM inspection

The current standard procedure for evaluating a CSAM image is through manual inspection. A trained technician evaluates the image as a whole and determines if the wafer is problematic depending if a defect is observed. However with the number of devices on a single wafer in excess of 55,000 units; a manual inspection can only be performed at the wafer-level due to cost and time limitations. As such, with manual inspection, only large defects are filtered resulting in smaller defects proceeding further into the manufacturing process or a wafer is scrapped and many potentially viable devices are discarded.

In addition to the large number of devices, the automotive semiconductor sector requires substantially high yield rates to achieve Zero-Defect strategy, which contributes to an extremely low amount of defective devices available. Furthermore it is not necessary that any failures from the wafer stem from physical defects that can be detected by the CSAM. This then contributes to the lack of defect data in particular resulting in an extremely imbalanced dataset.

As part of the standard, the CSAM images are produced by subjecting a wafer to a tool to perform the imaging. In full production, a standard resolution is utilized at 50 x 50 um per pixel. Further analysis of the image and the target device yields an area of interest no larger than 14x14 pixels. Some additional wafers were subjected to higher resolution scans but are not viable for mainline production due to the differentials in time and cost to perform these scans.

1.3 Model selection

With the defined objectives within the iRel4.0 project, the approach was designed to utilize machine learning methods to perform the evaluation of the devices for defect classification. However, given the nature of the semiconductor manufacturing industry, a supervised approach does not

appear to be practical. The reasoning for this can be based on the amount of work required to generate a well distributed training dataset encompassing all the desired defect classes and still be robust to be able to handle unseen or unknown defect classes. Furthermore the process would most probably need to be repeated for every single application which would require extremely large amount of work for limited gains.

As such a fully unsupervised approach was selected to automate the visual inspection of the CSAM images. This reduces the amount of work required to have a functional AI model with the capability to stop anomalies. This in turn could then be potentially utilized to begin gathering defect data to potentially be used in a more powerful supervised or semi supervised AI model.

2 Methodology

2.1 Image processing

An automated script was built on open-cv in Python to automatically detect and map all devices within the CSAM image of the wafer to their corresponding entities within the production database at Elmos. This preliminary step ensures that all devices on the wafer are accounted for and a specified region of interest around the target device is isolated for evaluation by the AI. The identification of potentially good devices can be seen in **Figure 2**.

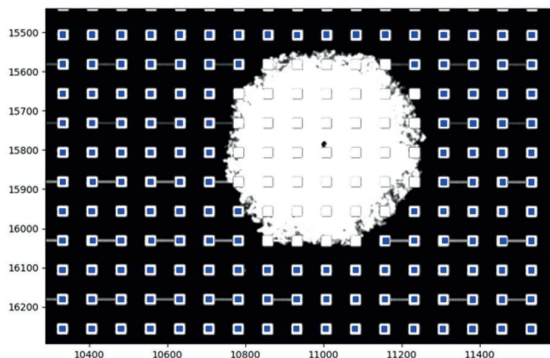


Figure 2 Die detection by image processing. Obscured devices are still mapped but marked as NOKs.

A by-product of the above process is the automatic filtering of obviously defective devices when large defects/damage obscure the target device as in **Figure 3**. As part of standard protocol, these large defects will automatically fail all devices within range if not the entire wafer.

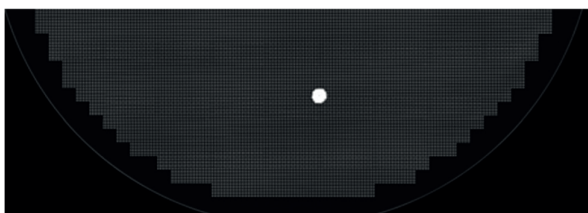


Figure 3 Example of extremely large defect

2.2 Autoencoder

The autoencoder is based of a standard convolutional autoencoder with some fine-tuning to reduce the number of convolutional layers to accommodate the imaging limitations. The block diagram of the autoencoder can be found in the following **Figure 4**.

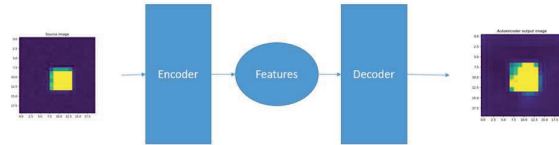


Figure 4 Autoencoder block

To function as an anomaly detector, the autoencoder model was trained on only good devices. Anomalies are then defined by the difference between the output image and the input image. A threshold was then set to determine if the device is anomalous or good.

2.3 Training and validation data

For this work, a set of wafers were manufactured within the iRel4.0 project with cooperating partners. These wafers were manufactured with a variety of simulated defects of varying sizes in the two primary failure modes which are scratches and embedded particles.

These wafers were then processed using the standard CSAM process as well as specially commissioned high resolution CSAM scans. Additionally, a subset sample of the devices was also tested to confirm failure of the device due to the manufactured defects.

For validation of the results a random sample of production wafers were selected for evaluation by the trained AI method and the results evaluated by a trained engineer to determine performance of the system.

3 Results

3.1 Manufactured defect wafers

After processing of the defect wafers, the model was demonstrated to be highly successful in detecting anomalies within the device. The performance of the AI was evaluated at all resolutions with result as in **Figure 4,5,6**

Resolution	Defects detected	Accuracy
50x50 um	11/13	~85%
20x20 um	12/13	~93%
10x10 um	13/13	100%

These results show that the unsupervised autoencoder approach can be extremely successful in detecting anomalies and its performance is limited to the actual resolution of the scans. For the missed defects, the size of the defects is actually smaller than the resolution resulting in the defects not being detected even through manual inspection. An example of an anomaly being detected can be seen in the following figures.

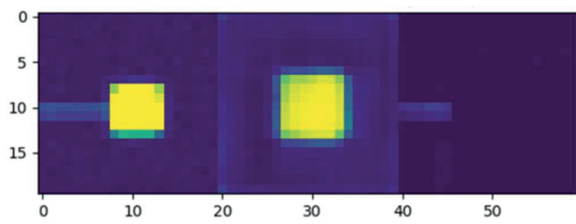


Figure 4 Input image (left), Autoencoder output (middle), Difference (right) at 50um x 50um resolution.

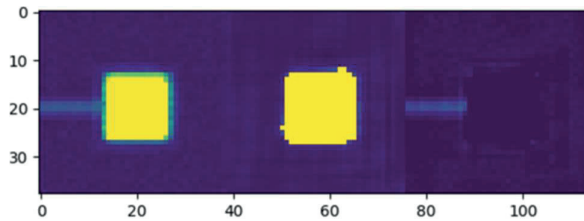


Figure 5 Input image (left), Autoencoder output (middle), Difference (right) at 20um x 20um resolution.

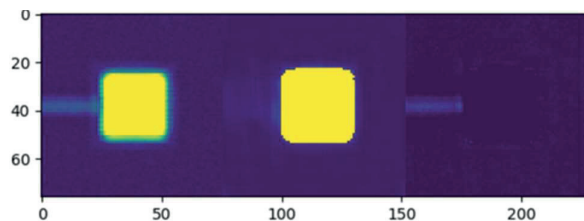


Figure 6 Input image (left), Autoencoder output (middle), Difference (right) at 10um x 10um resolution.

As can be visualized in the previous figures, the source resolution of the images play a critical role in the quality of the recreation of the autoencoder. However in situations where the anomaly information is not existent in the image due to the size of the defect being smaller than the resolution limits of the CSAM, these would not be detectable.

The anomalies detected were then analysed and demonstrated strong correlation with failure rates of the wafer. This can be seen in **Figure 7** where the failure detection by the AI demonstrate the same bimodal behaviour as the test indicator. Due to ambiguity in the manufacturing process resulting in imperfect 1-to-1 traceability, it is not feasible to confirm that all failures stem from the physical defects detected by the AI. However given the nature of the defects manufactured into the wafers this behaviour is to be expected in the failure rate as the manufactured defects are expected to cause failures in the device.

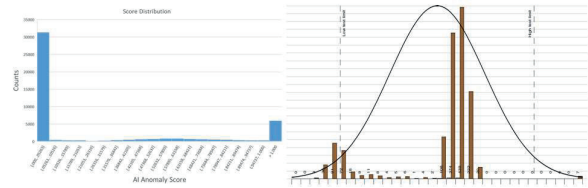


Figure 7 AI score (left) and test indicator (right) distribution.

3.2 Production wafers

The model was then applied to actual production wafers where the model was able to surpass manual inspection by detecting anomalies missed by the standard inspection procedure as can be seen in **Figure 8**.

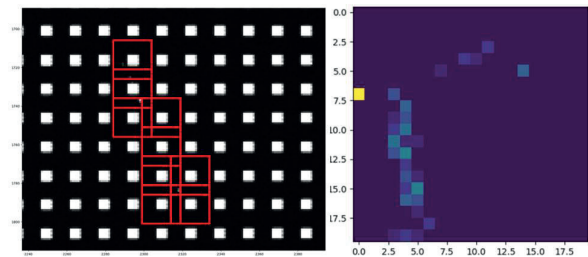


Figure 8 Cluster of anomalous device detected(left) Normalized error of single device from cluster (right)

Although judged to be non-critical, there were no available methods to verify that the anomaly detected did not cause a failure due to the current lack of single die traceability in the final pressure sensor system containing the pressure sensor as well as the signal conditioning IC in one package. Traceability toward the specific pressure sensor gets lost in the assembly process.

4 Conclusion

The presented work demonstrates an industrial application for unsupervised autoencoders surpassing human based optical inspection of CSAM images. This work can be further extended to other optical inspection based tasks and have been utilized by Elmos for other image inspection applications.

Although the AI model was demonstrated to achieve high performance, it is limited in great part to the original resolution of the images. If the anomalous component or the defect is non-existent in the source image, the approach will be unsuccessful in detecting such defects. This should be taken as an indication to imaging vendors and test manufacturers that with AI, large scale optical inspection is now possible and should be leveraged to improve quality and reliability of devices in any manufacturing process.

Given the nature of manufacturing where stopping defective devices is extremely important, it is extremely valuable to begin AI based control with an unsupervised approach to gather the required defect data or required information to form a suitable dataset for more powerful supervised approaches that can allow engineers to automatically classify defects as well.

4.1 Acknowledgement

iRel40 is a European co-funded innovation project that has been granted by the ECSEL Joint Undertaking (JU) under grant agreement No 876659. The funding of the project comes from the Horizon 2020 research program and participating countries. National funding is provided by Germany, including the Free States of Saxony and Thuringia, Austria, Belgium, Finland, France, Italy, the Netherlands, Slovakia, Spain, Sweden, and Turkey.”

5 Literature

- [1] iRel4.0 project homepage; www.irel40.eu Accessed: 29.01.2024

Investigation of the real-time feasibility of NMPC for air-path control in automotive fuel cell systems

Thuc Anh Nguyen^a, Verena Neisen^a, and Dirk Abel^a

^aInstitute of Automatic Control, RWTH Aachen University, Aachen, Germany

Abstract

This paper presents the development and evaluation of a nonlinear model predictive control (NMPC) algorithm in the context of air-path control for automotive fuel cell hybrid systems, with a focus on its real-time performance. The study is motivated by the need to establish real-time capability as a key criterion for the practical deployment of such advanced control systems. Our approach involves the design of an optimal control problem, followed by its efficient conversion into a nonlinear program, to which the sequential quadratic programming method is applied. The resulting quadratic programs are solved by the open-source numerical solver HPIPM. A notable outcome of this study is the controller's mean turnaround time of 10.3 ms in numerical simulations on embedded hardware, utilizing a Gauss-Newton Hessian approximation. While this marginally exceeds the desired sampling time of 10 ms, our results demonstrate the potential of NMPC as a viable solution for managing the complexities inherent in automotive fuel cell hybrid systems, potentially contributing to enhance their operational reliability, efficiency and durability.

1 Introduction

Amid global efforts to reduce carbon emissions, fuel cells (FCs) have emerged as a promising zero-emission technology with diverse applications. Proton exchange membrane (PEM) FCs, which convert the chemical energy in hydrogen directly into electrical energy, have gained significant attention in the automotive industry. Apart from zero local emission of harmful byproducts, they promise high efficiency, high power and energy density, rapid startup, low operating temperatures, and fast refueling time [1].

Typically, FC vehicles operate on a hybrid power system, combining a traction battery with a PEM fuel cell system (FCS). The interaction between the battery and the FCS and the operating conditions within the FCS critically impact hydrogen consumption and lifespan of the fuel cell hybrid vehicle (FCHV). Therefore, effective control strategies significantly contribute to optimizing FCHV performance and ensuring robust operation across various conditions [2]. They guarantee that the vehicle meets the power demands while maintaining safety and efficiency. To address these control objectives, this work utilizes model predictive control (MPC). More specifically, we apply nonlinear model predictive control (NMPC), as we expect the system behavior to be more accurately captured through a nonlinear prediction model, especially during rapid load changes. Obtaining a control model appropriate for NMPC use, including the judicious selection of control inputs, states, and outputs, presents its own set of challenges. Once a suitable model is established, the ability of NMPC to directly handle nonlinear dynamics, constraints, and objectives, as well as systems with multiple in- and outputs, allows for a natural translation of design requirements into mathematical statements. Nonetheless, a significant challenge in practical NMPC applications is solving the resulting dynamic optimization problem in real time. Thus, the primary focus of this paper is to address the demanding real-time constraints and assess the real-time capability of our proposed NMPC algorithm.

1.1 State of the art

Several researchers have addressed the challenge of real-time capability in NMPC methods by developing numerical techniques to enhance their computational efficiency, which have been applied across various applications [3, 4]. Others have applied NMPC to automotive FCSs, but often neglecting the practical aspect of real-time capability [5, 6]. Despite the available methods, achieving fast, reliable, real-time NMPC solutions remains a recurring challenge due to the intricacies of every application.

Research addressing this gap remains limited. Schmitt et al. have notably demonstrated the real-time capability of NMPC algorithms for the air-path control of a small-scale FCS. In their studies, [7] uses a Wiener model as the control model, while [8] employs a physics-based control model. Efficient numerical integration schemes and condensing methods are explored in [8] and [9], respectively. This work, however, uniquely focuses on the FCS within an automotive hybrid power system, incorporating an auxiliary battery system to manage highly transient and safe operation. Furthermore, our work adopts more stringent time constraints compared to the sampling times in Schmitt et al.'s studies.

1.2 Contribution

This paper introduces an NMPC-based control algorithm designed for the air-path control of an automotive FCS within a hybrid power setup. The control objectives are formulated within an optimal control problem (OCP), which is discretized efficiently to reduce computation time while maintaining simulation accuracy. The resulting nonlinear program (NLP) is solved via sequential quadratic programming (SQP), employing a structure-exploiting solver for the emerging quadratic programs (QPs). Finally, we assess the algorithm's real-time capability on a rapid prototyping platform. The paper is structured as follows: Section 2 outlines the control problem including the control architecture. Section 3 presents the physics-based model of the

FCS and the hybrid power system. Section 4 details the NMPC algorithm and Section 5 discusses simulation results.

2 Control task & control architecture

Rapid response is crucial in automotive propulsion systems, often outweighing efficiency due to the need to adapt to quickly changing and unpredictable loads. The primary control objective for the FCHV is therefore to consistently meet dynamic power demands requested by the driver. The response capabilities of FCSs are, however, limited by the intricate dynamics of the FC stack and its balance-of-plant (BoP) subsystems. More specifically, the dynamic performance is predominantly determined by the air-supply subsystem, as oxygen kinetics and mass transport become the limiting factors when pressurized pure hydrogen is supplied. To address these limitations, a FCS is commonly integrated with a battery via a DC/DC converter. Alongside performance, safe operation of the FCS is paramount. For the air path, this entails ensuring sufficient oxygen excess ratio (OER) and adhering to compressor surge and choke limits during transients. Moreover, a key goal is to optimize the FCS's efficiency, achieved by operating the FC stack under favorable conditions and reducing parasitic losses, especially from the compressor, which can account for up to 14 % of stack power [10]. To address these control objectives, we adopt the control architecture depicted in Fig. 1.

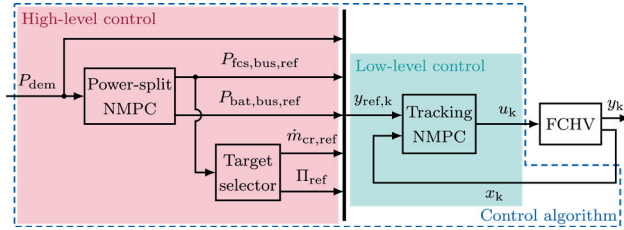


Fig. 1 Schematic overview of the control architecture.

The high-level control consists of a power-split NMPC and a target selector. The NMPC manages the power distribution between the FCS ($P_{fcs,bus,ref}$) and the battery ($P_{bat,bus,ref}$), executing on a 1 s sampling time. To this end, the underlying economic cost function maximizes FCS efficiency while maintaining a stable battery state of charge (SOC). At the same time, an equality constraint enforces adherence to the total power demand (P_{dem}), if feasible. The optimization problem is based on a dynamic single-state battery model and a static approximation of the FCS. Utilizing the static FCS model, the target selector employs a static optimization to derive a closed-form solution for the maximum FCS efficiency, given the required FCS power output $P_{fcs,bus,ref}$. From the solution, we also obtain the compressor's optimal operating point, given by the corrected compressor mass flow rate $\dot{m}_{cr,ref}$ and pressure ratio Π_{ref} .

The low-level control comprises a tracking NMPC that directly manipulates the FCHV. This controller operates with a 10 ms sampling time and integrates a dynamic model of

the FCS (Sec. 3.1 – 3.4) and a static model of the battery (Sec. 3.5). The references for the low-level NMPC, $y_{ref,k}$, are derived from the total power demand, the optimal power distribution from the power-split NMPC, and the compressor's optimal operating point. Its control objectives include: (i) precise tracking of power demand, (ii) maintaining adequate OER, (iii) avoiding compressor surge and choke, (iv) following compressor setpoints and (v) ensuring real-time feasibility. For the latter, the related nonlinear optimization problem must be solved within 10 ms, which requires careful controller design. This is the focus of this paper.

3 Control-oriented model

The control-oriented model of the FCHV, utilized as the prediction model within the low-level NMPC, consists of a FCS and a battery, as depicted in Fig. 2.

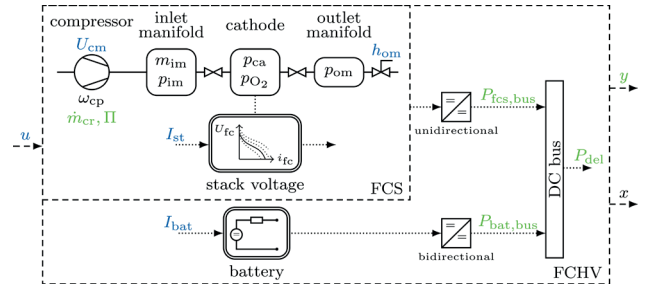


Fig. 2 Schematic overview of the model structure used for control-oriented modeling the FCHV.

