ICAS 2025

The Twenty-First International Conference on Autonomic and Autonomous Systems

ISBN: 978-1-68558-241-8

March 9th –13th, 2025

Lisbon, Portugal

ICAS 2025 Editors

Oliver Michler, TU-Dresden, Germany

ICAS 2025

Foreword

The Twenty-First International Conference on Autonomic and Autonomous Systems (ICAS 2025), held between March 9 - 13, 2025, was a multi-track event covering related topics on theory and practice on systems automation, autonomous systems and autonomic computing.

The main tracks referred to the general concepts of systems automation, and methodologies and techniques for designing, implementing and deploying autonomous systems. The next tracks developed around design and deployment of context-aware networks, services and applications, and the design and management of self-behavioral networks and services. We also considered monitoring, control, and management of autonomous self-aware and context-aware systems and topics dedicated to specific autonomous entities, namely, satellite systems, nomadic code systems, mobile networks, and robots. It has been recognized that modeling (in all forms this activity is known) is the fundamental for autonomous subsystems, as both managed and management entities must communicate and understand each other. Small-scale and large-scale virtualization and model-driven architecture, as well as management challenges in such architectures are considered. Autonomic features and autonomy requires a fundamental theory behind and solid control mechanisms. These topics gave credit to specific advanced practical and theoretical aspects that allow subsystem to expose complex behavior. We aimed to expose specific advancements on theory and tool in supporting advanced autonomous systems. Domain case studies (policy, mobility, survivability, privacy, etc.) and specific technology (wireless, wireline, optical, e-commerce, banking, etc.) case studies were targeted. A special track on mobile environments was indented to cover examples and aspects from mobile systems, networks, codes, and robotics.

Pervasive services and mobile computing are emerging as the next computing paradigm in which infrastructure and services are seamlessly available anywhere, anytime, and in any format. This move to a mobile and pervasive environment raises new opportunities and demands on the underlying systems. In particular, they need to be adaptive, self-adaptive, and context-aware.

Adaptive and self-management context-aware systems are difficult to create, they must be able to understand context information and dynamically change their behavior at runtime according to the context. Context information can include the user location, his preferences, his activities, the environmental conditions and the availability of computing and communication resources. Dynamic reconfiguration of the context-aware systems can generate inconsistencies as well as integrity problems, and combinatorial explosion of possible variants of these systems with a high degree of variability can introduce great complexity.

Traditionally, user interface design is a knowledge-intensive task complying with specific domains, yet being user friendly. Besides operational requirements, design recommendations refer to standards of the application domain or corporate guidelines.

Commonly, there is a set of general user interface guidelines; the challenge is due to a need for cross-team expertise. Required knowledge differs from one application domain to another, and the core knowledge is subject to constant changes and to individual perception and skills.

Passive approaches allow designers to initiate the search for information in a knowledge-database to make accessible the design information for designers during the design process. Active approaches, e.g., constraints and critics, have been also developed and tested. These mechanisms deliver information (critics) or restrict the design space (constraints) actively, according to the rules and guidelines. Active and passive approaches are usually combined to capture a useful user interface design.

We take here the opportunity to warmly thank all the members of the ICAS 2025 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICAS 2025. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICAS 2025 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICAS 2025 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the fields of autonomic and autonomous systems.

We are convinced that the participants found the event useful and communications very open. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ICAS 2025 Chairs:

ICAS 2025 Steering Committee
Roy Sterritt, Ulster University, UK
Mark J. Balas, Texas A&M University, USA
Karsten Böhm, Fachhochschule Kufstein, Austria
Jacques Malenfant, Sorbonne Université | LIP6 Lab, France
Claudius Stern, biozoom services GmbH - Kassel | FOM University of Applied Sciences – Essen, Germany
Philippe Martine, INRIA Sophia Antipolis, France

ICAS 2025 Publicity Chairs
Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

ICAS 2025

Committee

ICAS 2025 Steering Committee

Roy Sterritt, Ulster University, UK
Mark J. Balas, Texas A&M University, USA
Karsten Böhm, Fachhochschule Kufstein, Austria
Jacques Malenfant, Sorbonne Université | LIP6 Lab, France
Claudius Stern, biozoom services GmbH - Kassel | FOM University of Applied Sciences – Essen, Germany
Philippe Martine, INRIA Sophia Antipolis, France

ICAS 2025 Publicity Chairs

Francisco Javier Díaz Blasco, Universitat Politècnica de València, Spain
Ali Ahmad, Universitat Politècnica de València, Spain
Sandra Viciano Tudela, Universitat Politecnica de Valencia, Spain
José Miguel Jiménez, Universitat Politecnica de Valencia, Spain