A typical FCS includes the FC stack and various BoP subsystems. Among those, we specifically only consider the air-path subsystem. The air-path components, including the compressor, inlet manifold, cathode, and outlet manifold, as well as the stack voltage, are modeled based on [11]. The zero-dimensional model treats each air-path component as a single control volume, akin to a continuously-stirred tank reactor, without any spatial variations. The dynamic states of the model stem from this lumped-parameter approach to the air-path system, while the stack voltage as well as the battery are described by static algebraic expressions. For completeness, we provide most of the model equations in the subsequent sections.

3.1 Compressor model

A lumped rotational inertia J_{cp} is used to represent the compressor with the compressor rotational speed ω_{cp} :

$$J_{cp} \cdot \frac{d\omega_{cp}}{dt} = \tau_{cm} - \tau_{cp} \quad (1)$$

Here, τ_{cm} is the compressor motor torque and τ_{cp} is the load torque required to drive the compressor. The former is calculated using a simplified static DC motor model

$$\tau_{cm} = \frac{\eta_{cm} \cdot k_t}{R_{cm}} \cdot (U_{cm} - k_v \cdot \omega_{cp}), \quad (2)$$

where k_t , R_{cm} and k_v are motor constants, η_{cm} is the motor mechanical efficiency and the voltage U_{cm} is the motor control command. The latter is given by the thermodynamic equation

$$\tau_{cp} = \frac{c_p}{\omega_{cp}} \cdot \frac{T_{atm}}{\eta_{cp}} \cdot \left(\Pi^{\frac{\gamma-1}{\gamma}} - 1 \right) \cdot \dot{m}_{cp}^{out}, \quad (3)$$

where c_p is the constant-pressure specific heat capacity and γ is the ratio of the specific heat capacities of air, η_{cp} is the compressor efficiency, $\Pi = p_{im}/p_{atm}$ is the ratio of the pressure inside the inlet manifold (Sec. 3.2) and the atmospheric pressure, and T_{atm} is the atmospheric temperature. The compressor flow is $\dot{m}_{cp}^{out} = \dot{m}_{cr} \cdot \delta/\sqrt{\theta}$, where $\theta = T_{atm}/288$ K and $\delta = p_{atm}/1$ atm. The corrected compressor flow $\dot{m}_{cr} = f(\Pi, \omega_{cp})$ and the efficiency $\eta_{cp} = f(\dot{m}_{cr}, \Pi)$ are modeled using static maps. The temperature of the air leaving the compressor is

$$T_{cp}^{out} = T_{atm} + \frac{T_{atm}}{\eta_{cp}} \cdot \left(\Pi^{\frac{\gamma-1}{\gamma}} - 1 \right) \quad (4)$$

and the power consumed by the compressor motor is

$$P_{cm} = \frac{1}{R_{cm}} \cdot (U_{cm}^2 - k_v \cdot \omega_{cp} \cdot U_{cm}). \quad (5)$$

3.2 Inlet manifold model

The inlet manifold includes pipe and stack manifold volumes between the compressor and the FCs. Its dynamics is governed by mass continuity and energy conservation laws, which yield differential equations for the inlet manifold mass m_{im} and pressure p_{im} :

$$\frac{dm_{im}}{dt} = \dot{m}_{cp}^{out} - \dot{m}_{im}^{out} \quad (6)$$

$$\frac{dp_{im}}{dt} = \frac{\gamma \cdot R_a}{V_{im}} \cdot (\dot{m}_{cp}^{out} \cdot T_{cp}^{out} - \dot{m}_{im}^{out} \cdot T_{im}) \quad (7)$$

Here, R_a is the gas constant for ambient dry air, V_{im} is the inlet manifold volume, and T_{im} is the temperature of the flow inside the manifold, computed from the ideal gas law. The exit flow, \dot{m}_{im}^{out} , is calculated as a function of p_{im} and the cathode pressure p_{ca} (Sec. 3.3.2) using a linearized nozzle flow equation with nozzle constant $k_{ca,in}$:

$$\dot{m}_{im}^{out} = k_{ca,in} \cdot (p_{im} - p_{ca}) \quad (8)$$

The oxygen mass fraction in the inlet manifold's dry air, $y_{O_2}^{im}$, is derived from the oxygen mole fraction in the incoming ambient dry air, $x_{O_2}^{atm}$, using the molar masses of oxygen M_{O_2} and nitrogen M_{N_2} :

$$y_{O_2}^{im} = \frac{x_{O_2}^{atm} \cdot M_{O_2}}{x_{O_2}^{atm} \cdot M_{O_2} + (1 - x_{O_2}^{atm}) \cdot M_{N_2}} \quad (9)$$

Finally, the outgoing mass flow rate of oxygen, $\dot{m}_{O_2,im}^{out}$, is:

$$\dot{m}_{O_2,im}^{out} = y_{O_2}^{im} \cdot \dot{m}_{im}^{out} \quad (10)$$

3.3 Fuel cell stack model

In this study, the modeling of the FC stack is focused on two components: the stack voltage model and the cathode flow model. The original model [11] encompasses a broader range of submodels, including an anode flow model and a membrane hydration model. In this adapted version, these are omitted to concentrate on the air-path dynamics.

3.3.1 Stack voltage model

Given that the FC stack comprises multiple FCs connected in series, and assuming that all cells are identical, the stack

voltage U_{st} is calculated as the product of the number of cells and the cell voltage: $U_{st} = n \cdot U_{fc}$. The combined effect of thermodynamics, kinetics, and ohmic resistance determines the cell's output voltage

$$U_{fc} = E - U_{act} - U_{ohm} - U_{conc}, \quad (11)$$

where E is the open-circuit voltage (OCV), U_{act} is the activation loss, U_{ohm} is the ohmic loss, and U_{conc} is the concentration loss. A detailed explanation of the FC voltage, also known as polarization characteristic, is provided in [11]. As we assume instantaneous electrochemical reaction, the FC voltage is given as a static function of the current density i_{fc} and the dynamically varying stack variables of oxygen p_{O_2} and hydrogen partial pressures p_{H_2} , cathode pressure p_{ca} , stack temperature T_{st} and membrane humidity λ_m . The current density is defined as stack current per unit of cell active area: $i_{fc} = I_{st}/A_{fc}$.

In this work, our focus is on the dynamics of the air-path variables, specifically p_{O_2} and p_{ca} . Consequently, we assume perfect control of the anode pressure, i.e. $p_{an} = p_{ca}$, and for further simplification, we set $p_{H_2} = p_{an}$. Additionally, we assume that T_{st} and λ_m are accurately controlled with negligible delay to constant values of 80 °C and 12.5, respectively.

3.3.2 Cathode flow model

This model captures the dynamics of cathode airflow. By neglecting the presence of water vapor, the cathode's dry air mass is equal to its total mass. Applying mass continuity to the dry air and oxygen in the cathode volume V_{ca} , and utilizing the ideal gas law, we obtain

$$\frac{dp_{ca}}{dt} = \frac{R_a \cdot T_{ca}}{V_{ca}} \cdot (\dot{m}_{im}^{out} - \dot{m}_{O_2}^{react} - \dot{m}_{ca}^{out}), \quad (12)$$

$$\frac{dp_{O_2}}{dt} = \frac{R \cdot T_{ca}}{M_{O_2} \cdot V_{ca}} \cdot (\dot{m}_{O_2,im}^{out} - \dot{m}_{O_2}^{react} - \dot{m}_{O_2,ca}^{out}), \quad (13)$$

where R is the universal gas constant, and the cathode air temperature T_{ca} is assumed equal to the stack temperature T_{st} . Sec. 3.2 defines the inlet flow rates, while the outlet flow rates are

$$\dot{m}_{ca}^{out} = k_{ca,out} \cdot (p_{ca} - p_{om}), \quad (14)$$

$$\dot{m}_{O_2,ca}^{out} = y_{O_2}^{ca} \cdot \dot{m}_{ca}^{out}. \quad (15)$$

Here, \dot{m}_{ca}^{out} is obtained analogously to (8), with nozzle constant $k_{ca,out}$ and outlet manifold pressure p_{om} (Sec. 3.4). Similarly, $y_{O_2}^{ca}$ is determined as in (9), with $x_{O_2}^{ca} = p_{O_2}/p_{ca}$. Electrochemical principles are used to calculate the rate of oxygen consumption $\dot{m}_{O_2}^{react}$ from the stack current

$$\dot{m}_{O_2}^{react} = \frac{M_{O_2} \cdot n}{4 \cdot F} \cdot I_{st}, \quad (16)$$

where F is the Faraday constant. The OER λ_{O_2} corresponds to the ratio between the oxygen supplied and the oxygen reacted:

$$\lambda_{O_2} = \frac{\dot{m}_{O_2,im}^{out}}{\dot{m}_{O_2}^{react}} \quad (17)$$

3.4 Outlet manifold model

Contrary to the inlet manifold where temperature variations are significant, the outlet manifold temperature T_{om} is assumed constant and equal to T_{st} . The outlet manifold

pressure p_{om} is then derived using mass conservation and the ideal gas law

$$\frac{dp_{om}}{dt} = \frac{R_a \cdot T_{om}}{V_{om}} \cdot (\dot{m}_{ca}^{out} - \dot{m}_{om}^{out}), \quad (18)$$

where V_{om} is the outlet manifold volume. The air flow rate exiting the outlet manifold, \dot{m}_{om}^{out} , is calculated via the non-linear nozzle equation for subcritical flow. It is dependent on p_{om} and back-pressure valve opening area, $h_{om} \cdot A_T$, with C_D being the discharge coefficient of the nozzle:

$$\dot{m}_{om}^{out} = \frac{C_D \cdot h_{om} \cdot A_T \cdot p_{om}}{\sqrt{R \cdot T_{om}}} \cdot \left(\frac{p_{atm}}{p_{om}} \right)^{\frac{1}{\gamma}} \cdot \Psi \quad (19)$$

$$\Psi = \sqrt{\frac{2 \cdot \gamma}{\gamma - 1} \cdot \left[1 - \left(\frac{p_{atm}}{p_{om}} \right)^{\frac{\gamma-1}{\gamma}} \right]} \quad (20)$$

3.5 Battery model

The battery is represented using a static circuit model, which depicts the battery as an OCV source U_{ocv} in series with an internal ohmic resistance R_i . The terminal voltage U_{bat} is:

$$U_{bat} = U_{ocv} - R_i \cdot I_{bat} \quad (21)$$

U_{ocv} is modeled as the sum of a constant term E_0 , a polarization component E_{pol} , and an exponential term E_{exp} :

$$U_{ocv} = E_0 - E_{pol} + E_{exp} \quad (22)$$

Both E_{pol} and E_{exp} are static functions of the battery SOC, with E_{exp} capturing the battery's nonlinear behavior. [12]

3.6 Power equations

The power generated by the FC stack, P_{st} , is the product of the stack voltage and the stack current: $P_{st} = U_{st} \cdot I_{st}$. To calculate the net power output of the FCS, $P_{fcs,net}$, we deduct the power used by the compressor motor, P_{cm} , and a fixed auxiliary power, P_{aux} , where P_{aux} accounts for the consumption of other actuators in the FCS:

$$P_{fcs,net} = P_{st} - P_{cm} - P_{aux} \quad (23)$$

The power losses in the DC/DC converter are accounted for using the converter efficiency $\eta_{DC/DC}$, which is modeled as a second-order polynomial fitted to measurement data:

$$P_{fcs,bus} = \eta_{DC/DC} \cdot P_{fcs,net} \quad (24)$$

For the battery, the net power $P_{bat,net}$ is the product of the battery voltage and current, factoring in the quantity of battery packs and cells: $P_{bat,net} = U_{bat} \cdot I_{bat} \cdot n_{packs} \cdot n_{cells}$. Power losses from DC/DC conversion are considered with the same converter efficiency as in (24):

$$P_{bat,bus} = \eta_{DC/DC} \cdot P_{bat,net} \quad (25)$$

Finally, the total delivered power P_{del} at the DC bus is

$$P_{del} = P_{fcs,bus} + P_{bat,bus}. \quad (26)$$

4 Low-level NMPC algorithm

4.1 Prediction model formulation

For simplicity, the dynamic states and output equations of the model in Sec. 3 are denoted as:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p(t)) \\ y(t) &= g(x(t), u(t), p(t)) \end{aligned} \quad (27)$$

For MPC design, we adopt the delta formulation of (27). This approach allows for constraints to be imposed on the rates of change of the inputs, while, given the application of multiple shooting, resulting in a sparse OCP structure of the ensuing NLP and QP subproblems. This structure is well-suited for the employment of efficient, structure-exploiting solvers. Accordingly, we form the following augmented system based on the delta formulation:

$$\tilde{x} = \begin{bmatrix} p_{ca} & p_{im} & p_{om} & p_{O_2} & m_{im} & \omega_{cp} \\ I_{st} & U_{cm} & h_{om} & I_{bat} \end{bmatrix}^T \in \mathbb{R}^{10} \quad (28)$$

$$\tilde{u} = [\dot{I}_{st} \quad \dot{U}_{cm} \quad \dot{h}_{om} \quad \dot{I}_{bat}]^T \in \mathbb{R}^4 \quad (29)$$

The approach extends the state vector, denoted as \tilde{x} in the augmented state space, to include the actuated variables, while the input vector \tilde{u} comprises their rates of change. The implementation of this augmentation can vary. In this work, we choose to augment the system dynamics prior to discretization. The resulting prediction model is

$$\dot{\tilde{x}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{u}(t) \end{bmatrix} = \begin{bmatrix} f(x(t), u(t - \tau) + \tilde{u}(t) \cdot \tau, p(t)) \\ \tilde{u}(t) \end{bmatrix}, \quad (30)$$

where the augmented state $\tilde{x}(t)$ combines the current state $x(t)$ with the input from the previous timestep $u(t - \tau)$, while $\tilde{u}(t)$ denotes the input's rate of change $\dot{u}(t)$. Note that, from this point forward, the augmented state \tilde{x} and input vector \tilde{u} will be referred to simply as x and u , respectively.

4.2 Optimal control problem

The NMPC is defined by the OCP in (31). The objective function steers the outputs to its reference values $y_{ref}(t) \in \mathbb{R}^5$, starting from the initial state \hat{x}_0 , and penalizes input changes over the prediction time t_p . The last two terms minimize the violation of soft constraints, where $\epsilon \in \mathbb{R}^3$ is the vector of slack variables. These are penalized both quadratically and linearly to ensure exact penalty.

$$\begin{aligned} \min_{x(\cdot), u(\cdot), \epsilon} & \frac{1}{2} \int_0^{t_p} (\|y(t) - y_{ref}(t)\|_Q^2 + \|u(t)\|_R^2) dt \\ & + \|\epsilon\|_{S_q}^2 + |\epsilon|_{S_l} \\ \text{s.t.} & \quad 0 = x(0) - \hat{x}_0, \\ & \quad \dot{x} = f(x(t), u(t), p(t)), \quad \forall t \in [0, t_p], \\ & \quad 0 \geq h(x(t), u(t), p(t), \epsilon), \quad \forall t \in [0, t_p]. \end{aligned} \quad (31)$$

To account for actuator limits, the inequality function h imposes specific input constraints as part of the OCP:

$$\begin{aligned} I_{st}^{\min} &\leq I_{st}, \\ h_{om}^{\min} &\leq h_{om} \leq h_{om}^{\max}. \end{aligned}$$

The lower bound on the stack current I_{st}^{\min} is established to avoid operation near OCV, which is a known catalyst and membrane stressor [13]. Additionally, we constrain the rates of change of the inputs to reflect unmodeled actuator dynamics:

$$u_{\min} \leq u \leq u_{\max}$$

These actuator constraints are enforced strictly as hard constraints, allowing no violations. In contrast, safety and degradation concerns are addressed through soft constraints with non-negative slack variables ϵ to ensure feasibility. Compressor safety is ensured by the following:

$$\begin{aligned} 13.125 \cdot \dot{m}_{cr} + 0.82 - \Pi &\leq \epsilon_{choke} \\ \Pi - 278.6852 \cdot \dot{m}_{cr}^2 - 17.8817 \cdot \dot{m}_{cr} - 0.7888 &\leq \epsilon_{surge} \end{aligned}$$

Moreover, to maintain adequate oxygen supply to the cathode channels, OER λ_{O_2} is constrained to not fall below a minimum value, $\lambda_{O_2}^{\min}$. Since $\lambda_{O_2} = f(I_{st}^{-1})$, see (17), we implement an equivalent linear formulation of this constraint to avoid the introduction of unnecessary nonlinearity:

$$\lambda_{O_2}^{\min} \cdot \dot{m}_{O_2}^{\text{react}} - \dot{m}_{O_2,im}^{\text{out}} \leq \epsilon \lambda_{O_2}$$

4.3 Nonlinear program

In direct optimal control methods, the continuous-time OCP in (31) is discretized, resulting in a finite-dimensional NLP. Here, multiple shooting is applied and the NLP reads

$$\begin{aligned} \min_{\substack{x_0, \dots, x_{N-1}, \\ u_0, \dots, u_{N-1}, \\ \epsilon}} \quad & \frac{1}{2} \sum_{k=0}^{N-1} c_k + \|\epsilon\|_{S_q}^2 + |\epsilon|_{S_1} \\ \text{s.t.} \quad & 0 = x_0 - \hat{x}_0, \\ & x_{k+1} = F(x_k, u_k, p_k), \quad \forall k \in \mathcal{H}, \\ & 0 \geq h(x_k, u_k, p_k, \epsilon), \quad \forall k \in \mathcal{H}, \end{aligned} \quad (32)$$

with $c_k = \|y_k - y_{\text{ref},k}\|_Q^2 + \|u_k\|_R^2$ and $\mathcal{H} = \{0, \dots, N-1\}$. The computational efficiency and effectiveness of the solution obtained from the NLP hinge on key design choices in the transition from OCP to NLP. Here, we discuss the choice of the prediction horizon N , the integration method with its step size, and the sampling time $T_s = t_p/N$.

4.3.1 Sampling time

The controller's sampling time T_s defines its maximum bandwidth. To determine the most suitable T_s , we simulate the system and linearize it at various operating points. This approach involves an eigenvalue analysis of the resulting system matrices, from which the reciprocals of these eigenvalues provide the respective time constants for the system's modes. The analysis across varying power demands revealed time constants between 0.0096 s and 1.1015 s. Consequently, we select a sampling time of $T_s = 10^{-2}$ s for our internal model, ensuring it aligns with the system's fastest dynamics.

4.3.2 Numerical integration scheme

To identify a suitable integration scheme, we analyze the trade-off between discretization error and computational cost for various schemes. To this end, we apply both explicit (ERK) and implicit Runge-Kutta methods, specifically Gauss-Legendre and Radau IIa, to the prediction model, each across a range of integration steps from 1 to 10. For each scheme, we calculate the overall relative discretization error e_{rel} and monitor the computation time. This approach allows us to establish a Pareto front, which helps us identify the most efficient schemes based on accuracy and computational load. For instance, the explicit schemes, when limited to a single integration step, dominate for accuracies of $e_{\text{rel}} \geq 10^{-6}$. Implicit schemes only begin to dominate on the Pareto front at higher accuracy levels. Considering the trade-off between accuracy and computational cost, we opt for the standard ERK with stage four, i.e. ERK4 scheme, for its satisfactory accuracy of $\mathcal{O}(10^{-6})$ and computational efficiency among the schemes evaluated.

4.3.3 Prediction horizon

The selection of the prediction horizon N in MPC is crucial as it defines the extent to which the controller anticipates future events. Ideally, N should align with the system's slowest dynamics to allow the MPC to adequately account for the impact of its control actions. Our analysis in Sec. 4.3.1 indicates that a prediction horizon of $N = 110$ would match the slowest system dynamics. However, longer horizons increase the size of the optimization problem, affecting execution time and memory usage. Through evaluating the relative cost difference w.r.t. to a horizon of $N = 130$ and the mean computation time for solving the optimization problem in closed-loop for various horizons, we established the specific trade-off between cost and computational efficiency. Balancing these factors, we select a prediction horizon of $N = 35$, which keeps the cost increase under 1 % and has a low computation time in comparison to the tested horizons.

4.4 Sequential quadratic programming

An SQP method applied to the NLP in (32) proceeds in each iteration by solving the following structured QP subproblem:

$$\begin{aligned} \min_{\substack{\Delta x_0, \dots, \Delta x_{N-1}, \\ \Delta u_0, \dots, \Delta u_{N-1}}} \quad & \frac{1}{2} \sum_{k=0}^{N-1} \left(\begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}^\top H_k \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + g_k^\top \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} \right) \\ \text{s.t.} \quad & 0 = \Delta x_0 - \Delta \hat{x}_0, \end{aligned}$$

$$\begin{aligned} \Delta x_{k+1} &= A_k \Delta x_k + B_k \Delta u_k, \quad \forall k \in \mathcal{H}, \\ 0 &\geq C_{k,x} \Delta x_k + C_{k,u} \Delta u_k, \quad \forall k \in \mathcal{H}, \end{aligned}$$

where $H_k = \Delta_{v_k}^2 c_k$ and $g_k = \Delta_{v_k} c_k$ with $v_k = [x \ u]^\top$. $A_k = \frac{\partial F}{\partial x_k}(x^j)$, $B_k = \frac{\partial F}{\partial u_k}(u^j)$, $C_{k,x} = \frac{\partial h}{\partial x_k}(x^j)$ and $C_{k,u} = \frac{\partial h}{\partial u_k}(u^j)$ denote the constraint Jacobian matrices. At its core, the SQP algorithm successively linearizes the NLP around the current iterate v^j , solves the QP subproblem and updates the current iterate by $v^{j+1} = v^j + \alpha \Delta v$. Note that, two simplifications are made here: (i) slack variables are not implemented and (ii) the Hessian is the second derivative of the cost functional c_k instead of the Lagrangian. In addition, we apply the SQP method that employs the Gauss-Newton (GN) Hessian, i.e. $H_k = \frac{\partial r_k}{\partial v_k}(v^j)^\top \frac{\partial r_k}{\partial v_k}(v^j)$,

$$\text{with } r_k = [\sqrt{Q}(y_k - y_{\text{ref},k}) \ \sqrt{R}u_k]^\top.$$

In both cases one SQP iteration is performed per sampling instant and the solution is shifted to obtain an initial guess for the next instant. A maximum number of six QP iterations is allowed where the step size α is set to 0.5.

The algorithm is implemented in MATLAB/Simulink [14] using the symbolic framework CasADi [15]. The algorithm is then run on the embedded platform, which is a dSPACE MicroLabBox with an NXP (Freescale) QorIQ P5020, dualcore, 2 GHz, 1 GB DRAM processor. The QPs are solved using the structure-exploiting QP solver from HPIPM [16].

5 Simulation results

We evaluate the proposed control scheme in a step response scenario characterized by rapid, substantial changes

in power demand, as illustrated in Fig. 3a. Closed-loop simulations are performed against the prediction model, employing three solution methods for the low-level NMPC: (i) solving the NLP with IPOPT [17], (ii) applying the SQP method, and (iii) applying the SQP method with GN Hessian approximation. The tracking performance and constraint compliance of these methods are depicted in Fig. 3 and Fig. 4, respectively.

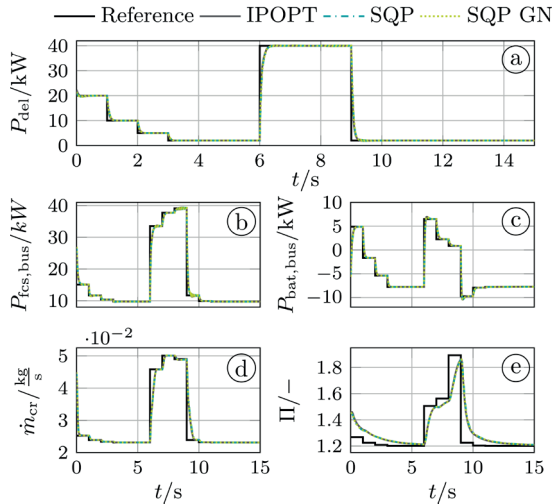


Fig. 3 Tracking of power references and compressor setpoints.

All employed methods demonstrate effective and sufficiently dynamic output tracking behavior, with the control errors converging towards zero offset for all outputs, as anticipated in the absence of model mismatch.

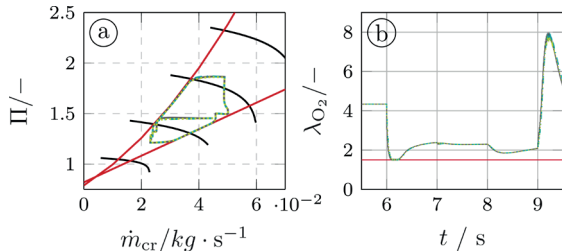


Fig. 4 Compressor choke and surge (a) and OER constraint (b).

The step response scenario poses a significant challenge for constraint compliance. Yet, while all controllers closely approach the surge, choke, and OER constraints for optimality, they do so without violating them. Despite the SQP methods not incorporating slack variables, and their underlying Hessians omitting constraint considerations, the closed-loop control results achieved by SQP are very close to those obtained using IPOPT. In fact, the disparity in the output and constraint trajectories between them is on the order of $\mathcal{O}(10^{-3})$. The mean (worst-case) turnaround times on the dSPACE MicroLabBox are 11.2 ms (12.2 ms) and 10.3 ms (11.4 ms) for the SQP method and SQP method with GN Hessian, marginally exceeding the 10 ms sampling time and thus not fully meeting the real-time capability requirement.

6 Conclusions

We presented an NMPC design that effectively formulates the control problem relevant to the air-path control in a

FCHV as an OCP. This is efficiently transcribed into an NLP, considering computational cost and solution quality. The resulting algorithm, utilizing different solution methods, proves capable of accurate power tracking and compliance with safety constraints. We evaluate the computational performance of two different solution methods on embedded hardware, finding that the SQP method with a GN Hessian outperforms the alternative. To further enhance computational efficiency, future work might explore the use of condensing algorithms or different formulations of the prediction model.

7 Literature

- [1] O. Z. Sharaf and M. F. Orhan, "An overview of fuel cell technology: Fundamentals and applications," *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 810–853, 2014.
- [2] W. R. W. Daud, R. E. Rosli, E. H. Majlan, S. A. A. Hamid, R. Mohamed, and T. Husaini, "PEM fuel cell system control: A review," *Renewable Energy*, vol. 113, pp. 620–638, 2017.
- [3] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," *Nonlinear Model Predictive Control*, pp. 391–417, 2009.
- [4] R. Quirynen, "Numerical Simulation Methods for Embedded Optimization," KU Leuven, 2017.
- [5] J. Luna, S. Jemei, N. Yousfi-Steiner, A. Husar, M. Serra, and D. Hissel, "Nonlinear predictive control for durability enhancement and efficiency improvement in a fuel cell power system," *Journal of Power Sources*, vol. 328, pp. 250–261, 2016.
- [6] V. Neisen, J. Mannhardt, and D. Abel, "Dynamic Tracking of Power Demand for Integrated Fuel Cell Systems using Nonlinear Model Predictive Control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 13216–13223, 2020.
- [7] L. Schmitt and D. Abel, "Nonlinear MPC for Fuel Cell Air Path Control with Experimental Validation," in *2023 31st Mediterranean Conference on Control and Automation (MED)*.
- [8] L. Schmitt and D. Abel, "Numerical Integration for Nonlinear Model Predictive Control of a Fuel Cell System," in *2023 American Control Conference (ACC)*.
- [9] L. Schmitt, N. Nickig, M. Bahr, S. Gößling, and D. Abel, "Review, Evaluation and Application of Condensing Algorithms for Model Predictive Control based on a First-Order Method," in *2023 European Control Conference (ECC)*.
- [10] K. W. Suh, "Modeling, analysis and control of fuel cell hybrid power systems," University of Michigan, 2006.
- [11] J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng, *Control of Fuel Cell Power Systems: Principles, Modeling, Analysis and Feedback Design*. Springer Science & Business Media, 2004.
- [12] O. Tremblay, L.-A. Dessaint, and A.-I. Dekkiche, "A Generic Battery Model for the Dynamic Simulation of Hybrid Electric Vehicles," in *2007 IEEE Vehicle Power and Propulsion Conference*.
- [13] A. Goshtasbi and T. Earsal, "Degradation-conscious control for enhanced lifetime of automotive polymer electrolyte membrane fuel cells," *Journal of Power Sources*, vol. 457, p. 227996, 2020.
- [14] The MathWorks Inc., *MATLAB* (2023). Natick, Massachusetts, United States: The MathWorks Inc.
- [15] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Math. Prog. Comp.*, vol. 11, no. 1, pp. 1–36, 2019.
- [16] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.
- [17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, 2006.

Time-Triggered Organic Computing Architecture for Autonomous Driving Vehicles Using List Scheduling

Mario Qosja, Simon Meckel, Roman Obermaisser

Chair for Embedded Systems, University of Siegen, Siegen, Germany

mario.qosja@uni-siegen.de, simon.meckel@uni-siegen.de, roman.obermaisser@uni-siegen.de

Abstract

As the autonomous vehicles market is expected to grow in the future, their functionalities will increase too, leading to complex embedded computer systems. To address this, organic computing has emerged as a research area that takes inspiration from biological entities to handle complex distributed embedded computer systems. Organic computing has improved adaptability and robustness, while also reducing development efforts. However, it has some drawbacks in terms of determinism, composability, and dependability, which are key features for safety-critical applications. Distributed computer systems using time-triggered communication networks possess these characteristics and thus show distinct advantages for safety-critical systems. By combining artificial DNA and hormone features with time-triggered communications, we can make these systems safer, more reliable, and suitable for safety-critical applications. Therefore, we present a time-triggered organic computing architecture where the time-triggered schedule is produced by a list scheduling algorithm during run-time. All the tasks on the systems are executed according to the computed schedule. To demonstrate the concepts and the system model, we performed evaluation test in a simulator in which the system executes tasks according to predefined schedules. The evaluation of use cases shows improved temporal predictability and fault containment.

1 Introduction

The market demand for autonomous vehicles has increased in recent years and is expected to grow exponentially in the future. This increase in demand will result in vehicles with complex functionalities, leading to challenges in their handling, especially in fault situations. In current designs, the human driver provides adaptability and flexibility in challenging driving situations, so driver assistance functions do not need to be fail-operational. As we move towards self-driving vehicles, more and more driver assistance functions will need to be fail-operational, i.e., they will need to provide system services even in the event of faults (such as failure of computing nodes or sensors). Therefore, we need computing systems that can dynamically and autonomously adapt to these complex situations and provide sufficient redundancy at a limited cost. IBM defines systems as autonomous if they contain self-x properties (such as self-organization, self-healing, self-configuration) [9]. Organic Computing (OC) is a paradigm for organizing distributed computer systems with a high degree of flexibility and self-healing, which is inspired by the concepts and principles of biological systems. It adapts the working principles of organic systems to manifest their self-organizational nature into complex embedded systems. This is done by employing an Artificial Hormone System (AHS) as a real-time middleware that brings improved adaptability and robustness by exhibiting a self-organizing mechanism that can self-configure and heal the system. This is achieved by exchanging artificial hormones (i.e., small messages) between all computing nodes in a distributed system to determine the suitability of task allocations, initially in the start-up phase (self-configuration), in the event of node failures (self-healing) and after potentially degrading system services (reconfigurations). However, the AHS currently lacks support for dependability, determinism, and composability, which are crucial for safety-critical

systems such as autonomous driving systems. On the other hand, this is provided by time-triggered systems. Such systems ensure resource adequacy and predictability through a priori scheduled tasks and messages. Knowledge of the permitted temporal behavior of components allows effective fault containment in the time domain and simplifies certifiability. In addition, in this paper a Time-Triggered Organic Computing (TTOC) architecture for the automotive domain has been presented that combines the advantages of both organic computing and time-triggered systems. The flexibility in terms of task (re-)allocations is maintained by the artificial hormone system and the predictability is realized by a list scheduler that will organize task execution times, message injection times, and message paths. Furthermore, it deals with the distributed scheduling problem in the TTOC environment for each computing node. In typical designs, the scheduler organizes the temporal and spatial allocation of both tasks and messages, but in the proposed architecture the AHS middleware handles the allocation of application tasks to ECUs meanwhile the scheduling algorithm will handle execution times and message paths. The paper is divided as follows: In section 3 it provides a short introduction to the specifics of OC and gives an overview of the combination of time-triggered concepts with OC. Section 4 will discuss the scheduling problem for the TTOC and the newly proposed algorithm designed by us. Section 5 describes the evaluation of concepts that have been proposed using the TTOC simulator. Section 6 draws the conclusion and the future work.

2 Related Work

Self-X properties of autonomous systems have been an area of research that has received significant attention over the past few years. Organic Computing was established as a

research field by Deutsche Forschungsgemeinschaft (German National Science Foundation) in 2003 [8] to bring the principles of biological systems into distributed computer systems. A distributed self-organization OC based on the observer/controller architecture is introduced in [12] with the ability to control unexpected behaviors of the system. Another observer/controller design is developed in [3]. This OC system is tested on a traffic light controller. In addition, there have been OC constructed from different approaches. Similar to how genetic instructions encode the functioning and growth of organisms, ADNA can also encode the structure and organization of embedded computer systems [4] and store it inside each computing node (ECU), following the same principles as biological DNA. An additional, organic computing technology inspired by biological systems has been established at [7]. It is a real-time middleware based on an artificial hormone system. This middleware exhibits self-organizing property, allocating tasks to the most suitable processing node by itself. AHS uses artificial DNA to construct distributed embedded systems.

Furthermore, significant research has been conducted to introduce self-organization and OC properties in the automotive domain. OC will handle the increasing complexity of embedded systems in the autonomous vehicle sector. The dynamic concepts of ADNA and AHS can be used in AUTOSTAR [6]. Moreover, organic computing has also been used to improve the dependability of automotives as described in [10]. About the middleware layer, an autonomic middleware for automotive embedded systems that exhibits high flexibility and automatic runtime reconfigurations is presented in [1]. An alternative middleware approach [2] dynamically configures automotive embedded systems by providing transparency and flexible platform-independent support for portability.

The proposed architecture combines time-triggered concepts with organic computing, based on artificial hormone system middleware for autonomous vehicles. This time-triggered OC middleware achieves high flexibility and reliability that comes from the self-x properties and predefined task execution. The schedules get dynamically calculated for each computing node by a heuristic scheduling algorithm.

3 Time-Triggered Organic Computing Architecture

We have taken inspiration from the concept of ADNA and AHS, two organic computing technologies following the same philosophy as biological systems. ADNA and AHS offer several advantages, such as robustness, reduced development efforts, and increased adaptability. In section 1, we proposed combining ADNA and AHS with time-triggered concepts to make embedded systems more deterministic and reliable. We call this new architecture Time-Triggered Organic Computing (TTOC). To begin with, in this section, we provide a brief overview of ADNA and AHS, followed by a detailed explanation of the new architectural concepts.

3.1 Artificial DNA

Different complex embedded systems can be constructed by inserting the structure and organization of the system in a single file and storing it in each computing node [5]. The functionalities (e.g., task structures and messages) are encoded in the ADNA file using simple basic elements.

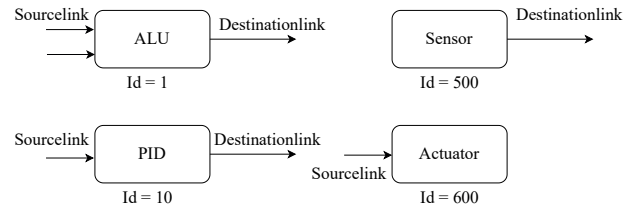


Figure 1 Different basic elements

Figure 1 exemplarily shows functional basic elements of an embedded system (e.g., filters, actuators, sensors) where *SourceLink* denotes a reactive link that responds to incoming requests and *DestinationLink* is an active link for sending requests. In the automotive domain, a basic element can also be, for example, the ABS functionality. In the current designs, the ABS is bound to a fixed ECU and a backup one. In case of a failure in both ECUs, the system loses the ABS functionality, which may lead to wheels locking up during breaking. With the ADNA file located in each processor core, the ABS functionality will be transferred to other active ECUs, thus preventing the lockup of the wheels.

3.2 Artificial Hormone System

The AHS middleware is designed to read the ADNA computer file and create system functionalities. It can also implement other self-x features, such as self-configuring, where the system reconfigures itself during run-time. Once the tasks are created, AHS will allocate them to specific computing nodes based on their suitability, organizing the system (self-organization) accordingly. For example, in control loops, tasks from the PID controller will have higher suitability on computing nodes that perform arithmetic calculations better.

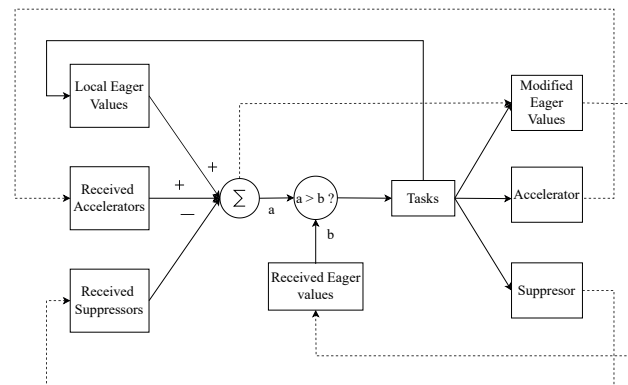


Figure 2 Hormone Loop of AHS

Furthermore, AHS also distributes tasks to minimize communication distances (self-optimizing). Each ECU ex-

ecutes an hormone loop for sending hormones and calculating task suitability from these hormones. Each task contains three main hormone types: *Suppressors* for lowering suitability, *Accelerators* for increasing suitability, and *Eager values* for determining task suitability of the ECU. The hormone loop shown in **Figure 2** sums up all received accelerators, suppressors, and local eager values, and compares these results with received eager values. AHS advantages rely on self-x properties such as self-organization, self-building, self-optimization, and self-healing. In AHS, the system is self-organized because task distribution is done internally by exchanging hormones between the nodes, taking into consideration the suitability of tasks but also the load of a node. The system becomes more optimized by preventing electronic control units from experiencing high loads. The entire system is built using simple artificial DNA files, which allows for reconfiguration in the event of computing node failure. In such cases, all tasks are migrated to other computing nodes.