ICAS 2025 Technical Program Committee

Faouzi Adjed, Institut de Recherche Technologique SystemX, France
Lounis Adouane, Université de Technologie de Compiègne (UTC), France
Waseem Akram, University of Calabria, Italy
Ghassan Al-Falouji, Kiel University, Germany
Getachew Alemu, Addis Ababa University, Ethiopia
Alba Amato, Institute for High-Performance Computing and Networking (ICAR), Napoli, Italy
Vijayan K. Asari, University of Dayton, USA
Mark Balas, Texas A&MUniversity, USA
Jagadish Bapanapally, University of Wyoming, USA
John S. Baras, University of Maryland, USA
Safya Belghith, National Engineering School of Tunis | University of Tunis El Manar, Tunisia
Malek Ben Salem, Accenture Labs, USA
Sylvain Bertrand, ONERA - The French Aerospace Lab - Centre de Palaiseau, France
Navneet Bhalla, University College London, UK
Estela Bicho, University of Minho / Centre Algoritmi / CAR group, Portugal
Karsten Böhm, Fachhochschule Kufstein, Austria
Assia Boukhamla, Badji Mokhtar University-Annaba, Algeria
Kamel Bouzgou, Université des sciences et de la technologie Oran USTO-MB, Algeria / Université Paris Saclay - Univ. Evry, France
Ziv Brand, Sami Shamoon College of Engineering, Israel
Estelle Bretagne, University of Picardie Jules Verne / lab MIS (modeling, information and systems), France
Ivan Buzurovic, Harvard University, Boston, USA
Alberto Castagna, Trinity College Dublin, Ireland
Rahul Chandan, Amazon Robotics, USA
Wen-Chung Chang, Taipei Tech, Taiwan

Shuo Cheng, Georgia Tech, USA
Marco Claudio Campi, University of Brescia, Italy
Valérie Camps, Paul Sabatier University - IRIT, Toulouse, France
Elisa Capello, Politecnico di Torino and CNR-IEIIT, Italy
Constantin F. Caruntu, “Gheorghe Asachi” Technical University of Iasi, Romania
Samuel Cerqueira Pinto, Magna Advanced Development Center, USA
Meghan Saephan, NASA Ames Research Center, USA
Wen-Chung Chang, National Taipei University of Technology, Taiwan
Colin Chibaya, Sol Plaatje University, South Africa
Stéphanie Combettes, University Paul Sabatier of Toulouse | IRIT Lab, France
Shreyansh Daftry, NASA Jet Propulsion Laboratory | California Institute of Technology, Pasadena, USA
Angel P. del Pobil, Jaume I University, Spain
Sotirios Diamantas, Tarleton State University | Texas A&M System, USA
Manuel J. Domínguez Morales, University of Seville, Spain
Nikolaos Dourvas, Centre for Research & Technology Hellas (CERTH), Greece
Hind Bril El Haouzi, University of Lorraine, France
Larbi Esmahi, Athabasca University, Canada
Anna Esposito, Università della Campania “Luigi Vanvitelli”, Italy
Nicola Fabiano, Studio Legale Fabiano, Italy / International Institute of Informatics and Systemics (IIIS), USA
Ziyue Feng, Clemson University, USA
Hugo Ferreira, INESC TEC / Porto Polytechnic Institute, Portugal
Daniel Filipe Albuquerque, Polytechnic Institute of Viseu, Portugal
Terrence P. Fries, Indiana University of Pennsylvania, USA
Munehiro Fukuda, University of Washington Bothell, USA
Wai-keung Fung, Robert Gordon University, Aberdeen, UK
Lokesh Gagnani, LDRP-ITR | KSV University | SVKM, Gujarat, India
Zhiwei Gao, Northumbria University, UK
Jemin George, DEVCOM Army Research Laboratory (ARL), USA
Imam Barket Ghiloubi, University of Biskra, Algeria
Martin Giese, University Clinic of Tuebingen, Germany
Philippe Giguère, Laval University, Canada
Jordi Guitart, Universitat Politècnica de Catalunya (UPC), Spain
Maki Habib, The American University in Cairo, Egypt
Taru Hakanen, VTT Technical Research Centre of Finland, Tampere, Finland
Cédric Herpson, University Pierre and Marie Curie (UPMC) | LIP6, Paris, France
Gerold Hölzl, University of Passau, Germany
Wladyslaw Homenda, Warsaw University of Technology, Poland
Wei-Chiang Hong, Asia Eastern University of Science and Technology, Taiwan
Konstantinos Ioannidis, Information Technologies Institute - Centre for Research and Technology Hellas, Thessaloniki, Greece
Kamran Iqbal, University of Arkansas at Little Rock, USA
Minsu Jang, Electronics and Telecommunications Research Institute, South Korea
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Miquel Kegeleirs, IRIDIA | Université Libre de Bruxelles, Belgium
Lial Khaluf, Independent Researcher, Germany
Hasan Ali Khattak, National University of Sciences and Technology (NUST), Islamabad, Pakistan
Igor Kotenko, SPIIRAS and ITMO University, Russia