3.3 Time-Triggered Architecture

Distributed embedded computer systems are built using nodes consisting of communication controllers and computer hosts that communicate via time-triggered networks. In the TTOC, each ECU will serve as one node of the system as presented in **Figure 3**.

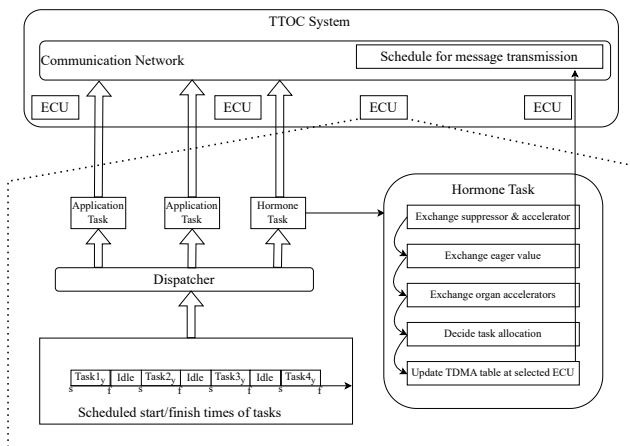


Figure 3 Time-Triggered Organic Computing Architecture

In organic computing based on ADNA, different ECUs concurrently perform various tasks, such as sensor or actuator tasks. In distributed embedded systems, ensuring a guaranteed consistent system behavior is crucial. This is achieved by processing events on all nodes in the same consistent order. Using a global time base to execute operations ensures the same order for all nodes, which brings determinism to the system. Being predictable makes the systems deterministic, thereby increasing overall reliability. The global time is also utilized for error detection, communication protocols, and interfaces of the nodes. For instance, in the case of a babbling idiot failure, where a computing node sends untimely messages, all nodes have predefined times when they can communicate. If communication occurs outside the specified time, bus guardians will block the messages that occur outside of the predefined time slots.

In TTOC, any communication network that supports time-triggered concepts is suitable. At the operating system layer, each ECU has a task dispatcher and a time-triggered schedule. The dispatcher is responsible for reading the schedule and executing different tasks at particular points in time. A schedule table with the start and finish execution times of the tasks is shown in Figure 3. In the architecture, alongside the application tasks from the application model, such as sensor data, there are also the TTOC middleware tasks composed of hormone message exchange, which is also time-triggered. The ECU can also be in an idle stage, where it waits for the next task to be processed.

4 Scheduling

There are two types of real-time applications, hard and soft. If the failure of meeting the deadline causes a fatal fault this is called a hard deadline. It is called soft where missing the deadline will not have a big impact on the application. For autonomous vehicle systems, if the deadlines are missed, the caused failure can lead to vehicle malfunction. To be able to ensure that all application tasks will meet their deadlines, real-time scheduling must be performed. The scheduling algorithm determines the order of the tasks that are going to be processed by the computing system.

There are two types of scheduling algorithms: static and dynamic. If the task priorities and the execution times are determined before the program starts, the scheduling is static. In dynamic scheduling, everything must be calculated during the run-time of the system. The scheduler comes up with a scheduling algorithm for the system, which is in two forms: preemptive or non-preemptive. In the preemptive form, if a task with a lower priority has blocked one with a higher one, the process with lower priority will be terminated, allowing the execution of the higher one. In the non-preemptive the process does not terminate but it waits until the CPU burst time is complete. In classical designs, the scheduler organizes:

- Allocation of application tasks to computing nodes
- Application task execution times
- Message injection times
- Messages paths

In the proposed architecture, this process is split and performed incrementally. The AHS middleware first determines the allocation of tasks (of the application) in the system at run-time, without execution times, thus allowing for the desired flexibility that saves hardware cost, yet realizing the same level of redundancy. The list scheduling algorithm in TTOC then dynamically calculates the tasks execution times along with the message injection times and paths for the application messages, according to the given tasks assignments. The hormone exchange middleware tasks that run on each ECU are also time-triggered, but their timing within the schedule period is predefined.

4.1 List Scheduling

As mentioned, in TTOC the scheduling algorithm must be computed at each ECU dynamically without knowing the priorities of tasks. The dynamic scheduling algorithm that has been taken into consideration for TTOC is List Scheduling (LS). List scheduling is a process used to schedule tasks represented as a directed acyclic graph (DAG).

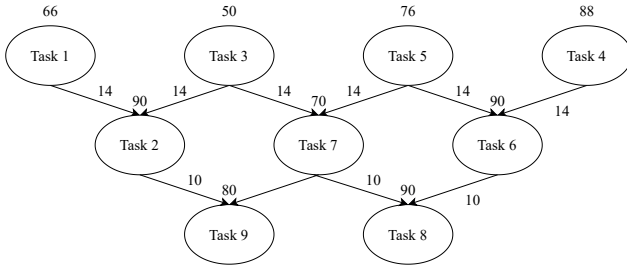


Figure 4 Example of DAG graph

There is a DAG graph consisting of nine application tasks and their worst case execution times (wcet) alongside message size as shown in **Figure 4**. The tasks are arranged based on their depth in the graph and the time required for their execution. Each task is represented by a node and dependencies between them are represented by edges in the graph. The scheduling algorithm determines the order in which the tasks should be executed and aims to assign each task to an available computing unit (ECU). It takes into consideration the data dependencies, timing constraints, system model, critical path, scheduling priorities, and optimizing for factors such as load balancing together with minimizing idle time. Since the spatial allocation is taken care of by TTOC middleware, the AHS will start processing the hormone loop tasks based on their execution order. For that reason, it is important to analyze and set the execution order of tasks.

In our above DAG graph (Figure 4), task two is data dependent from task one and task three because the messages that arrive from them are needed for its computations. In this case, task two must wait for the messages coming from task one and task three before it can start executing. Task one and task three are denoted as immediate parents of task two. For the timing constraints, as it is written in the entry of the section, in our case our system must meet hard deadlines. For the system model, Figure 3 describes that each ECU will execute the scheduling algorithm and produce its own schedule. In addition, the critical path in the DAG is calculated by taking into consideration the computation time (WCET) of ECU and the communication times (message size) on that particular path. Hence, we start processing tasks from the most critical path of the DAG. The b-level has been used to determine the node levels and scheduling priority.

After the tasks are ordered, the AHS can start the hormone loop to determine the suitability levels of the task and assign it to the most suitable ECU. Only then the scheduling algorithm can produce the temporal allocation of the application tasks and the (spatial + temporal) allocation of messages. The algorithm below describes the procedure for

schedule generation of each task. As an input, the algorithm will take the application task that has to be scheduled. As stated, the TTOC middleware tasks will also be executed in a time-triggered manner, which is why they will have predefined slots on the schedule in each period. The period of these fixed slots is denoted as the TTOC period and the whole period as the hyper period. This means that all the application tasks and messages will need to be scheduled after the TTOC period, otherwise the AHS will not allocate application tasks to ECU and the whole system will malfunction. For this reason, in the algorithm, it is constantly checked if the start time, or the start time plus the width of the application tasks, does not overlap with the TTOC period.

Algorithm Schedule(ECU)

```

Data: Task to be scheduled
for all tasks parents do
  for Parent sending messages do
    if Message receivers Id == Task ID
      then
        if Tasks run on same ECU then
          Message schedule generation;
          Update TDMA table of ECU ;
        end
        if Tasks run on different ECU then
          Message schedule generation;
          Update TDMA table of ECU;
        end
      end
    end
  end
end
if Task is not scheduled then
  Task start time == ECU time ;
  if Task is the first one to be scheduled then
    Task schedule generation;
    Update TDMA table of ECU;
  end
  if Task start time < hyper period + TTOC
  period then
    start time = hyper period + TTOC
    period;
    Task schedule generation;
    Update TDMA table of ECU;
  end
  if Task start time + task wcet > hyper
  period + TTOC period then
    start time = 2 * hyperperiod +
    TTOCperiod ;
    Task schedule generation;
    Update TDMA table of ECU;
  end
end
end

```

Furthermore, the complexity increases since the schedule is calculated for one task at a time. Since the allocation of the tasks that receive messages from the task that is being scheduled is not present till the AHS assigns it on an ECU, only the temporal allocation can be calculated.

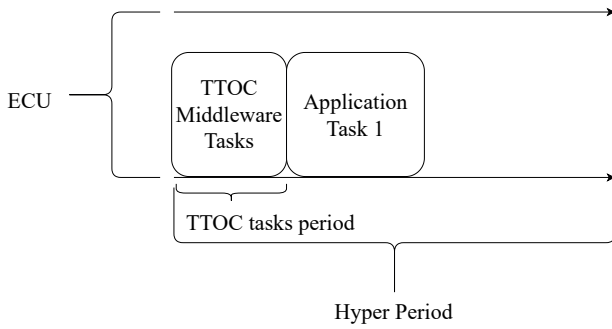


Figure 5 Schedule generation example per each ECU

The solution proposed is to first schedule all the incoming messages from all the task parents and then continue with the application task schedule. **Figure 5** represents the schedule generation phase for each ECU that happens dynamically while the system is running.

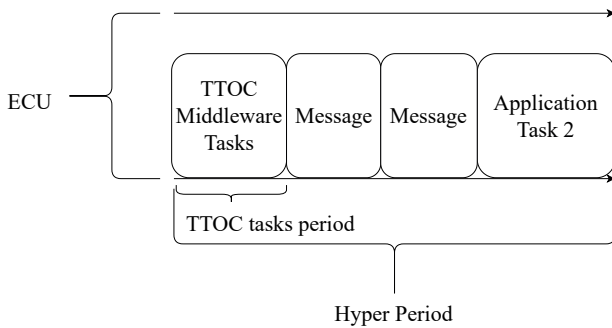


Figure 6 Schedule generation example per each ECU

5 Evaluation of use cases

In the evaluation part, we tested the new scheduling algorithm, the time-triggered concepts of TTOC, and the self-x properties of OC. The simulations were conducted on the TTOC simulator developed by [11]. In this system, the ECUs are implemented as processing elements that contain a task dispatcher, which utilizes a schedule to execute AHS middleware tasks or dummy application tasks.

5.1 Testing List Scheduling

For the testing of a list scheduling algorithm, the TTOC simulator has been used. In this simulator, automotive application tasks are represented in the form of dummy tasks. For the schedule calculation, multiple JSON files, that contained different applications and physical models, were used. The range of application models varies to examples up to forty tasks and with a total of fifty messages. All the results, from the ordering of tasks, spatial and temporal allocations, and execution times of the dispatcher are dumped into a log file. The analysis of the log file concludes that the schedule is generated dynamically as intended on each ECU.

5.2 Testing Time-Triggered Concepts

For the time-triggered concepts, we are interested in the system executing application tasks and communication of messages, according to the generated table schedules. As stated in this section, in the TTOC simulator each ECU contains a task dispatcher that operates based on the time-triggered schedule. To evaluate the reliability of the TTOC architecture, babbling idiot failure was tested on the system. A random ECU was chosen to send untimely communication messages to the other ECUs. The results showed that the failed ECU was blocked from transmission outside its predefined communication schedule.

5.3 Testing Self-X Properties

We want the time-triggered organic computing to exhibit also the self-x properties. By creating artificial DNA files, that describe embedded systems in the automotive domain, and feeding these files into the TTOC simulator, it is possible to test the self-building property. Furthermore, with the AHS properties residing in the simulator the self-organization and self-configuration of the middleware can be examined. Here, we are particularly interested in the case of failures of ECUs, if the functionalities of the ECU will be transferred to the other ones, and in the system reconfiguration. In the simulator, it is possible to create timed events that denote ECU failures. By implementing this feature, we observed that right after the ECU was in a failure state, the system reconfigured and reorganized itself, with all the tasks distributed to the remaining ECUs.

6 Conclusion and Future Work

In this paper, we propose the building of a scheduling algorithm that will serve to calculate table schedules during run-time for the newly proposed time-triggered organic computing architecture. TTOC architecture is based on the combination of self-x properties residing in ADNA and AHS with time-triggered techniques to improve the reliability, determinism and safety of embedded systems in autonomous driving vehicles. The TTOC architecture enhances the temporal predictability of the system by ensuring that all tasks are executed at specific times. In addition, the simulation results performed on the TTOC simulator showed that the functionalities of ADNA and AHS remained unchanged.

For our future work, we are planning on testing the communication network between ECUs in the network simulator called OMNet++. The selected time-triggered network is Time Sensitive Network (TSN). The interaction between the network simulator and TTOC process will be handled by a new designed co-simulation controller. Furthermore, we want to test our new architecture with an autonomous driving vehicle simulator like CARLA. In this case we can test our architecture with real autonomous driving application tasks.

Acknowledgment

This work was supported by research project SelfAutoDOC funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK).

References

- [1] Richard Anthony, DeJiu Chen, Martin Törngren, Detlef Scholle, Martin Sanfridson, Achim Rettberg, Tahir Naseer, Magnus Persson, and Lei Feng. Autonomic middleware for automotive embedded systems. *Autonomic Communication*, pages 169–210, 2009.
- [2] Richard Anthony, Paul Ward, DeJiu Chen, Achim Rettberg, James Hawthorne, Mariusz Pelc, and Martin Törngren. A middleware approach to dynamically configurable automotive embedded systems. 5 2010.
- [3] Jürgen Branke, Moez Mnif, Christian Müller-Schloer, Holger Prothmann, Urban Richter, Fabian Rochner, and Hartmut Schmeck. Organic computing—addressing complexity by controlled self-organization. In *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (isola 2006)*, pages 185–191. IEEE, 2006.
- [4] Uwe Brinkschulte. An artificial dna for self-describing and self-building embedded real-time systems. *Concurrency and Computation: Practice and Experience*, 28(14):3711–3729, 2016.
- [5] Uwe Brinkschulte. Technical report: Artificial dna—a concept for self-building embedded systems. *arXiv preprint arXiv:1707.07617*, 2017.
- [6] Uwe Brinkschulte, Eric Hutter, and Felix Fastnacht. Adapting the concept of artificial dna and hormone system to a classical autosar environment. In *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*, pages 35–42. IEEE, 2019.
- [7] Uwe Brinkschulte, Mathias Pacher, and Alexander von Renteln. An artificial hormone system for self-organizing real-time task allocation in organic middleware. In *Organic Computing*, pages 261–283. Springer, 2009.
- [8] DFG Schwerpunktprogramm 1183 Organic Computing, 2005–2011. <https://gepris.dfg.de/gepris/projekt/5472210>.
- [9] Jeffrey O Kephart and David M Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.
- [10] Timo Kisselbach, Simon Meckel, Mathias Pacher, Uwe Brinkschulte, and Roman Obermaisser. Organic computing to improve the dependability of an automotive environment. In *International Conference on Architecture of Computing Systems*, pages 211–225. Springer, 2022.
- [11] Mario Qosja, Simon Meckel, and Roman Obermaisser. Simulator for time-triggered organic computing. *Procedia Computer Science*, 220:127–134, 2023.
- [12] Urban Richter, Moez Mnif, Jürgen Branke, Christian Müller-Schloer, and Hartmut Schmeck. Towards a generic observer/controller architecture for organic computing. In Christian Hochberger and Rüdiger Liskowsky, editors, *INFORMATIK 2006 – Informatik für Menschen, Band 1*, pages 112–119, Bonn, 2006. Gesellschaft für Informatik e.V.

Robust Navigation of Autonomous Transport Units in the Extractive Industry*

David Benz¹, Dirk Abel¹

Abstract—Autonomous driving in temporarily GNSS-denied environments is challenging. Vehicle controllers of autonomous transport units require continuously precise information about vehicle position, speed, and heading. To cope with areas with no satellite signal reception, we introduce a multi-sensor navigation filter for articulated vehicles that fuses measurements from an inertial measurement unit (IMU), global navigation satellite systems (GNSS), wheel encoders, an optical speed sensor, and a barometer. Non-holonomic vehicle constraints are considered as well in the state estimation. The navigation filter uses two unscented Kalman filters (UKF) with a global fusion of the locally estimated states. This approach achieves improved robustness regarding single sensor failures compared to a centralized integration of all sensors in one filter. The developed navigation filter is evaluated experimentally with an articulated dumper in a gravel pit. With the proposed method, we achieved a mean position error of 0.19 m during a 190 s test drive in a gravel pit with a simulated GNSS interruption of 90 s.

I. INTRODUCTION

The production of renewable energy generation equipment requires various mineral raw materials and therefore demands a massive expansion of the mineral resource extraction. One way to meet the increasing demand for mineral resources is to automate the processes involved in the extraction. With autonomous transportation units, material transport could be automated. Autonomous driving requires continuous information about the vehicle's position, speed, and heading. GNSS is commonly utilized to calculate this information. In open-pit mines, however, the reception of those signals is not guaranteed due to the deep funnel-shaped structure. One way to overcome this problem is to use a sensor fusion.

State-of-the-art navigation is done by fusing inertial measurements and GNSS measurements within a Kalman filter. Inertial measurements are processed within the prediction step using the strapdown algorithm [1]. Pseudoranges and deltaranges obtained from a GNSS receiver are used to correct the propagation. Under normal conditions, this approach leads to sufficient accuracy. However, without continuous GNSS measurements, the accuracy decreases within seconds as measurement noise of the inertial measurement unit (IMU) disturbs the strapdown algorithm. Integrating further sensors helps to overcome this drawback.

Odometry describes the use of information from the drive train to obtain changes in position and heading. Modern

odometry methods are lidar [2], radar [3] and visual odometry [4]. These approaches extract the movement information not from the drive train but from perception sensors. To achieve acceptable accuracy for autonomous driving, the algorithms require sufficient textures in the environment. Large open-pit mines cannot always provide this textured environment.

Wheel encoders are the most common way to measure movements [5]. They operate reliably in rough mining terrain but have one major drawback. Measurements from a slipping wheel will be misinterpreted within the filtering algorithm. Optical speed sensors on the other hand deliver odometry information as well but are not affected by slip as they measure the velocity over ground. However, due to the optical measuring principle, measurements on certain surfaces, such as puddles, can be incorrect [6].

In this work, we propose a federated fusion architecture with two unscented Kalman filters (UKF) and a following single-epoch fusing algorithm. This architecture was initially proposed by [7]. It allows the combination of several different filters. This means that not only different sensors can be used, but also the process models can vary. The prerequisite is that all local filters estimate at least one subset of the same states and provide a covariance matrix [8]. In our case, two state-of-the-art navigation filters fuse IMU, GNSS, and barometer measurements. One filter is further aided by measurements from an optical speed sensor and the other one by wheel odometry measurements. Both navigation filters were already successfully evaluated in [6] and [9]. We expect from the federated architecture a more robust state estimation.

This work is based on previous developments of our research group. In [10] and [11] was already shown that fusing inertial measurements with pseudo- and deltaranges from GPS and Galileo leads to a highly accurate state estimation. The benefits of GNSS integrity and differential corrections are shown in [12] and [13]. In [14], [6], [9], [15], the integration of optical measurements, the use of wheel odometry and the integration of nonholonomic constraints is explained, evaluated, and discussed in detail.

The remainder of this paper is structured as follows: Section II describes the underlying methods. Section III describes the hardware setup and evaluation protocol. In Section IV, experimental results are presented and discussed. A conclusion and outlook are given in Section V.

*This work was funded by the German Federal Ministry of Education and Research in the research project ARTUS (grant 033R126DN)

¹D. Benz and D. Abel are with the Institute of Automatic Control, RWTH Aachen University, 52074 Aachen, Germany
d.benz@irt.rwth-aachen.de

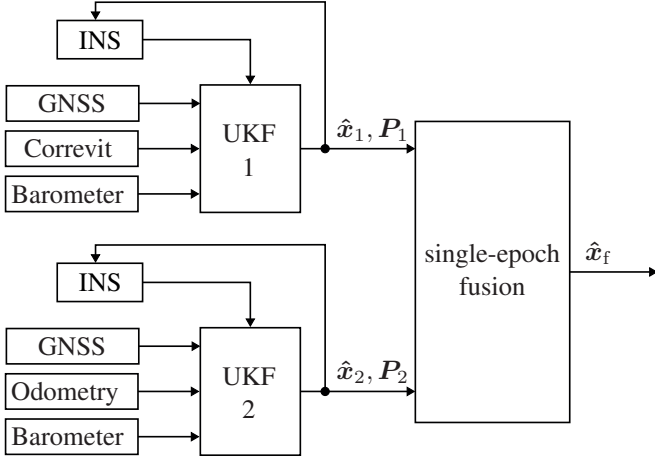


Fig. 1: Federated integration of IMU, GNSS, barometer, wheel odometry and optical speed sensor (Correvit) measurements (figure based on [8]).

II. METHODS

A. Navigation Filter Overview

The navigation filter utilizes the federated filtered integration architecture from [7] and [8], shown in Fig. 1. Each underlying navigation filter is implemented as UKF and estimates 18 states:

$$\mathbf{x} = [\mathbf{p}_{eb}^e \ \mathbf{v}_{eb}^n \ \mathbf{q}_b^n \ \mathbf{b}_a \ \mathbf{b}_g \ c_b \ c_d]^T. \quad (1)$$

The state vector consists of the three-dimensional position of the body frame origin (b) with respect to the Earth-Centered-Earth-Fixed (ECEF) frame (e) \mathbf{p}_{eb}^e , the three-dimensional velocity of the body frame with respect to the ECEF frame in navigation frame (NED) coordinates (n) \mathbf{v}_{eb}^n , the quaternion representing the alignment of the body frame to the navigation frame \mathbf{q}_b^n , the biases of the accelerometer triad \mathbf{b}_a , the biases of the gyroscope triad \mathbf{b}_g , the GNSS receiver clock bias c_b , and the clock drift c_d .

A schematic of UKF 1 is depicted in Fig. 2. UKF 2 is implemented accordingly. The filters are initialized with the position, velocity, and heading calculated by the GNSS receiver. Within the GNSS preprocessing, common-mode signal errors are corrected with reference station data received via mobile connection before feeding them into the filters, according to [13]. Further, an integrity check is performed according to [12].

B. Process Model

The navigation filter uses a 6-DoF nonlinear discrete-time state-space model to estimate the states:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k)\Delta t + \mathbf{x}_k + \mathbf{G}_k \mathbf{w}_k, \quad (2)$$

where k denotes the time step of length Δt . The matrices \mathbf{f}_k and \mathbf{G}_k are defined as follows:

$$\mathbf{f}_k = \begin{pmatrix} \mathbf{C}_{n,k}^e \mathbf{v}_{eb,k}^n \\ \mathbf{C}_{b,k}^n (\tilde{\mathbf{f}}_{ib,k}^b - \mathbf{b}_{a,k}) - (2\boldsymbol{\omega}_{ie,k}^n + \boldsymbol{\omega}_{en,k}^n) \times \mathbf{v}_{eb,k}^n + \mathbf{g}_{1,k}^n \\ \mathbf{q}_{b,k}^n \otimes \Delta \mathbf{q}_k \\ \mathbf{0} \\ \mathbf{0} \\ c_{d,k} \\ 0 \end{pmatrix},$$

$$\mathbf{G}_k = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{b,k}^n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{b,k}^n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{3 \times 3} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

The position \mathbf{p}_{eb}^e , velocity \mathbf{v}_{eb}^n , and the quaternion \mathbf{q}_b^n are predicted with measured accelerations $\tilde{\mathbf{f}}_{ib}^b$ and rotation rates $\boldsymbol{\omega}_{ib}^b$ and the strapdown algorithm according to [1]. The rotation matrices \mathbf{C} rotate the vectors between the different coordinate systems (b, e, n). The vectors $\boldsymbol{\omega}_{ie}^n$ and $\boldsymbol{\omega}_{en}^n$ are the Earth's rotation rate and the transport rate, respectively. The gravitation is compensated within the accelerometer measurements with the WELMEC model [16], which estimates the local gravity \mathbf{g}_1^n . The new orientation results from a change in angle $\boldsymbol{\omega}_{nb,k}^b \Delta t$, represented here as the quaternion $\Delta \mathbf{q}$.

The remaining states are predicted as follows: The IMU biases \mathbf{b}_a and \mathbf{b}_g are modeled as a random walk, according to [17]. The two GNSS receiver parameters clock error c_b and clock drift c_d are modeled as a first-order Gauss-Markov process, according to [18]. The accelerometer noise \mathbf{w}_{b_a} and gyroscope noise \mathbf{w}_{b_g} are given in the body frame. The shape matrix \mathbf{G} is therefore used to rotate the values into the navigation frame [9].

The propagation of the orientation is processed using orientation vectors within the filter, although the orientation is represented as a quaternion in the state vector. Therefore, $\mathbf{G}_{[3,2]} \mathbf{w}_g$ corresponds only to a three-dimensional matrix, although the quaternion in the state vector is four-dimensional. For better comprehensibility, the conversion is omitted here [19].

C. Measurement Models

1) *GNSS*: The GNSS update step processes pseudorange ρ_k^i and deltaranges d_k^i of GPS and Galileo from a single antenna. The measurement model is according to [13] and calculates these measurements for each satellite i :

$$\underbrace{\begin{pmatrix} \rho_k^i \\ d_k^i \end{pmatrix}}_{\mathbf{z}_k} = \underbrace{\begin{pmatrix} \|\mathbf{p}_{es,k}^{e,i} - \mathbf{p}_{ea,k}^e\| + c_{b,k} \\ (\mathbf{e}_{as,k}^{e,i})^\top (\mathbf{v}_{es,k}^{e,i} - \mathbf{v}_{ea,k}^e) + c_{d,k} \end{pmatrix}}_{\mathbf{h}_k(\mathbf{x})} + \begin{pmatrix} \nu_{\rho,k}^i \\ \nu_{d,k}^i \end{pmatrix}. \quad (3)$$

The vector $\mathbf{e}_{as,k}^{e,i}$ is the normalized direction vector from the antenna to the i -th satellite, ν_{ρ}^i and ν_{d}^i are the measurement

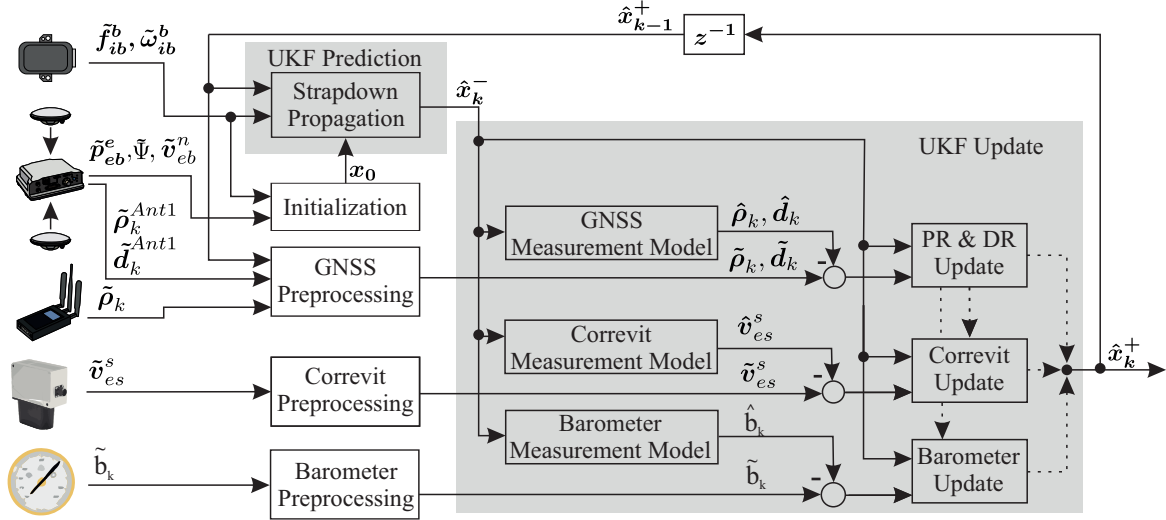


Fig. 2: Schematic of UKF 1. The pseudorange (PR) and deltarange (DR) update is performed before the optical speed sensor (Correvit) update if GNSS data is available. Update steps are skipped if no data is available. The state vector and covariance matrix are updated in the update blocks according to the UKF theory.

noise of the pseudorange and deltarange, respectively, and $p_{es}^{e,i}$ is the known position of each satellite. The position of the main antenna is derived from the estimated position of the body frame and the lever arm between the body frame and the main antenna:

$$p_{ea,k}^e = p_{eb,k}^e + C_{n,k}^e C_{b,k}^n l_{ba}^b. \quad (4)$$

The velocity of the i -th satellite $v_{es}^{e,i}$ and the velocity of the main antenna v_{ea}^e are calculated as follows:

$$v_{ea,k}^e = C_{n,k}^e (v_{eb,k}^n + C_{b,k}^n (\omega_{eb,k}^b \times l_{ba}^b)). \quad (5)$$

2) *Odometry*: The research dumper is equipped with wheel encoders and an articulation angle sensor. The measurement model is derived from a kinematic vehicle model. The comprehensive measurement model is not shown here due to reasons of space. We, therefore, refer to [9].

3) *Optical Speed Sensor*: Optical speed sensors have the benefit of slip-free measurements as the two-dimensional velocity over ground is measured. The sensor used is a Kistler Correvit S-Motion with a distance resolution of <1 mm. The measurement model is as follows [14]:

$$\underbrace{v_{es,k}^s}_{z_k} = \underbrace{C_b^s (C_{n,k}^b v_{eb,k}^n + \omega_{ib,k}^b \times l_{bs}^b)}_{h_k(x)} + \nu_{s,k}. \quad (6)$$

The matrices C rotate the velocity estimated by the navigation filter into the sensor coordinate system denoted with s . Lever arm effects due to the distance l_{bs}^b between the two coordinate systems are compensated with the measured IMU rotation rate $\omega_{ib,k}^b$. The variable $\nu_{s,k}$ describes the measurement noise.

4) *Barometer*: The measurement model for the integration of atmospheric pressure measurements b is derived from [20] and gives the expected pressure b_k for a given height $h_{eb,k}^e$

and a given temperature T :

$$\underbrace{b_k}_{z_k} = b_0 \underbrace{\left(1 + \frac{k_T (h_{eb,k}^e - h_{eb0}^e)}{T} \right)^{-\frac{g_0}{Rk_T}}}_{h_k(x)} + \nu_{b,k}. \quad (7)$$

It allows for estimated changes in the height and, therefore, correction of the state vector. Whenever the GNSS signal reception is interrupted, the current height h_{eb0}^e and atmospheric pressure b_0 is stored and the measurement model is activated. The atmospheric temperature gradient $k_T = 6.5 \times 10^{-3} \text{ K m}^{-1}$, the ideal gas constant $R = 287,1 \text{ J kg}^{-1} \text{ K}^{-1}$ and gravitation $g_T = 9.81 \text{ m s}^{-2}$ are assumed to be constant. The variable $\nu_{b,k}$ describes the measurement noise.

D. Single-epoch Fusion

The navigation solutions from the m local filters, each consisting of the state vector \hat{x}_i and the corresponding covariance matrix P_i , are fused as stated in [8]:

$$\hat{x}_f = P_f \sum_{i=1}^m P_i^{-1} \hat{x}_i \quad (8)$$

$$P_f = \left(\sum_{i=1}^m P_i^{-1} \right)^{-1}. \quad (9)$$

E. Initialization and Parametrization

The GNSS receiver output is used to get the initial position, velocity, heading, clock bias, and drift. The IMU is used to calculate the initial roll and pitch angles.

The IMU biases can be either obtained by averaging the measurements over a few seconds or by setting them to zero.

The UKF covariance matrices Q_0 , $R_{\text{GNSS},0}$ and P_0 are initialized as described in [18], the covariance matrix of the wheel odometry and the optical speed sensor as explained in

[9] and [14], respectively, and the covariance matrix of the barometer as described in [15].

III. EXPERIMENTAL EVALUATION

A. Hardware Setup

The proposed method in this work is validated with a research dumper, shown in Fig. 3. The used IMU is a 3DM-Gx5-25 from LORD MicroStrain and the used GNSS receiver is a Novatel PwrPak7 dual-antenna receiver. The auxiliary antenna was only used to get a reference heading.

All wheels were equipped with a wheel encoder and a reluctor wheel with 100 teeth. The optical speed sensor is the Kistler S-Motion DTI. The algorithms were performed on a programmable logic controller (Bachmann MH230).

B. Evaluation Protocol

The performance of the federated filter is validated using the individual performance of UKF 1, UKF 2, and a filter with all sensors integrated in a centralized architecture (like the filter in Fig. 2).

For validation, a 190-second trajectory was driven in a gravel pit. A GNSS-denied area was simulated by blocking manually all GNSS signals for 90 seconds. Fig. 4 shows the driven trajectory.



Fig. 3: Research dumper used for evaluation (graphic from [9]).

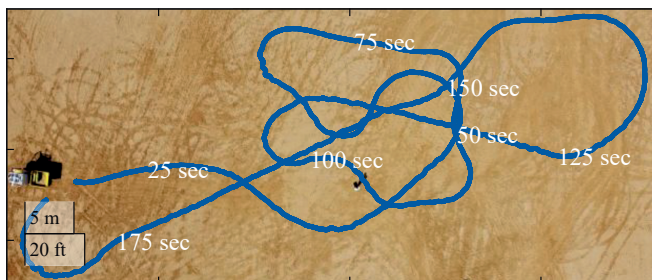


Fig. 4: Birds-eye view of the driven trajectory in the gravel pit. The data belongs to the GNSS receiver's reference solution (graphic from [9]).

The GNSS receiver in RTK-fix mode delivers the position, speed, and heading as ground truth. The 2D position error used to validate each filter is the Euclidean distance between the filter and the receiver position.

IV. RESULTS AND DISCUSSION

Figure 5 shows the 2D position error and yaw angle error of the analyzed filters. For the first 75 seconds up to the section without GNSS, all filters behave almost identically with regard to the position error. In the area without GNSS, the wheel odometry-aided filter (UKF 2) performs worst. After switching GNSS back on, all filters perform almost equally well until a pothole appears. The optical sensor is disturbed by the impact, whereas the wheel encoders aren't.

The federated and centralized filters weight the odometry and the Correvit measurements differently. From a purely subjective point of view, the federated filter forms the mean value between UKF 1 (Correvit) and UKF 2 (wheel odometry). The centralized filter is partly better and partly worse than UKF1 and UKF 2. Both, centralized filter and federated filter, perform better when the pothole occurs compared to UKF 1.

In terms of yaw angle error, the federated filter also lies in the middle between the Correvit and odometry filters. The centralized filter, on the other hand, weights the odometry update higher and is therefore very close to the solution of the odometry filter.

To summarize, it can be said that the federated architecture reduces the influence of individual sensors on the overall solution while allowing the creation of redundancies.

A. Limitations and Future Work

Tab. I lists important error values. Due to the overall small error values of all filters, it is not possible to make a clear determination on whether the federated filter is superior or inferior. A future measurement run should be more challenging for the optical sensor (e.g. potholes and puddles) and the wheel odometry (e.g. wheel slip).

TABLE I: Position and heading errors of the four evaluated filters (\bar{x} : mean value, σ : standard deviation).

Filter	2D-error (m)		Heading error (°)	
	\bar{x}	σ	\bar{x}	σ
UKF1	0,21	0,12	1,37	1,59
UKF 2	0,21	0,12	1,63	1,02
centralized	0,20	0,13	1,59	0,95
federated	0,19	0,12	1,36	1,15

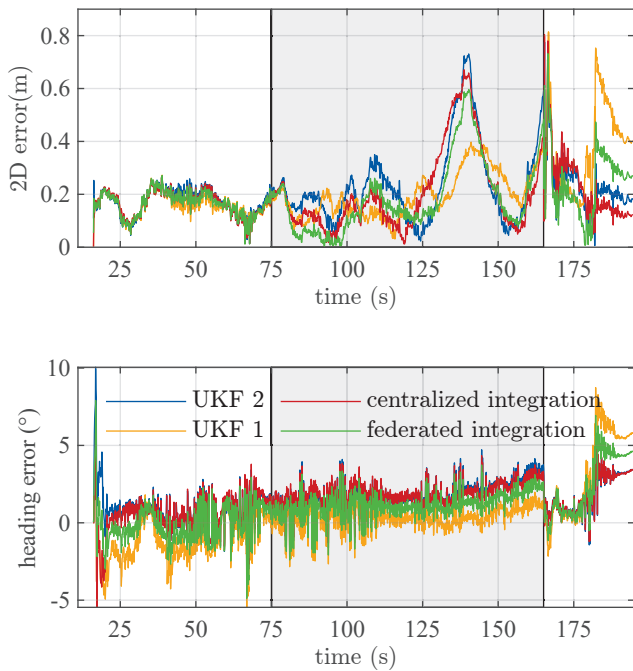


Fig. 5: Evaluation plots showing position and yaw angle error. The gray area indicates the simulated GNSS interruption.

V. CONCLUSION

Both architectures, the centralized and the federated, are suitable for navigation in terms of accuracy. The odometry-based and Correvit-based filters themselves have a centralized architecture and achieve a low position error. However, if redundant sensors are to be fused, better performance can be achieved with a cascaded architecture, as shown in the evaluation. As only one test run was used to validate the different integration architectures, further tests have to be conducted.