Hak Keung Lam, King's College London, UK
Jisun Kim, Aston University, UK
Elhaouari Kobzili, National Polytechnic School, Algeria
Charles Lesire, ONERA/DTIS | University of Toulouse, France
Baoquan Li, Tiangong University, China
Guoyuan Li, Norwegian University of Science and Technology, Norway
Jiaoyang Li, University of Southern California, USA
Ji-Hong Li, Korea Institute of Robot and Convergence, Republic of Korea
Juncheng Li, East China Normal University, Shanghai, China
Sen Li, The Hong Kong University of Science and Technology, Hong Kong
Yangmin Li, The Hong Kong Polytechnic University, Hong Kong
Zhan Li, Swansea University, UK
Enjie Liu, University of Bedfordshire, UK
Eric Lucet, CEA | LIST | Interactive Robotics Laboratory, Palaiseau, France
Ren C. Luo, National Taiwan University, Taiwan
Nacer K M'Sirdi, Aix-Marseille Université, France
Jacques Malenfant, Sorbonne Université - CNRS, France
Giuseppe Carlo Marano, Politecnico di Torino, Italy
Guilhem Marcillaud, Paul Sabatier University, IRIT, France
Aurelian Marcu, Center for Advance Laser technology CETAL - National Institute for Laser Plasma and Radiation Physics, Romania
Raúl Marín Prades, Jaume-I University, Spain
Philippe Martinet, INRIA Sophia Antipolis, France
Rajat Mehrotra, Teradata Inc., Santa Clara,, USA
René Meier, Hochschule Luzern, Switzerland
Jérôme Mendes, University of Coimbra, Portugal
Oliver Michler, Technical University Dresden, Germany
Sérgio Monteiro, Centro Algoritmi | University of Minho, Portugal
Naghmeh Moradpoor, Edinburgh Napier University, UK
Paulo Moura Oliveira, UTAD University, Vila Real, Portugal
Luca Muratore, Istituto Italiano di Tecnologia, Genova, Italy / The University of Manchester, UK
Taro Nakamura, Chuo University, Japan
Roberto Nardone, University of Reggio Calabria, Italy
Rafael Oliveira Vasconcelos, Federal University of Sergipe (UFS), Brazil
Flavio Oquendo, IRISA - University of South Brittany, France
Luigi Palmieri, Robert Bosch GmbH, Corporate Research, Germany
Eros Pasero, Politecnico di Turin, Italy
James Patten, University of Limerick, Ireland
Timothy Patten, University of Technology Sydney, Australia
Ling Pei, Shanghai Jiao Tong University, China
Damien Pellier, Université Grenoble Alpes, France
Van-Toan Pham, Vietnam Maritime University, Vietnam / National Taipei University of Technology, Taiwan
Agostino Poggi, DII - University of Parma, Italy
Radu-Emil Precup, Politehnica University of Timisoara, Romania
Md Muzakkir Quamar, King Fahd University of Petroleum and Minerals (KFUPM), KSA
Adrian Redder, Paderborn University, Germany
Leonīds Ribickis, Riga Technical University, Latvia

Douglas Rodrigues, Paulista University - UNIP, Brazil
Majid Roshanfar, Concordia University, Canada
Joerg Roth, Nuremberg Institute of Technology, Germany
Loris Roveda, Università della Svizzera italiana (USI), Switzerland
Spandan Roy, International Institute of Information Technology, Hyderabad, India
Fariba Sadri, Imperial College London, UK
Mohammad Safeea, Coimbra University, Portugal / ENSAM, Lille, France
Lotfi Saidi, University of Sousse, Tunisia
Mehrdad Saif, University of Windsor, Canada
Lakhdar Sais, CNRS | Artois University, Lens, France
Michael A. Saliba, University of Malta, Malta
Ivan Samylovskiy, Lomonosov Moscow University, Russia
Ricardo Sanz, Universidad Politecnica de Madrid, Spain
Jagannathan Sarangapani, Missouri University of Science and Technology, USA
André Schneider de Oliveira, Federal University of Technology - Parana, Brazil
Vesna Sesum-Cavic, TU Wien, Austria
Mahmoud Shafiee, University of Kent, Canterbury, UK
Abdel-Nasser Sharkawy, South Valley University, Qena, Egypt
Inderjeet Singh, University of Texas at Arlington Research Institute (UTARI), USA
Edoardo Sinibaldi, Istituto Italiano di Tecnologia (IIT), Italy
J. Tanner Slagel, NASA Langley Research Center, USA
Mohammad Divband Soorati, University of Southampton, UK
Iderli Souza, Federal University of Pampa, Brazil
Bernd Steinbach, University of Mining and Technology, Freiberg, Germany
Claudius Stern, FOM University of Applied Sciences, Essen, Germany
Roy Sterritt, Ulster University, UK
Adrian-Mihail Stoica, National University of Science and Technology Politehnica Bucharest | Faculty of Aerospace Engineering, Romania
Yun-Hsuan Su, University of Washington, Seattle, USA
Saied Taheri, Virginia Tech, USA
Omar Tahri, PRISME | INSA Centre Val-de-Loire, France
Brahim Tamadazte, FEMTO-ST Institute / CNRS, France
Maryam Tebyani, University of California, Santa Cruz, USA
Francesco Tedesco, University of Calabria, Italy
Ravi Tejwani, MIT, USA
Giorgio Terracina, Università della Calabria, Italy
Carlos M. Travieso-González, Institute for Technological Development and Innovation in Communications (IDeTIC) | University of Las Palmas de Gran Canaria (ULPGC), Spain
Xuan-Tung Truong, Le Quy Don Technical University, Vietnam
Eddie Tunstel, Motiv Space Systems Inc. / Motiv Robotics, USA
Paulo Urbano, Universidade de Lisboa, Portugal
Vivek Shankar Varadharajan, École Polytechnique de Montréal, Canada
Ramon Vilanova, School of Engineering - UAB, Spain
Nikolaos (Nikos) Vitzilaios, University of South Carolina, USA
Holger Voos, University of Luxembourg, Luxembourg
Wenbin Wan, University of New Mexico, USA
Xiangxiang Wang, University of Electronic Science and Technology of China, Chengdu, China
Yin-Tien Wang, Tamkang University, Taipei, Taiwan

Copyright Information

Table of Contents

Synthetic Data Generation for Autonomic Computing

Catherine Saunders, Roy Sterritt, Peter Nicholl, Ian McChesney
School of Computing and Mathematics
Ulster University
Belfast, Northern Ireland
e-mail:{ce.saunders, r.sterritt, p.nicholl, ir.mcchesney}@ulster.ac.uk

***Abstract*—This paper discusses an approach that integrates data generation capabilities into the Autonomic Computing MAPE-K (Monitor Analyse Plan Execute and Knowledge Loop) to mitigate problems with data scarcity in autonomous space missions. The purpose of this work is to enhance the decision-making abilities of an Autonomic Manager by providing it with the ability to use simulation and data generation. A Conditional Tabular Generative Adversarial Network (CTGAN) is used to generate new synthetic datasets. Synthetic datasets are then evaluated to assess their utility. The evaluation results show that synthetic data can closely resemble the original data. However, this paper does not address the challenges of equipping a swarm with the necessary hardware, focusing instead on the feasibility of the proposed data generation pipeline.**

Keywords-Autonomic Computing; conditional generative adversarial networks; ctgan; autonomic manager; mape-k loop.