REFERENCES

- [1] J. Wendel, *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*, 2nd ed. Oldenbourg Wissenschaftsverlag. [Online]. Available: <https://www.degruyter.com/document/doi/10.1524/9783486705720/html>
- [2] X. Zheng and J. Zhu, "Efficient LiDAR Odometry for Autonomous Driving," vol. 6, no. 4, pp. 8458–8465.
- [3] K. Burnett, D. J. Yoon, A. P. Schoellig, and T. D. Barfoot. Radar Odometry Combining Probabilistic Estimation and Unsupervised Feature Learning. [Online]. Available: <http://arxiv.org/abs/2105.14152>
- [4] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 298–304.
- [5] B. Liu, M. Adams, and J. Ibanez-Guzman, "Multi-aided Inertial Navigation for Ground Vehicles in Outdoor Uneven Environments," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 4703–4708.
- [6] D. Benz, J.-J. Gehrt, R. Zweigel, and D. Abel, "Speed Sensor-Aided Navigation Filter for Robust Localization in GNSS-Denied Mining Environments," vol. 70, no. 1. [Online]. Available: <https://navi.ion.org/content/70/1/navi.566>
- [7] N. A. Carlson, "Federated Filter for Distributed Navigation and Tracking Applications," pp. 340–353. [Online]. Available: <http://www.ion.org/publications/abstract.cfm?jp=p&articleID=968>
- [8] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd ed. Artech House.
- [9] D. Benz, J. Weseloh, D. Abel, and H. Vallery, "CIOT: Constraint-Enhanced Inertial-Odometric Tracking for Articulated Dump Trucks in GNSS-Denied Mining Environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10 587–10 593. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10160664>
- [10] T. Konrad, J.-J. Gehrt, J. Lin, R. Zweigel, and D. Abel, "Advanced state estimation for navigation of automated vehicles," vol. 46, pp. 181–195.
- [11] J.-J. Gehrt, T. Konrad, J. Lin, M. Breuer, D. Abel, and R. Zweigel, "High Precision Localisation with Dual-Constellation for Railway Applications," in *Proceedings of the International Technical Meeting of The Institute of Navigation*, pp. 35–45. [Online]. Available: <https://www.ion.org/publications/abstract.cfm?articleID=15570>
- [12] S. Liu, J.-J. Gehrt, D. Abel, and R. Zweigel, "Dual-Constellation Aided High Integrity and High Accuracy Navigation Filter for Maritime Applications," in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*, pp. 762–774. [Online]. Available: <http://www.ion.org/publications/abstract.cfm?jp=p&articleID=16723>
- [13] M. Nitsch, J.-J. Gehrt, R. Zweigel, and D. Abel, "Tightly Coupled INS/GNSS Navigation Filter for the Automation of a River Ferry," vol. 54, no. 16, pp. 139–145. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896321014877>
- [14] D. Benz, J.-J. Gehrt, R. Zweigel, and D. Abel, "Speed Sensor-Aided Navigation Filter for Robust Localization in GNSS-Denied Mining Environments," in *Proceedings of the 2022 International Technical Meeting of The Institute of Navigation*, pp. 1457–1468. [Online]. Available: <http://www.ion.org/publications/abstract.cfm?jp=p&articleID=18194>
- [15] D. Benz, E. Clausen, and D. Abel, "Robuste Zustandsschätzung für die navigation von autonom fahrenden transporteinheiten in der rohstoffindustrie." [Online]. Available: <https://publications.rwth-aachen.de/record/973938> (Accessed 2024-01-19).
- [16] Welmec, "Guide on Common application of Directive 2009/23/EC Non-automatic weighing instruments," WELMEC Secretariat.
- [17] M. G. Petovello, "Real-time integration of a tactical-grade IMU and GPS for high-accuracy positioning and navigation." [Online]. Available: <https://prism.ualgary.ca/handle/1880/42707> (Accessed 2021-09-06).
- [18] M. Breuer, T. Konrad, and D. Abel, "High Precision Localisation in Customised GNSS Receiver for Railway Applications," in *29th International Technical Meeting of The Satellite Division of the Institute of Navigation*, pp. 779–787. [Online]. Available: <https://www.ion.org/publications/abstract.cfm?articleID=14696>
- [19] J.-J. Gehrt, T. Jeinsch, and D. Abel, "Hochgenaue und robuste gnss-gestützte zustandsschätzung für die autonome schiffahrt." [Online]. Available: <https://publications.rwth-aachen.de/record/835859> (Accessed 2022-09-24).
- [20] ICAO, "Manual of the ICAO standard atmosphere, Dokument 7488/3, International Civil Aviation Organization," INTERNATIONAL CIVIL AVIATION ORGANIZATION.

Self-Locked Asynchronous Controller for RISC-V Architecture on FPGA

Florian Deeg, Sebastian M. Sattler

Chair of Reliable Circuits and Systems

Friedrich-Alexander-University Erlangen-Nuremberg

Paul-Gordan-Str. 5, 91052 Erlangen, Germany

Email: {florian.deeg,sebastian.sattler}@fau.de

Abstract—We present a new approach for designing an asynchronous control unit for a RISC-V processor using dual-rail domino logic and a self-locking mechanism. The proposed method is based on the observation that dual-rail domino logic can be mapped to look-up tables in FPGAs. This allows for the design of a self-locking asynchronous control unit that is both inherently structurally safe and efficient. First we discuss the concept of dual-rail domino logic and its advantages for asynchronous circuits. A self-locking mechanism is presented that can be used to prevent asynchronous circuits from entering erroneous states. The mechanism is based on the use of a pulse circuit that locks the input, triggers a precharge and then an evaluate phase until it acknowledges the outputs and unlocks the input. This ensures that the circuit is in a stable state before it starts the computation. Afterwards, we apply the proposed approach to the design of an asynchronous control unit for a RISC-V processor. The control unit is implemented using look-up tables and function stable circuits. The result is a control unit that is both safe and efficient.

I. INTRODUCTION

Digital circuitry distinguishes between synchronous and asynchronous circuits. Synchronous circuits use a common clock signal to control circuit functions, while asynchronous circuits use alternative synchronization methods instead of a global clock. This alternative synchronization method results in increased resistance to noise and other disturbances compared to other circuit types. Synchronous circuits are commonly used in microprocessor technology because they are easier to implement and debug than asynchronous circuits. However, in certain cases, such as when performance or power consumption is a concern, asynchronous circuits may offer advantages. RISC-V is a novel Instruction Set Architecture (ISA), proposed and designed by Berkeley, that is rapidly gaining prominence on the scene. The key concept behind this ISA is to enable the implementation of a Reduced Instruction Set Computer (RISC) processor with a load-store architecture without the need to pay royalties for its use. [6] In addition, the ISA was designed with modularity in mind. It is therefore possible to enable more features of the processor architecture by including different ISA extensions. Modularity and royalty-free have proven to be key concepts for hardware developers, who are now more inclined to provide solutions specifically tailored to a niche problem. RISC-V is still in its infancy, and clearly behind the current market dominance of the x86 and ARM ISA for the high-end embedded market in System-on-a-chip (SoC) [1] with automotive applications, for

example. However, we are likely to see a growth of RISC-V ISA.

In this paper, we show how an asynchronous controller is designed which will control a RISC-V multicycle processor. To achieve high switching speed with low power consumption, dual-rail domino logic (DRDL) design will be used [4]. DRDL stages consist of two output paths, called rails, with each a Pin for F and \bar{F} , which are organized to complement each other. Dual-rail domino circuits are highly immune to noise and other interference, making them suitable for various applications such as microprocessor and field-programmable gate array (FPGA) technology. This paper investigates the capabilities of domino logic circuits in an FPGA and presents a technique for their implementation in the overstated platform.

In order to guarantee the feasibility and safety of the design, a low-level implementation is required. Our asynchronous design requires the specification of design rules that deviate from the standard approaches adopted clocked design. This automaton is self-clocking and fault-tolerant due to the dual-rail approach. A comparison with the clocked version highlights the advantages of the self-clocking version. The complexity of the automaton is manageable, using the necessary z-variables to define all the states specified by the ISA. In addition, the design concept is transferable to more complex applications. The pipeline has been successfully tested and is confirmed to be free of glitches, hazards and races.

II. STRUCTURE OBSERVATIONS OF DOMINO LOGIC CIRCUITS

Domino logic circuits are created using domino gates that have precharge PMOS for charging, n-complexes for function realization, and NMOS transistors for evaluation. To prevent the cascading follower stage from opening prematurely during precharge, additional inverters are placed at the output since function F generates a 0. A clock pulse controls the domino inverter, which outputs a signal that can be 0 or 1 and is then passed on to the next domino gate. In this section, we will briefly review the individual structures on transistor level (TL), starting with single-rail domino logic. We will then demonstrate how to implement dual-rail domino logic circuits in a single look-up table (LUT).

A. Single-Rail Domino Logic in FPGA

First, we will examine a single-rail domino logic (SRDL) circuit with a keeper on TL, as displayed in Figure 1. This is now

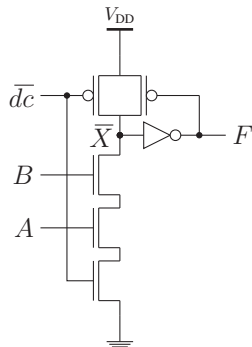


Figure 1: Single Rail Domino Logic

mapped onto a multiplexer (MUX) structure of pass transistors, which realizes a LUT, as shown in Figure 2. Because the lower

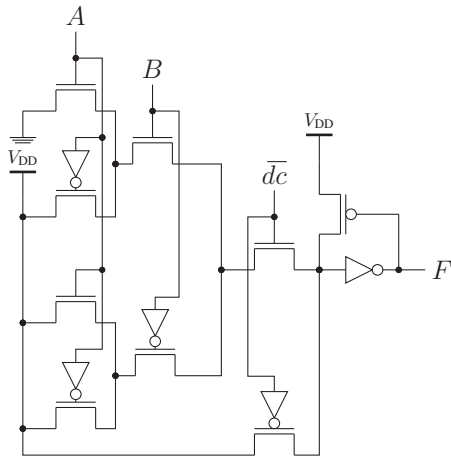


Figure 2: Single-Rail Domino Logic mapped to LUT3

path for $\bar{dc} = 0$ is to charge the inverter, all assignments are mapped to 1. The structure was not drawn to include this path for simplicity, but only the connection to V_{DD} is included. The node preceding the NMOS, which is controlled by \bar{dc} , can only be charged to V_{DD} or go high-Z and hold the charge. Therefore, this simplification accurately reflects the structure. Moreover, the node \bar{X} can solely be pulled over AB to GND, meaning that the top path is the only one capable of triggering the transition from 1 to 0. Hence, the bottom path loads from 0 to 1 (Precharge), and the upper path discharges from 1 to 0 (Evaluate). By implementing this simplification and demonstrating solely the paths for the transitions, we achieve the structure depicted in Figure 2 with the exception that the transistor for Evaluate is closer to the output, as shown in Figure 3. This structure can be replicated by exchanging the control inputs of the LUT.

B. Dual-Rail-Domino Logic

If two complementary SRDL circuits are used, they can be merged to create a dual-rail domino logic circuit. An illustration

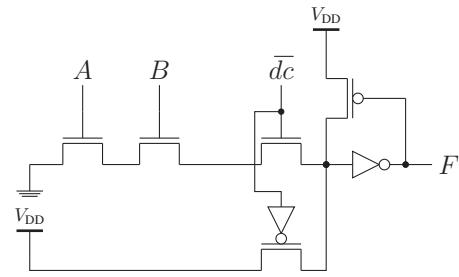
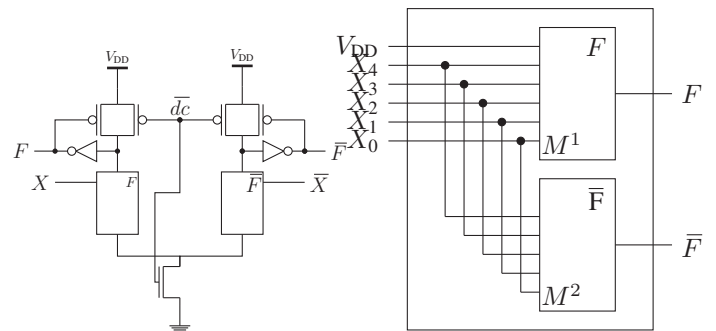


Figure 3: Transitions of SRDL mapped to LUT3

of a DRDL circuit at gate level as applied in the FPGA can be viewed in Figure 4a. Here, we have constructed two SRDL



(a) Dual Rail Domino Logic TL (b) DRDL as LUT6_2

Figure 4: Dual Rail Domino Logic

circuits with their corresponding functions F and \bar{F} . These circuits are built in parallel and governed by the same \bar{dc} . It is possible to implement the DRDL circuit in a XILINX ARTIX-7 LUT6_2 with input X_5 connected to V_{DD} , which contains two outputs $F = O_6$ and $\bar{F} = O_5$, see Figure 4b [7]. A pulse circuit is now connected to the input of the domino logic module to implement a self-locking circuit (self-X), see Figure 5. The

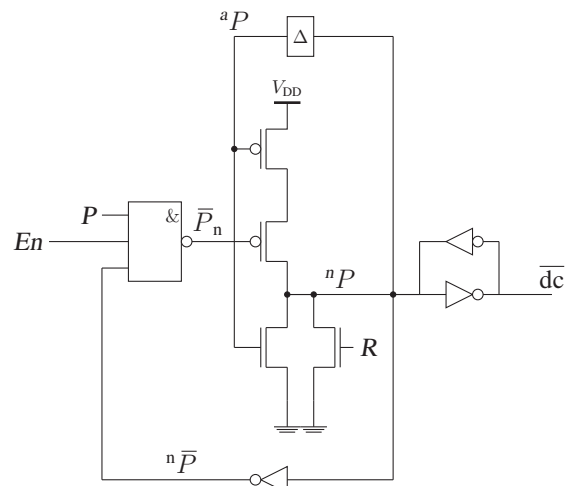


Figure 5: Pulse Circuit

self-X circuit self-locks when a pulse passes through, directly

locking the input. It is only released again by the En pin once the pulse line has passed once. If the pulse lacks sufficient energy, no switching operation is triggered. To solve this issue in FPGAs, a function stable circuit is utilized [2]. This circuit freezes at 1 in the internal state and then sets the output of the pulse circuit to HIGH (Evaluate) once a precharge delay of Δ has passed. This remains the case until the pipeline is completed, after which the input is unlocked again by the en signal. Refer to Figure 6 for the implementation in the FPGA on Gate Level (GL). The circuit self-locks once function stability

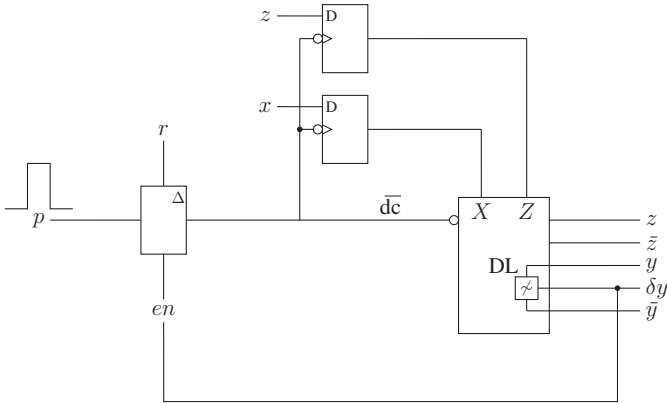


Figure 6: Self-Locked Dual-Rail Domino Logic on FPGA

is ensured, which is the case when $\tau_{inv} < \tau_{LUT}$ of the LUT [2]. This stable self-locking design principle can be now applied to pipelines in general. Function stability is ensured by limiting the input combinations to the LUT for stable feedback. To accomplish this, a Design Rule Check (DRC) was developed. The DRC ensures function stability by comparing the verified LUT assignments for feedback with that used in the current design.

III. CONTROL UNIT FOR RISC-V ARCHITECTURE

The simplified instruction set at the core of RISC-V is small and orthogonal, allowing for a thriving ecosystem of innovation. This simplified approach reduces the hardware requirements and improves overall performance by eliminating the complexity and overhead associated with complex instruction sets. The paper presents the design of a control unit for a 32-bit RISC-V architecture.

A. Control Unit Design

The implementation of an asynchronous multicycle control unit in the FPGA is demonstrated below. A multicycle unit was chosen to fully utilize the benefits of asynchronicity by dividing cycles and handling particularly long cycles separately. The multicycle controller from [3] serves as our reference point for the design. The GL representation of the automaton can be seen in Figure 7. The state transfer function takes op-code variables 6:3 as inputs. The output function takes funct3, funct7₅, and the zero flag as additional inputs. The corresponding state diagram with the states of the multicycle processor control unit are shown in Figure 8. [3]

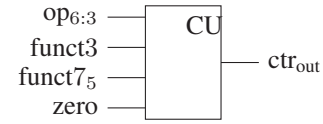


Figure 7: Block Diagram of the Control Unit

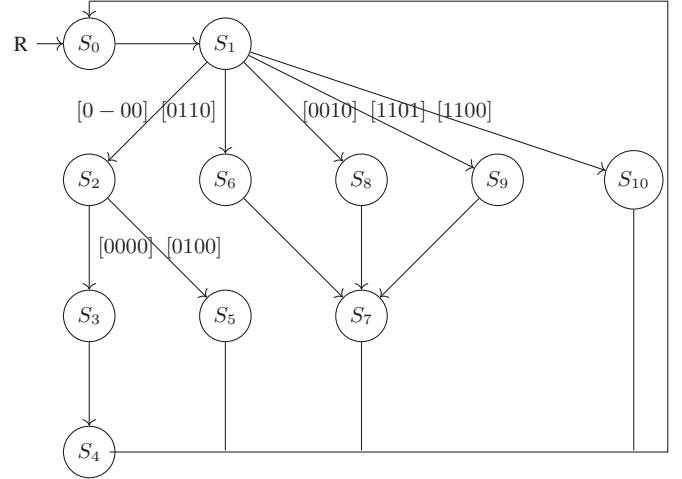


Figure 8: Automaton Graph

B. Control Unit

The control unit has 11 states, each of which generates different outputs for different instructions. To understand the different states and their function in the central processing unit (CPU), please refer to [3]. It is designed as a Moore machine, which needs more states than mealy machines in general. The machine is coded as a multi-cycle machine, which means that there are different cycles (different number of states to get back to S_0) for different instructions, e.g. the load word instruction goes through 5 states (needs 5 cycles), while the branch-equal (beq) instruction needs only 3 cycles. This results in different path delays and especially for asynchronous architectures in better performance. We have used only bits 6 to 3 of the opcode ($op_{6:3}$) as input, because they are all disjoint to each other, and therefore each event can be controlled individually by four input variables. The automaton is now designed as a dual-rail domino logic pipeline circuit.

C. Pipelined Automaton

The automaton with constant instructions through all states can be designed with a single self-X pipeline. The automaton is coded one-hot (except for state [00000]) to make the pipeline as easy to design as possible. Since the automaton has a maximum of five runs, a pipeline with five states and five z-variables is designed, see Figure 9.

Edge A and B from the opcode are $A = [1100]$ and $B = [01 - 0] \vee [0 - 10] \vee [1101]$.

The domino logic then simply passes the 1 for each state until the final state is reached. The pipeline is designed to be function

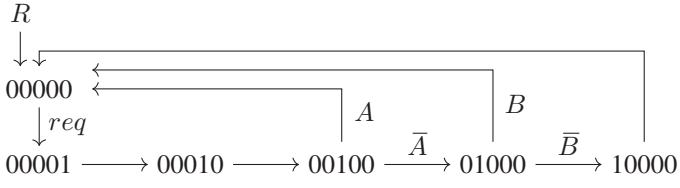


Figure 9: One-Hot encoded Pipeline

stable and self-locking, i.e. each incoming pulse locks the input and sets the next stage only when $en = 1$ and an incoming pulse P arrives. The stabilization is valid for each node and is therefore not drawn. To simplify the pipeline, the automaton was coded as a mealy machine, which means that the output function also depends on the input. This is not a problem in this case because the input is also locked by the self-locking mechanism and will be stable until the automaton has run through the pipeline once. Equation 1 to Equation 11 depict the equations for the implemented pipeline.

$$f = (f_4, f_3, f_2, f_1, f_0)$$

$$= ((F_4, \bar{F}_4), (F_3, \bar{F}_3), (F_2, \bar{F}_2), (F_1, \bar{F}_1), (F_0, \bar{F}_0)) \quad (1)$$

$$F_4 = \bar{X}_2 \bar{X}_1 \vee \bar{X}_3 X_0 \vee X_3 X_1 \vee X_3 X_2 \bar{X}_0 \quad (2)$$

$$\bar{F}_4 = \bar{X}_3 X_1 \bar{X}_0 \vee \bar{X}_3 X_2 \bar{X}_0 \vee X_3 X_2 \bar{X}_1 X_0 \quad (3)$$

$$F_3 = \bar{X}_3 \vee \bar{X}_2 \vee X_1 \vee X_0 \quad (4)$$

$$\bar{F}_3 = X_3 X_2 \bar{X}_1 \bar{X}_0 \quad (5)$$

$$F_2 = F_1 \quad (6)$$

$$\bar{F}_2 = \bar{F}_1 \quad (7)$$

$$F_1 = F_0 \quad (8)$$

$$\bar{F}_1 = \bar{F}_0 \quad (9)$$

$$F_0 = REQ \quad (10)$$

$$\bar{F}_0 = \overline{REQ} \quad (11)$$

The resulting pipeline can be seen in Figure 10. Each stage i has outputs $f_i = (F_i, \bar{F}_i)$ which feed the next stage.

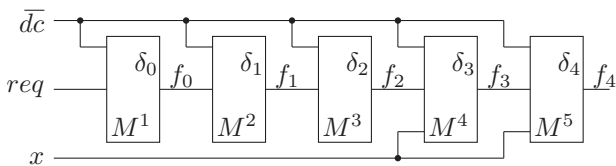


Figure 10: Pipeline in GL

IV. IMPLEMENTATION

The complete control unit consists of the pulse circuit, the DRDL pipeline and the completion detection to generate the en -signal to unlock the input. The completion detection consists of an exclusive-or (XOR) gate for each stage and evaluates after each transition to a new state whether the outputs of the dual-rail LUTs are complementary to each other. When all XOR gates are 1, means all dual-rail stages have disjoint outputs, enable is set to 1 and the input is unlocked. It exhibits the same GL representation as in Figure 6. To realize our structures, we have integrated the control unit in a given synchronized RISC-V CPU and used an ARTY-A7 board with an ARTIX-7 FPGA (XC7A35TICSG324-1L). The control unit was programmed in VHDL at low level. and the device realization in Xilinx Vivado can be seen in Figure 11.

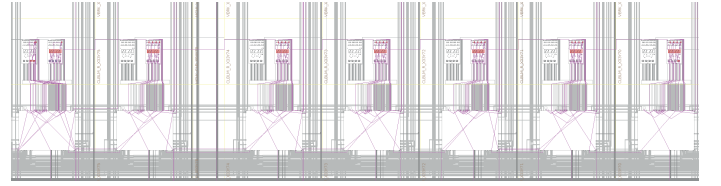


Figure 11: Control Unit after Implementation step in Vivado

A. Implementation Results

Table I depicts the utilization report of the FPGA. The

Unit	LUTs	Registers	Slice
Synchronous	15	16	6
Asynchronous	15	7	5

Table I: Used Ressources in FPGA

structure that was implemented was expected to occupy a larger area due to the dual-rail approach [5]. However, the dual-rail stages were successfully implemented within a single LUT, resulting in no change in the number of LUTs. Furthermore, the self-synchronization technique reduced the number of registers. When dealing with larger projects, it is important to consider the trade-off between area and other factors, such as clock skew issues. Self-clocked pipelines reduce the need for complex clock distribution networks and do not suffer from clock skew. Another benefit of asynchronous circuits is performance and power consumption. To understand the benefits of performance different modules of the CPU can be fed by different clocks and the overall performance can be optimized. Since our synchronous CPU is clocked with one global clock line there is no change in performance as of now. This will be a future project. The dynamic power consumption of the asynchronous automaton was approximately one-third that of the synchronous automaton.

V. CONCLUSION AND FUTURE WORK

This paper presents a new approach for designing an asynchronous control unit for a RISC-V processor using DRDL and a self-locking mechanism. The approach offers several advantages over traditional synchronous approaches, including

safety and reliability, as the self-locking mechanism prevents the circuit from hazards or races and the use of dual-rail technology makes it resistant to glitches. Another advantage is constituted by high switching speeds and low power consumption. Additionally, it is scalable and can easily be adapted to different instruction sets and clocks. The feasibility of the proposed approach was demonstrated by implementing the control unit in an FPGA. The pipelined automaton was programmed in VHDL at a low level, and its implementation in Vivado was successful. The results indicate that the pipeline operates correctly and efficiently. The proposed approach has the potential to be used in a wide range of applications.

In future work, we plan to investigate the use of more asynchronous components in the CPU. Request and acknowledge signals will be used to communicate between different components, further improving safety, reliability, and performance of the architecture. We consider dividing the pipeline into smaller steps to increase parallelism and improve performance. To expand the use of asynchronous design, a computer-aided design tool must be accessible to convert functionality into secure structures.

REFERENCES

- [1] Enfang Cui, Tianzheng Li, and Qian Wei. Risc-v instruction set architecture extensions: A survey. *IEEE Access*, 11:24696–24711, 2023.
- [2] Florian Deeg, Florian Eiermann, and Sebastian M. Sattler. Verification of function stable muller c-element in fpga. In *AmE 2023 – Automotive meets Electronics; 14. GMM Symposium*, pages 62–67, 2023.
- [3] S. Harris and D. Harris. *Digital Design and Computer Architecture, RISC-V Edition*. Elsevier Science, 2021.
- [4] David Hodges, Horace Jackson, and Resve Saleh. Analysis and design of digital integrated circuits : In deep submicron technology / d.a. hodges, h. g. jackson, r.a. saleh. 01 2004.
- [5] Hossein Rezaei and Soodeh Aghli Moghaddam. Implementation of low-power and high-performance asynchronous dual-rail join using domino logic gates in 16-nm technology. In *2016 24th Iranian Conference on Electrical Engineering (ICEE)*, pages 142–147, 2016.
- [6] Andrew Waterman. Design of the risc-v instruction set architecture. 2016.
- [7] Xilinx. *Xilinx Vivado Design Suite 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide*, 2 edition, July 2012.

Integration of a 77GHz automotive radar system into plastic surfaces using MID-technology

Thomas Mager^{#1}, Jabil Diri^{#2}, Pascal Kneuper^{†3}, Stephan Kruse^{‡4}, J. Christoph Scheytt^{‡5}

[#] Product Engineering, Fraunhofer Research Institute for Mechatronic Systems Design IEM, Paderborn, Germany

[†] Heinz Nixdorf Institute, Department of Electrical Engineering, Paderborn University, Paderborn, Germany

[‡] Institute for Photonic Quantum Systems, Department of Electrical Engineering, Paderborn University, Paderborn, Germany {1thomas.mager, 2jabil.diri}@iem.fraunhofer.de, {3pkneuper, 4stkruse, 5cscheytt}@hni.upb.de

Abstract

Novel and innovative technologies like automated and autonomous driving will provide solutions to pressing global issues related to traffic. Autonomous cars will contribute by optimizing traffic flows, resulting in a more efficient use of energy resources and infrastructure. In addition, they ensure better traffic regulation, which results in fewer casualties and more safety. To maintain a leading position in this sector, European car manufacturers must adapt to the needs of the changing market by addressing key technical challenges around autonomous driving.

Radar systems are a basic prerequisite for automated and autonomous driving. However, the amount of radar sensors per car must increase from one to about ten in the future to allow for full autonomy. This, in turn, demands very cost-effective solutions without compromising on their performance. Moreover, these sensors must be easily integrable in the car exterior, yet invisible to the eye of the customer. The alignment within the vehicle must be precise while the replacement shall be possible without significant effort. To address this need, new manufacturing technologies, materials and different integration strategies are required in the vehicle. The integration challenges in this application area of automotive radar are to be enabled by applying 3D-MID technology.

3D-MID is a packaging and integration technology that allows for three-dimensional arrangements of components and great flexibility in the shape of the final module. In the past years it has matured from a pure research topic towards high technology readiness levels (TRLs) and is already explored and adopted in many domains. Also, for the automotive sector 3D-MID is a promising solution. However, the usually rather conservative European car industry has not yet started to explore its possibilities, even though Europe is home to some of the world-leading companies and research organisations in this area. Therefore, the project MID4automotive is aiming to bridge the gap between the car manufacturers on one hand and the 3D-MID companies and experts on the other hand to help both industries to maintain and expand their leadership in their areas.

The targeted innovation in this project is the adaptation of the technology towards its use in wireless automotive modules through the development of a radar sensor in 3D-MID technology that is directly integrated into the bumper of a car. This module aims to surpass the angular resolution of state-of-the-art radar modules by a factor of six. Moreover, the integration of bare-dies into 3D-MID technology as well the integration of fibre-optic components to allow for low-loss interconnections of multiple radar sensors are targeted innovations in this project.

Radar systems that cover the entire surroundings of a vehicle are a basic prerequisite for autonomous driving. Due to their current design, the installation of the radar systems is difficult. Particularly in the side and rear areas, space to mount these radar modules is not available. Furthermore, these modules are very complex in their mechanical construction, which increases the costs. The installation also requires mounting fixtures on the vehicle, which leads to a costly assembly and adjustment. Moreover, the bumper has a negative effect on the detection quality and accuracy due to scattering and refraction of the radar signals. Finally, as a crucial performance criterion for autonomous driving, the angular domain in both elevation and azimuth needs to be improved creating so called imaging radars. This could be achieved by larger antenna arrays or coherent processing over multiple antenna front-ends distributed over the car.

Keywords — 77GHz radar, permittivity, material characterisation, 3D circuit carrier, MID, corner radar.

1 Introduction

Just a few years ago, it seemed that autonomous driving was on the verge of production readiness and that an all-encompassing autopilot was only a few years away. However, the hype has been followed by a period of disillusionment as many challenges remain unresolved or inadequately addressed. In October 2023, for example, the Cal-

ifornia Department of Transportation withdrew the licence for its fleet of autonomous vehicles from Cruise, a subsidiary of the car manufacturer General Motors, for an indefinite period due to a series of accidents [1]. This clearly shows that automated driving is an enormous challenge for the automotive industry. It is not a matter of prototypes, but of validated series vehicles that have to drive safely and under control in a real traffic scenario.

Seamless detection of the car's surroundings is the key to safer automated driving. Basically 3 different methods can be used. Camera-based detection is the most popular and flexible method. In this case, cameras detect the vehicle's surroundings in the visible and, if necessary, in the IR spectrum. In addition to the detection of traffic signs and lane-keep assistance, the system can also be used for the detection of various objects and their distance from the vehicle. The biggest disadvantage of camera-based systems is that the optics must always be kept clean from contamination. This is difficult to achieve with camera systems that face to the side or rear. Another disadvantage is that it is highly dependent on the weather conditions and the ambient light situation.

Lidar systems are another way of recording the surroundings [2]. Lidar emits pulses of light and calculates the distance from the time of flight in a similar way to a radar system. It is therefore not an imaging process but has the same limitations as camera-based systems regarding contamination and visibility.

Radar technology is considered the most suitable sensor technology for detecting the vehicle's surroundings due to its viability for extreme environmental conditions and its ability to accurately measure distance and speed. Radar is also a cost-effective solution needed to cover Advanced Driver Assistance Systems (ADAS) at autonomy levels 2+, 2 and 3+. The requirements as well as the number and position of the required sensors are listed in Table 1.

Level 2+ 5 sensors	Front: 1 short range 1 medium range Rear: 1 short range 1 medium range 1 long range
Level 3+ 7 or more	All of the sensors named above for front and rear Plus sensors on each side of the car for 360° coverage
Level 4 and beyond	Front: Short and medium range Rear: Short, medium and long range Sides: May include all sensing modalities including cameras, radar and lidar

Table 1: Autonomy levels and their corresponding sensor requirements [3]

From level 3 and above, systems are required that cover the sides as well as the long-range area to the rear of the vehicle, see figure 1. The latter radar systems in particular are difficult to integrate into a modern vehicle because of the insufficient installation space, especially for side-view radar. Integrating the radar systems into the sills and doors is very complex and expensive. So simpler and

more cost-effective integration solutions need to be developed.

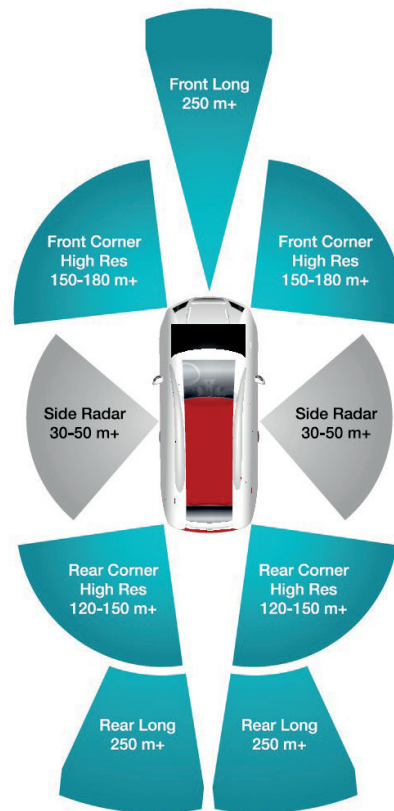


Figure 1: Radar Sensors for automated driving level 3+ [3]

2 New integration concept for a 77GHz radar system

In today's vehicles, advanced driver assistance systems (ADAS) are mainly integrated in the front of the vehicle. This is particularly useful for camera-based systems. They are easier to keep clean or can be mounted behind the windscreen. Lidar systems and radar sensors are usually installed in the front end, see figure 2. There is often enough space behind the bumper or in the radiator grille to mount the unit.

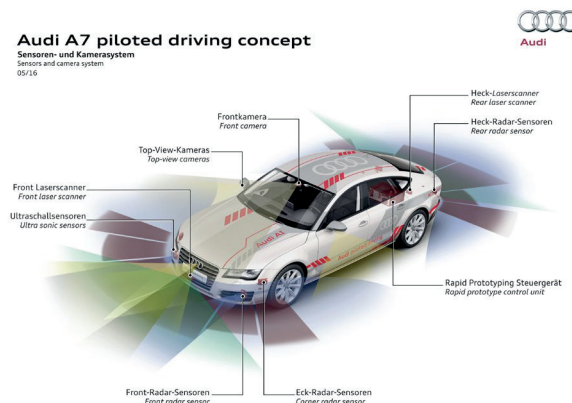


Figure 2: Sensors and cameras in a piloted Audi A7[Audi AG]

The disadvantage of this classic approach is that the position of the assistance systems is often determined by the space available and cannot be installed in the optimum position for best performance.

2.1 Direct surface integration with MID-technology

A new approach is being developed as part of the *MID4automotive* research project. In this project, the components are integrated directly onto the bumpers and plastic parts, without the need for a housing. Figure 3 illustrates this concept. The radar chip is placed directly on the inside of the bumper. This allows the RX and TX connections to be extremely short, by placing the chip very close to the antenna elements. The antenna elements are also integrated directly into the surface, with no air gap between the antenna and the bumper. This improves reflectivity and reduces scattering.

Two variations are possible:

- on the outside of the bumper
- on the inside of the bumper

The advantage of the position on the outside of the bumper is that the attenuation is minimised and the emitted electromagnetic wave is less reflected and scattered. The disadvantage is that the antenna is directly exposed to mechanical and environmental influences such as stone-chipping or contamination. For this reason, the antennas need to be additionally protected by special coatings. In addition, the RF signal must be routed from the inside of the bumper to the outside of the bumper. This requires structures (vias or couplers) that affect the signal transmission characteristics.

If the antenna is placed on the inside of the bumper, it is well protected from the mechanical impact of external influences. In addition, the signal connection is simpler as there is no need for vias or couplers. However, the radiation characteristics are degraded due to scattering and attenuation of the electromagnetic wave by the bumper substrate.

Hence, there is still a need for investigation to determine which of the two constellations has the most advantages for practical use in a car.

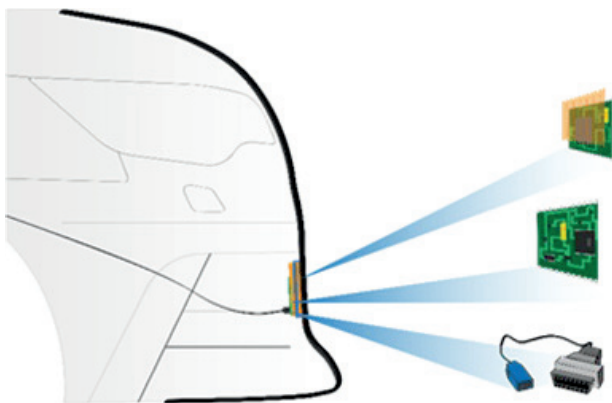


Figure 3: Planar system design for direct integration into surfaces with MID technology

2.2 MID LDS lacquer technology

This concept is implemented using so-called LDS lacquer technology. LDS refers to Laser Direct Structuring using a 1064nm laser [4]. A laser is used to activate the substrate surface previously coated with LDS lacquer. This process removes some of the paint and creates a roughened surface. The laser also heats the surface so that mixed oxides are reduced to copper seeds. Electroless plating deposits copper, nickel and gold on the activated areas [5]. The applied LDS lacquer is produced by Lackwerke Peters. Figure 4 shows a test sample that was produced with the MID LDS-lacquer. The technology can also be used to produce very fine structures of less than 100µm. This also allows BGAs and small SMD components such as 0402 to be placed.