I. Introduction

Integrating data generation capabilities into the MAPE-K loop can address data scarcity challenges in space missions. The current trend in space exploration involves the development of autonomous swarms of small spacecraft that collaborate with each other to complete a common goal. NASA's Autonomous Nano Technology Swarm (ANTS) mission proposed using a swarm of a 1,000 small craft organized into 10 different classes depending on the instrument they carry [1][2]. Managing a large swarm requires a high degree of autonomy since human operators cannot manage each craft individually [3] .

The field of Autonomic Computing [4] aims to solve the complexity associated with managing a large swarm of autonomous craft. By incorporating Autonomic Computing concepts, such as the MAPE-K loop [5][6], each individual craft can self-manage its internal state and plan its actions. This work contributes to the concept of the MAPE-K control loop by adding simulation and data generation capabilities to enhance the analysis and planning stages. An in-built component that enables each swarm member to monitor and adapt its internal state helps decrease the complexity involved when designing a large swarm. Additionally, a Mission-level Autonomic Manager craft could be designated to oversee higher-level reasoning and planning tasks for the entire swarm. The MAPE-K loop could incorporate predictive analytics within the Analyse and Plan phases to improve decision-making. Prediction algorithms generally require substantial amounts of data to provide accurate results [7].

Recent advancements in Machine Learning (ML) have introduced techniques such as Generative Adversarial Networks (GANs) to help solve the issue of data scarcity [8] [9]. A generative model can be trained to produce new synthetic data that is statistically similar to the training dataset. This is particularly useful for ML prediction algorithms, as larger datasets can lead to better accuracy when identifying trends and relationships between features. However, the quality of the data is as important as the quantity; therefore, evaluation of the synthetic data is necessary to determine its usefulness. An area that could benefit from data generation is space exploration. Collecting sufficient data from space missions is a significant challenge due to high costs and risk associated with operating in a hazardous environments [10]. Deploying large swarms of spacecraft to unforgiving environments adds further complexity to mission management. Attempting to gather enough data to cover every possible scenario could prove costly and inefficient. Generative models address data scarcity by augmenting existing datasets with synthetic data that reflects the statistical properties of the real data.

By integrating data generation into the MAPE-K loop the issue of data scarcity can be mitigated, especially at the beginning of missions. A data generation pipeline could increase the dataset size so that there is enough data for prediction analytics. Additionally, the pipeline could enhance the dataset by interpolating new scenarios not captured by the swarm. This would allow the mission to gather comprehensive data at a lower cost. This synthetic data could then be used to inform more precise planning and deployment strategies. Enhancing real data is a step up from simulation and more cost effective than a full scale mission deployment. Synthetic data reduces reliance on real world data and can help augment the data and improve planning and prediction for future missions.

This work focuses on enhancing the capabilities of a Mission-level craft that oversees the mission as a whole. By equipping this craft's Autonomic Manager with a simulation capability it could help improve planning by simulating the future mission data. This data can then be analysed and used to train prediction algorithms. In addition to prediction modelling, simulation data could be used to train a Conditional Tabular Generative Adversarial Network (CTGAN) [11] model that generates new synthetic data. The purpose of this step is to evaluate the ability of the Autonomic Manager's data generation pipeline to produce synthetic data that is a good proxy for the real data. By using simulation data to train the generative model, it functions as a first pass of the data generation pipeline.

Once generated, the synthetic data is evaluated to check whether it is similar to the original simulation data. The purpose of this is to provide assurance that any future real data gathered by the swarm and used for data generation will result in high quality synthetic data.

The data generation process could also include an interpolation feature that produces new data for unseen scenarios. This interpolated dataset could then be used to train the CTGAN to produce synthetic data for scenarios that aren't present in the original dataset. Data augmentation would help enhance real data that is scarce or incomplete. Enhancing limited datasets could reduce cost and unnecessary wastage of craft. For this work, we assume the AM has sufficient GPU resources to perform data processing. Hardware issues, such as power consumption, memory, and bandwidth in space missions are important considerations, however, these are outside the scope of this paper.

In Section II, we provide background information on previous work; Section III discusses the main contribution of Data Generation in the MAPE-K Loop; Section IV discusses the data evaluation experiments and results.

II. Background and related work

In our previous work [12], we developed a simulation tool for testing communication strategies for robot swarms, varying cooperation and cohesiveness. The ideas discussed in this paper build upon the previous work, the simulation output datasets were used to train a CTGAN model that produced synthetic data similar to the simulation output.

The Autonomic Computing concept of an Autonomic Manager (AM) that exists within each craft could be expanded so that the overseer craft processes the mission data and uses this to plan future tasks. Having an in-built component that enables each swarm craft to self-manage by monitoring and adapting their internal state helps to decrease the amount of complexity involved when designing a large swarm. The overseer AM could simulate mission data and use this simulated data to test a data generation pipeline that produces good quality synthetic data.