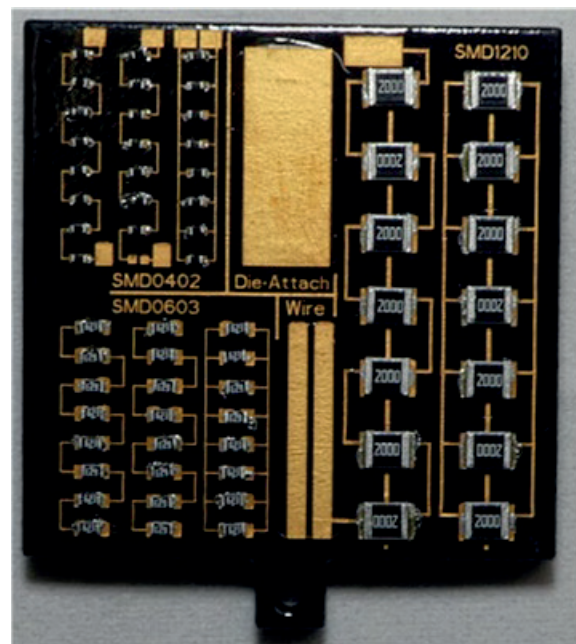


Figure 4: LDS lacquer test sample [Fraunhofer IEM]

The advantage of this process is that any existing component can be functionalised with this LDS coating. This is important for car manufacturers because otherwise they would have to do a requalification when using new materials.

The application of LDS lacquer is very similar to the application of automotive lacquer. This makes the technology particularly suitable for automotive radar integration. After cleaning of the plastic surfaces, the first step is the application of a primer. The primer increases the surface energy. This improves the adhesion of the LDS lacquer to the plastic surface.

Due to the various primers and pre-treatment methods, such as plasma activation, nowadays even difficult materials, such as HDPE (high density polyethylene), polypropylene, EPDM and polyethylene, can be reliably coated. The final coating is applied with LDS lacquer after a suitable evaporation time. It is important that all coats are applied wet on wet to achieve the best possible adhesion.

Figure 5 shows the result of the coating process.



Figure 5: coated plastic parts with LDS lacquer

Afterwards, the paint is cured at approx. 50 to 60°C. As a result, the paint reaches final strength faster and residual solvents diffuse out of the lacquer faster. Subsequently, the coated component can be structured using a 50µm IR-laser with a wavelength of 1064 nm [4]. Figure 6 illustrates the activation process inside the LPKF laser equipment. The next step is to clean the components of any laser ablation impurities which tend to stick to the surfaces in spite of the air suction.

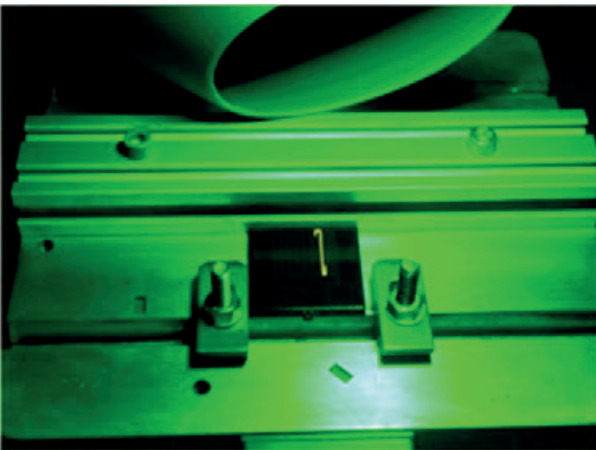


Figure 6: Laser activation

After final electroless metallisation, the component is finished and can be assembled with other electronic components. Figure 7 shows the conductors of a 3D shaped MID carrier in LDS lacquer technology.



Figure 7: Conductors on a 3D MID component

2.3 Assembly and integration on a substrate carrier

After the concept and technology have been introduced in the previous sections, the mounting and bonding technology used to integrate the radar chip into the plastic surface will be discussed in more detail.

The basic structure is illustrated in Figure 8. The chip is glued to the substrate. For better positioning, a recess or a mechanical stop is implemented. Gluing is achieved using a special UV-curing adhesive. The bond connection is made using a 25 micron aluminium bond. Figure 9 shows a demonstrator chip after this process.

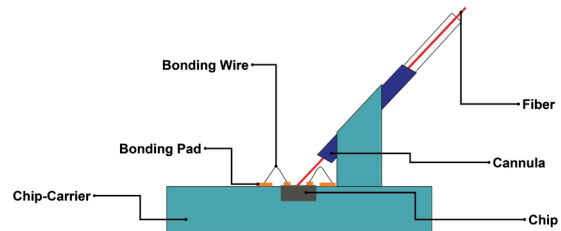


Figure 8: Mounting and bonding concept

In the next iteration of the system, a 77 GHz photonic radar chip will be integrated, which requires an optical input signal. This signal is fed to the chip via a fibre optic cable. In order to achieve a high coupling efficiency, the fibre must be precisely guided onto the chip's optical link, which is only several µm thick. A holder for the fibre is therefore included in the design of the substrate, as shown in Figure 8. A cannula is then inserted into this sleeve to stabilise the fibre and guide it to the chip's optical link. On the chip, an additional 3D-structure is manufactured onto the optical couplers via a photonic wire bonder to increase the precision of the fibre alignment. With the help of these guiding structures on the chip and the MID carrier, the fibre can be precisely aligned with the optical link under the microscope.

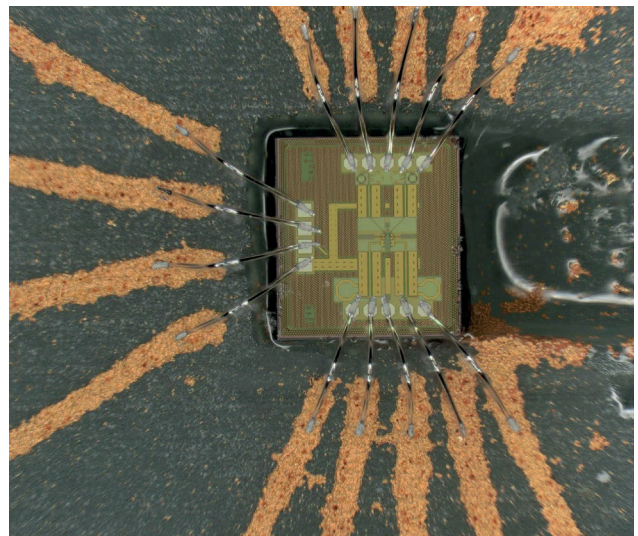


Figure 9: Mounted and bonded demo chip on a MID-substrate (without photonic interconnect)

After the fibre is aligned, the cannula is fixed in this position with a fast curing adhesive. The next step is the sophisticated process of fibre gluing on the optical link. A small, defined amount of a special adhesive is applied to the exposed part of the fibre under a microscope. This adhesive has the same optical index as the fibre. This minimises reflections on the optical link from this optical glue connection. The exact and reproducible amount of adhesive is applied with an automated dispenser. As soon as the adhesive has run down the fibre to the chip, it is immediately cured using a high-power UV source at 405nm.

2.4 Integrated 77 GHz photonic radar transceiver

The utilized integrated photonic radar transceiver in future module iterations is characterized in [6], [7]. It is fabricated in IHP 250nm SiGe BiCMOS technology and supports optical signals in both O-band (1310 nm) and C-band (1550 nm). The optical signal is generated in a remote base station at 19.25 GHz and distributed coherently to both transmitter and receiver circuits. In the transmitter, the input signal is up converted to the 77 GHz band via a frequency quadrupler. In the receiver, the received radar signal is IQ down-converted coherently. The double-sided IF bandwidth is 1 GHz. The signal is then converted back to the optical domain via an off-chip lithium niobate (LiNbO₃) Mach-Zehnder modulator (MZM). The system block diagram is shown in Figure 10. Transmitter and receiver require a combined power of 1670 mW from a 3.6 V supply. The MID chip carriers must therefore be able to dissipate heat sufficiently. For this reason, cooling structures will be investigated in further design iterations. With the current glueing process, this poses a significant challenge. A switch to a dedicated transceiver system integrated in low-power CMOS technology may alleviate the problem of heat dissipation. After successful integration of the radar chips onto the MID carrier, measurements will be performed to evaluate the impact of the MID substrate on the chips' high frequency performance.

3 Conclusion

A novel concept for integrating 77 GHz radar in cars was presented, allowing direct integration of chips, conductors and antennas onto the bumper surface. MID technology makes this possible for the first time in this application segment. This will enable more powerful radar applications in the future by eliminating the need for bulky radar modules. The technology is currently in its early stages and will need to be continuously developed as part of the project. However, it has been demonstrated that a die can be connected directly to an MID surface.

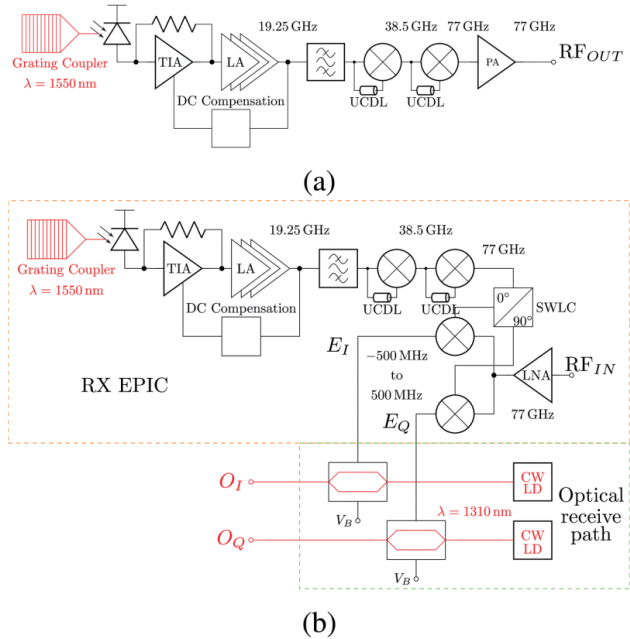


Figure 10: Caption: Block diagram of integrated 77 GHz photonic radar transmitter (a) and receiver (b) circuit with optical receive path [7].

4 Future work

The next step is to integrate and connect the photonic radar transceiver. The main focus here is on the connection of the optical fibres to the chip. New tools, methods and equipment must be developed to place the fibre correctly. Heat and RF performance of the module must be evaluated. Once the radar chip has been successfully implemented, the signal routing and antenna design will be progressed. In this context, the potential of this technology will also be explored. Figure 11 illustrates this by the example of an integrated horn antenna with dielectric lens.

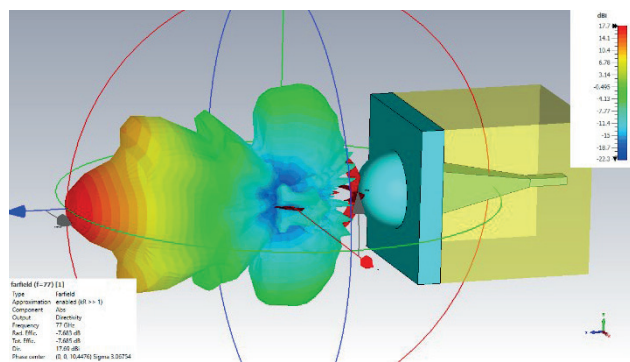


Figure 11: Concept of a horn antenna integrated in the bumper in MID-technology

5 Acknowledgment

This work was conducted as a part of the international research project MID4automotive, for a call by the EU-REKA cluster Xecs. The project is publicly funded by the German Federal Ministry of Education and Research (BMBF) and the Netherlands Enterprise Agency (RVO).

6 Literature

- [1] Lindner, R.: San Francisco stoppt Robotaxis von Cruise, FAZ, 25th Oct. 2023, online: <https://www.faz.net/aktuell/wirtschaft/auto-verkehr/general-motors-nach-unfall-mussten-auto-ome-fahrzeuge-von-der-strasse-19266701.html>, downloaded on 21th Jan. 2024
- [2] Bogdanov, V.: How Lidar Technology Helps Build Next-Gen Autonomous Vehicles, rinf.tech, online: <https://www.rinf.tech/how-lidar-technology-helps-build-next-gen-autonomous-vehicles/>, downloaded on 22th Jan. 2024
- [3] Desia, P.: How Radar is Displacing Traditional Technologies, White Paper, SPRY346, Texas Instruments, Dallas, March, 2022
- [4] LPKF Laser & Electronics AG: "LPKF-LDS: Laser-Direktstrukturierung für 3D Schaltungsträger, Info booklet, 10005424-080814-DE, Garbsen, Germany, 2014
- [5] Th. Mager, Ch. Jürgehake, R. Dumitrescu, "Functionalization of Additive Manufactured Components with Laser Direct Structurable Lacquer", 13th International Congress Molded Interconnect Devices, Conference, IEEE CatalogNo: CFP18J74-PRT, pp75-83, Würzburg, Germany, 25.-26. September, 2018
- [6] S. Kruse, S. Gudyriev, T. Schwabe, P. Kneuper, H. G. Kurz and J. C. Scheytt, "Silicon Photonic Radar Transmitter IC for mm-Wave Large Aperture MIMO Radar Using Optical Clock Distribution," in *IEEE Microwave and Wireless Components Letters*, vol. 31, no. 6, pp. 783-786, June 2021, doi: 10.1109/LMWC.2021.3062112
- [7] S. Kruse *et al.*, "Silicon Photonic Radar Receiver IC for mm-Wave Large Aperture MIMO Radar Using Optical Clock Distribution," in *IEEE Microwave and Wireless Components Letters*, vol. 32, no. 12, pp. 1447-1450, Dec. 2022, doi: 10.1109/LMWC.2022.3186432

AmEC Sponsors 2024



UNIVERSITÄT ZU LÜBECK



Supporters



AN **AVIX** GROUP COMPANY



BOSCH
Technik fürs Leben



A Cognizant Company



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG



HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES



A Textron Company



Passion to Create Value through Difference



technische universität dortmund



Offen im Denken



GMM-Fachberichte

Die Beiträge der GMM-Fachtagungen und -Workshops sind in den GMM-Fachberichten dokumentiert. Die von Programmausschüssen selektierten und zumeist reviewten Beiträge stellen aktuelle Entwicklungen aus der Mikroelektronik, Mikrosystem- und Feinwerktechnik vor.

VDE

VERLAG

Technik. Wissen.
Weiterwissen.



GMM-Fachbericht Band 106

AmE 2023

Automotive meets Electronics

14. GMM-Symposium, June 15 – 16, 2023 in Dortmund

► Today's automotive industry – still facing great challenges: Intelligent systems have become an essential part of our daily lives. The level of automation in vehicles is constantly increasing. However, the automotive industry is facing great challenges: The mobility of the future requires new concepts, such as mobility as a service, complex security & safety solutions as well as sophisticated hardware and software architectures.

2023, 113 Seiten, 140 x 124 mm, Slimlinebox, CD-ROM

49,- €

ISBN 978-3-8007-6113-5 (CD-ROM)

ISBN 978-3-8007-6114-2 (E-Book)

Titel / Untertitel	Jahr	Seiten	Buch/CD/E-Book	ISBN-Nummer	Preis
GMM-Fachbericht Band 105 Mikro-Nano-Integration Beiträge des 9. GMM-Workshops, 21. – 22.11.2022 in Aachen	2022	131	- / + / +	978-3-8007-5990-3 (CD-ROM) 978-3-8007-5991-0 (E-Book)	82,-
GMM-Fachbericht Band 104 AmE 2022 – Automotive meets Electronics 13. GMM-Symposium, September 29 – 30, 2022 in Dortmund	2022	104	- / + / +	978-3-8007-5955-2 (CD-ROM) 978-3-8007-5956-9 (E-Book)	67,-
GMM-Fachbericht Band 103 IKMT 2022 – Innovative Kleinantriebs- und Kleinmotorentechnik Innovative Small Drives and Motors, Beiträge der 13. GMM/ETG-Fachtagung, 14. – 15. September 2022, Johannes Kepler Universität Linz, Österreich	2022	310	- / + / +	978-3-8007-5938-5 (CD-ROM) 978-3-8007-5939-2 (E-Book)	161,-
GMM-Fachbericht Band 102 EASS 2022 – Energieautonome Sensorsysteme 2022 Beiträge der 11. GMM-Fachtagung, 5. – 6. Juli 2022 in Erfurt	2022	82	- / + / +	978-3-8007-5896-8 (CD-ROM) 978-3-8007-5897-5 (E-Book)	59,-
GMM-Fachbericht Band 101 ACTUATOR 2022 International Conference and Exhibition on New Actuator Systems and Applications June 29 – 30, 2022 in Mannheim	2022	369	- / + / +	978-3-8007-5894-4 (CD-ROM) 978-3-8007-5895-1 (E-Book)	192,-
GMM-Fachbericht Band 100 Elektromagnetische Verträglichkeit in der Kfz-Technik Beiträge der 8. GMM-Fachtagung, 31. Mai – 1. Juni 2022, Schwieberdingen	2022	100	- / + / +	978-3-8007-5697-1 (CD-ROM) 978-3-8007-5698-8 (E-Book)	67,-
GMM-Fachbericht Band 99 AmE 2021 – Automotive meets Electronics 12. GMM-Symposium, March, 10 – 11, 2021, Online Event	2021	80	- / + / +	978-3-8007-5487-8 (CD-ROM) 978-3-8007-5488-5 (E-Book)	46,-
GMM-Fachbericht Band 98 ACTUATOR 2021 International Conference and Exhibition on New Actuator Systems and Applications February 17-19, 2021, Online Conference	2021	453	- / + / +	978-3-8007-5454-0 (CD-ROM) 978-3-8007-5456-4 (E-Book)	191,-
GMM-Fachbericht Band 97 Mikro-Nano-Integration Beiträge des 8. GMM-Workshops, 15. – 17. September 2020, Online-Veranstaltung	2020	88	- / + / +	978-3-8007-5326-0 (CD-ROM) 978-3-8007-5327-7 (E-Book)	51,-

Sämtliche Veröffentlichungen der GMM-Fachberichte können auch im Abonnement bezogen werden. Preisänderungen und Irrtümer vorbehalten.

Bestellen Sie jetzt: (030) 34 80 01-222 oder www.vde-verlag.de/gmm-fachberichte



Tagungsbände online

Erwerb von einzelnen Tagungsbeiträgen im PDF-Format

Der VDE VERLAG publiziert jährlich rund 30 Tagungsbände mit mehr als 1.500 Fachbeiträgen. Diese können Sie auch einzeln erwerben. Zurzeit stehen Ihnen mehr als 29.000 Fachbeiträge im PDF-Format zum Download zur Verfügung. Die Kosten pro Beitrag betragen 15,- €.

Recherchieren Sie auf www.vde-verlag.de in einer täglich wachsenden Zahl von technisch-wissenschaftlichen Fachbeiträgen und erwerben Sie diese im direkten Download mit bequemen Bezahlmöglichkeiten per Kreditkarte oder PayPal.

Weitere Informationen: www.vde-verlag.de/buecher/proceedings/beitraege-suchen.html

Preisänderungen und Irrtümer vorbehalten.