To prove that simulation and data generation are a useful addition to the MAPE-K loop, it is important to evaluate the synthetic data's quality [13]. The evaluation of the synthetic data is necessary so that there is confidence in the quality of the synthetic data produced. If the simulated data is evaluated to be of high quality, the AM could use the real data and generate synthetic data to increase the dataset size. This data could also be enhanced to include swarm configurations that were not gathered by the swarm. The purpose of this would be to save on the cost of sending a large swarm. A relatively small swarm could gather a sample of data. The small dataset could then be used to generate a large amount of synthetic data. If the real data is insufficient for training prediction algorithms, the AM would have the option to use the synthetic data to train machine learning prediction algorithms used by its planning component.

The purpose of this paper is to show that data generation could be used as a proxy for real data, to prove this we have performed several comparative tests that evaluate the quality of the synthetic data. This is an important step as the data must be an accurate representation of the original dataset in terms of statistical similarity and feature relationships [14] [15]. Section III discusses the data generation process using CTGAN, 20 models were trained with various parameters.

III. Data Generation in the Loop

Data generation is accomplished by training a neural network that can learn the statistical properties of the training dataset. The goal of synthetic data generation is to improve the accuracy of machine learning models by increasing the size and diversity of datasets. They can also be used to enhance the privacy of individuals by creating synthetic data that anonymizes personal information contained within the original dataset.

A Generative Adversarial Network (GAN) consists of two neural networks models that compete against each other during the training process. The Generator model creates new data and the Discriminator model acts as a binary classifier that scores the new data on its accuracy to the training set. The adversarial training process continues until the Generator can produce data that can fool the Discriminator into classifying it as real.

Traditionally, GANs perform best with image data, however, a CTGAN was designed specifically for tabular data. It can work with categorical and numerical data it is also capable of learning the relationships between the features/columns. The GAN architecture consists of two models, each with an input layer, several hidden layers and an output layer. The input layer of the Generator takes a random noise vector and transforms it into output data resembling the training dataset. The Discriminator's classification of the output data is then used to calculate the loss function, a backward pass is performed through the Discriminator's network to update it's internal weights and improve its predictions.

The Discriminator is also used to help guide the Generator, a backward pass through the Generator network uses the Generator's loss value to determines how much the Generator's internal weights need to be adjusted in order to improve the data quality during the next iteration. The Generator never sees the training data; it relies on the feedback from the Discriminator. This learning technique is known as backpropagation and continues iteratively until the training process ends.

The training duration can be adjusted by modifying the number of Epochs. The Batch Size determines how many training samples are processed during one pass through the model. For this paper, we trained 20 CTGAN models and varied the Epochs and Batch Size parameters. In future work, additional hyperparameters may be considered to improve training performance. This could involve adjusting the Generator and Discriminator learning rates.

The tabular dataset outputted by the simulation includes many features, such as: 'Simulation Time', which robot discovered and found each item, and what time the items were discovered and found. The rules of the simulation stipulated that items could only be analysed by a robot of the same type as the item. This was simulated by rules that state that only a robot that is the same colour as the item can analyse that item. However, items could be discovered by any type of robot, messages would then be sent to the rest of the swarm asking for help from other robots. The swarm simulation output included all of the information discussed above, the dataset was compiled from 270 simulation runs and consists of ~50,000 rows. This was reduced to 4,600 rows for training, a sub-set of the data was used to train the CTGAN as this improved the chance of learning the distribution of the data. The subset also significantly reduced the amount of time required to train the CTGAN. The subset of data consisted of only the simulation runs in which the signal range was set to the lowest range and the robot swarm was split unevenly with 90% one type of robot and only 10% of the other type.

The flowchart in Figure 1 outlines the stages of the control loop for data generation using a CTGAN. The processing, training and evaluation code was written in Python. An AM could implement this pipeline to prepare real or simulated data for data generation. The pipeline performs data pre-processing to convert the time features to seconds so that there is consistency when performing calculations on the data. The original dataset is then filtered to a subset of the data based on key attributes such as robot split, signal type or type of communication protocol. Unique IDs were added to help maintain data integrity.

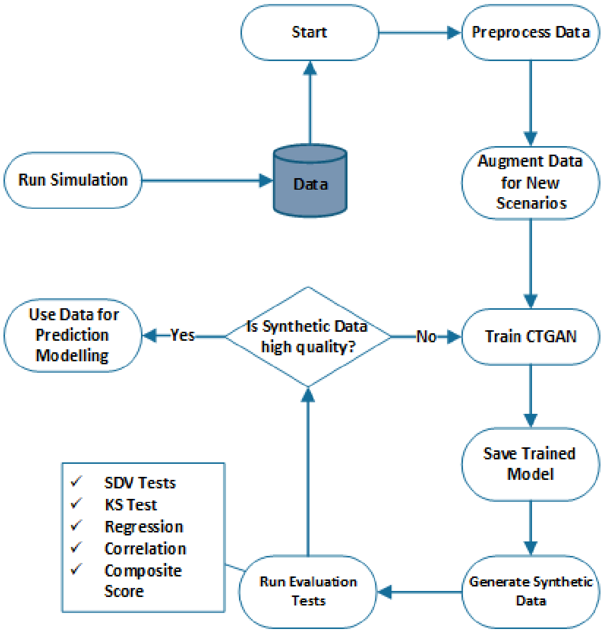


Figure 1. Flowchart of simulation and data generation pipeline.

Constraint rules are applied to ensure that the model learns the relationships between certain features and respects the rules of the original simulation. The constraint rules are essential to ensure that the synthetic data generated by the model follows the rules and relationships defined in the original simulation that produced the training dataset. In the swarm simulation, the following rules were defined:

- Time Relationship Rule: the Found (analysed) Time of the item must be greater than or equal to its Discovered Time.
- Robot and Item Matching Rule: this dictates that items can only be found/analysed by a robot of the same type (e.g., Colour).

An item can be discovered by robots that are unequipped to analyse it, but can only be found/analysed by robots that have the correct instrument. A robot that discovers but cannot analyse the item sends a help request to find a robot that has the correct instrument. If an item is discovered at time=10 seconds, it cannot have a found time less than this value. A Time Constraint check is added so that item ‘Found Time’ is always greater than or equal to ‘Discovered Time’. The Matching Constraint rule is added so that items can only ever be found and analysed by a robot of the same type (e.g., Colour). If a blue item is discovered by a red robot, the robot must send a help request to find a red robot that can analyse the item. Applying a constraint that enforces this rule ensures that the relationships between robots and items are preserved in the new data. The constraint step is necessary to maintain the fundamental rules of the simulation and ensure the synthetic data is realistic.

The ‘Train CTGAN’ step uses the Synthetic Data Vault’s (SDV) [16] implementation of CTGAN to train the model. The trained model can then be used to produce data that is similar in structure to the original data. The ‘Evaluate Data Quality’ stage uses a variety of metrics to assess the similarity of the generated data to the original test dataset. An overall composite score was then calculated from the key metrics were identified as most important. The composite score assigns a weighting to the key metrics tests.

IV. EXPERIMENTS

To assess the ability of the CTGAN to generate synthetic data similar to the original data, we conducted 20 training experiments. For each experiment, we varied the Epochs training time (100, 500, 1000, 1500, 2000), and Batch Size (50, 100, 250, 500), all other parameters stayed consistent. The Generator Learning rate was set to 0.0001 and the Discriminator Learning rate was 0.0002. The learning rate controls how much the models can learn within an iteration, with a lower value allowing the model to learn at a gradual rate. A lower batch size results in a longer training time as a smaller number of samples are viewed within each iteration. The entire dataset must be covered per epoch, therefore a lower batch size results in more iterations per epoch. Using a large batch size results in faster processing times and fewer iterations per epoch, however this can lead to less accurate weight updates and less accurate synthetic data. A larger batch size may require more epochs to reach the same results as a lower batch size and less epochs.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

The CTGAN model training script was implemented in Python, it used PyTorch [17] to enable GPU acceleration. An NVIDIA RTX A500 GPU was used for training the GANs. The training script used an instance of SDV's 'CTGAN Synthesizer', this is an implementation of Conditional Tabular GAN (CTGAN) [11] which was designed to work with tabular data. The script saves the trained CTGAN model as a pickle (.pkl) file and generates a new synthetic dataset for comparison. The model .pkl file can be reused to generate more data if necessary.

A. *Evaluation Metrics*

To evaluate the synthetic data generated by the CTGAN, we used the open-source SDV library that provides a suite of evaluation metrics. These SDV metrics assess the quality of the Synthetic Data in terms of its similarity to the original data. In addition to the SDV tests, we also implemented tests that check for similarity. These tests used Python libraries such as Scikit-learn, SciPy and Pandas to perform regression, statistical and correlation tests.

The primary objective was to generate synthetic data that closely mimics the real data, especially in terms of how the simulation time varies with different communication protocols. The goal was to have the CTGAN capture the same feature relationships and distributions.

- ***SDV Evaluation Metrics***

The SDV metrics verify that the synthetic data adheres to the schema as the original data. The schema refers to the data types, categories, and numerical ranges. The Validity Score result should always be 100%, indicating the data adheres to the schema but is not a similarity check. For all tests shown below the Validity Score was 100%. The results for each test are shown in Table 1, along with the number of Epochs and Batch Size.

TABLE I. SDV EVALUATION METRICS

Test	*Eps*	*Batch Size*	*Data Quality*	*Column Shapes*	*Pair Trends*
1	100	50	91%	91%	91%
2	100	100	91%	92%	90%
3	100	250	90%	90%	90%
4	100	500	90%	90%	89%
5	500	50	91%	91%	91%
6	500	100	92%	93%	91%
7	500	250	92%	93%	92%
8	500	500	91%	91%	91%
9	1000	50	93%	92%	93%
10	1000	100	92%	92%	93%
11	1000	250	92%	92%	91%
12	1000	500	91%	91%	90%
13	1500	50	92%	92%	92%
14	1500	100	93%	93%	92%
15	1500	250	91%	91%	91%
16	1500	500	92%	93%	92%
17	2000	50	93%	93%	93%
18	2000	100	91%	91%	91%
19	2000	250	92%	92%	92%
20	2000	500	93%	94%	92%

The Data Quality metric is a composite score calculated from the 'Column Shapes' and 'Pair Trends' values. Column Shapes measures how well the distribution of features in the data matches those in the real data. The Pair Trends metrics analyses the relationships between columns. All tests scored similar results, it was therefore necessary to perform additional tests to gain more insight into the quality of the data.

- ***Statistical Similarity Metrics***

This Kolmogorov-Smirnov (KS) test checks whether the distributions of continuous variables are similar. The variables checked were 'Discovered Time', 'Found Time', 'Simulation Time' and 'Time Difference'. The 'Time Difference' variables gives the time between an item being discovered and found. The results for 'Simulation Time' and 'Time Difference' are shown in Table II.

The KS Test consists of the KS Statistic and the KS Complement, it measures the difference between the numerical values in the two datasets. It helps assess whether the distributions are similar and if the synthetic data is a reliable replacement for the real data.

The KS Statistic measures the maximum difference between the two datasets, a smaller KS Statistic results indicates that the datasets are similar. The KS Complement test transforms the KS Statistic into a score that is more intuitive for comparison purposes, the closer the score is to 1 the higher the similarity between the datasets.

Model 20 had the best KS Complement result for 'Time Difference'. Models that were trained with higher epochs generally show higher KS Complement scores. This suggest that longer training times are better at capturing the distribution of the data. For 'Simulation Time', Model 18 performed best as it had the highest KS Complement score of 0.950. This indicates that the synthetic data closely matches the simulation times of the original data. Figure 2 shows the distribution for 'Simulation Time' for Model 18, the Synthetic data approximately matches the pattern of the real data.

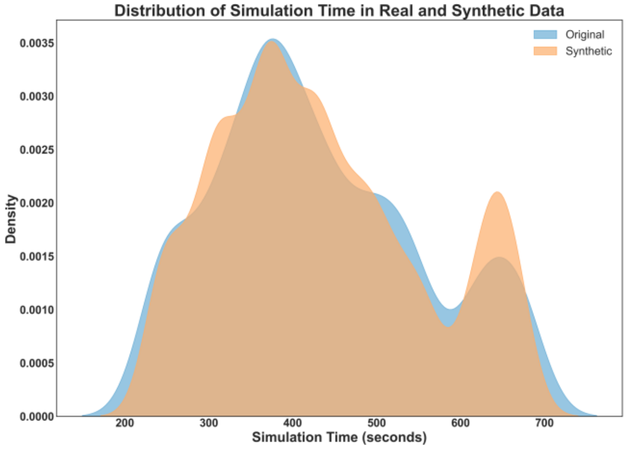


Figure 2. Simulation Time numerical distribution.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

TABLE II. STATISTICAL SIMILARITY TEST

Test	***Eps***	***Batch Size***	***Simulation Time KS Stat***	***Simulation Time KS Comp***	***Time Diff KS Stat***	***Time Diff KS Comp***
1	100	50	0.148	0.851	0.396	0.603
2	100	100	0.096	0.903	0.309	0.691
3	100	250	0.091	0.908	0.372	0.627
4	100	500	0.087	0.912	0.367	0.632
5	500	50	0.122	0.878	0.355	0.645
6	500	100	0.069	0.931	0.205	0.785
7	500	250	0.069	0.931	0.331	0.669
8	500	500	0.107	0.893	0.358	0.642
9	1000	50	0.098	0.902	0.296	0.704
10	1000	100	0.073	0.926	0.265	0.735
11	1000	250	0.118	0.882	0.259	0.741
12	1000	500	0.106	0.894	0.311	0.689
13	1500	50	0.072	0.928	0.290	0.710
14	1500	100	0.069	0.930	0.282	0.718
15	1500	250	0.079	0.921	0.330	0.670
16	1500	500	0.087	0.913	0.345	0.655
17	2000	50	0.075	0.925	0.356	0.644
18	**2000**	**100**	**0.050**	**0.950**	0.325	0.675
19	2000	250	0.061	0.939	0.438	0.562
20	**2000**	**500**	0.069	0.931	**0.174**	**0.826**

- ***Regression Results for Simulation Time***

The Regression Analysis Test compares how well a Random Forest Regressor model that's trained on the synthetic data performs against the real data when predicting 'Simulation Time'. This test demonstrates the utility of the synthetic data and how useful it is as a proxy for the real data. We trained Random Forest models on both the real and synthetic data. A combined model (Model B) was also created by augmenting the real data with synthetic data.

Table III shows the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 scores for models trained on the real data, synthetic data and the combined dataset. The MSE is the average of the error rate between actual values and those predicted by the model. The RMSE value gives the root of the MSE, it shows how much the predictions deviate from the actual values, this value is in seconds and is easier to interpret. A lower MSE and RMSE indicates that the model's predictions are closer to the actual real values. The R^2 value indicates how much the changes in the 'Simulation Time' variable can be explained by the independent variables – 'Discovered Time', 'Found Time', 'Communication Protocol'.

The range for the 'Simulation Time' variable is ~90-700 seconds. The results for the real data are, MSE 12,495 sec^2, RMSE 111.78 sec, and R^2 0.26. The model trained on the real data has a prediction error rate of 111.78 seconds, this is high and suggests that the data may be too variable for the Random Forest to learn effectively. However, the error rates for the models trained using the Synthetic data are similar to those for the real data, demonstrating that the synthetic data is a good proxy for the real data in predictive modelling.

The model trained in Test 5 performs best and has a lower RMSE than the real data. Of the combined models, Test 14 gives the lowest RMSE result. Several models trained on the synthetic data and the combined data outperform the model trained on the real data.

The limited number of independent variables may explain the poor R^2 result. The CTGAN training dataset was reduced from having all signal ranges and robot swarm splits to just one signal range and one type swarm split. This may have hindered the Random Forest from learning as it cannot use the signal and robot split as independent variables. The lack of dataset variation means both variables are constants in the dataset and do not contribute to explaining the variance in 'Simulation Time'. Despite the low R^2 values, the results are consistent across models trained on both real and synthetic data. This suggests that the CTGAN has been able to capture the relationships within the subset of data.

The results show that models trained on the synthetic data perform similar to, and in some cases better than the models trained on real data. This demonstrates the utility of synthetic data in improving predictive performance.

TABLE III. REGRESSION RESULTS FOR SIMULATION TIME

Test	***MSE (sec²)***	***RMSE (sec)***	***R²***	***Comb. MSE***	***Comb. RMSE***	***Comb. R²***
1	10,320	101.59	0.25	11,325	106.42	0.33
2	10,969	104.74	0.22	12,051	109.78	0.28
3	12,608	112.28	0.04	12139	110.18	0.28
4	18,654	136.58	-0.20	13,947	118.09	0.17
5	**9,623**	**98.10**	**0.31**	11,990	109.50	0.29
6	10,894	104.38	0.32	11,217	105.91	0.37
7	10,270	101.34	0.28	11,473	107.12	0.32
8	9,902	99.51	0.28	12,495	111.78	0.26
9	11,041	105.08	0.21	11,401	106.78	0.33
10	10,316	101.57	0.25	11,800	109.09	0.30
11	10,534	102.64	0.18	12,070	109.86	0.29
12	10,208	101.04	0.25	11,080	105.26	0.35
13	11,088	105.30	0.23	11,742	108.37	0.31
14	10,220	101.09	0.24	**10,426**	**102.11**	**0.38**
15	10,084	100.42	0.22	11,954	109.34	0.29
16	10,015	100.08	0.29	11,553	107.49	0.32
17	10,584	102.88	0.35	11,477	107.13	0.32
18	10,473	102.34	0.29	11,252	106.08	0.33
19	9,777	98.88	0.29	10,858	104.20	0.36
20	11,682	108.09	0.24	11,973	109.42	0.29

- ***Correlation of Feature Importances***

The purpose of this test was to determine the degree to which the CTGAN preserved the relationships between features. To do this, we compared the results from a Feature Importances Test to assess the correlation between feature importances values for real and synthetic data. The correlation of Feature Importances results are shown in Table IV.

Figure 3 shows the Feature Importances results for Test 10, the blue bars show the amount of importance assigned to that feature when predicting the 'Simulation Time' feature. The red bars show the importance assigned to the synthetic data features. The Feature Importances Test measures how much each feature contributes to predicting the target variables ('Simulation Time' and 'Time Difference'). Two Random Forest Regressor models were trained on both real and synthetic data, and the importance of each feature calculated.

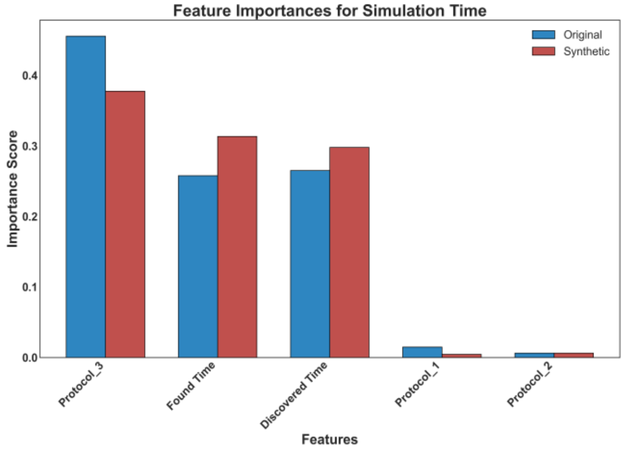


Figure 3. Feature Importances for Simulation Time.

To calculate the correlation of Feature Importances, we used the Pearson correlation coefficient. This compares the importances scores for both datasets and outputs a Correlation of Features result. A high correlation close to 1 indicates that the relationships between the features are preserved in the synthetic data. If the models disagree on which features are the most important then the correlation result will be low and closer to zero. The Correlation Similarity Score is a composite score for the Correlation of Features for both 'Simulation Time' and 'Time Difference'.

TABLE IV. CORRELATION OF FEATURE IMPORTANCES

Test	*Eps*	*Batch Size*	*Corr. of Features (Sim. Time)*	*Corr. of Features (Time Diff.)*	*Corr. Similarity Score*
1	100	50	0.950	0.996	0.960
2	100	100	0.910	0.996	0.959
3	100	250	0.683	0.990	0.919
4	100	500	0.442	0.982	0.880
5	**500**	**50**	0.981	**0.999**	**0.970**
6	500	100	0.980	0.995	0.968
7	500	250	0.980	0.999	0.923
8	500	500	0.985	0.999	0.966
9	1000	50	0.943	0.987	0.958
10	**1000**	**100**	**0.997**	0.973	0.967
11	1000	250	0.941	0.994	0.952
12	1000	500	0.974	0.999	0.966
13	**1500**	**50**	0.966	**0.999**	**0.970**
14	1500	100	0.960	0.999	0.964
15	1500	250	0.945	0.999	0.967
16	1500	500	0.984	0.993	0.957
17	2000	50	0.995	0.977	0.964
18	2000	100	0.993	0.995	0.968
19	2000	250	0.972	0.998	0.961
20	2000	500	0.964	0.982	0.962

Most tests show a high correlation above 0.9, this suggests that the synthetic data has preserved the relationships between features. Test 10 had a result of 0.997 for 'Simulation Time', indicating that the real and synthetic scores for feature importances are nearly identical.

B. Composite Results

The key metrics chosen to create the composite score were SDV Pair Trends, Overall Quality, Simulation Time KS Complement, Correlation of Feature Importances (Simulation Time), Correlation Similarity Score. These metrics were chosen to assess the statistical similarity between the synthetic and real data.

To derive an overall quality assessment, we used a weighted composite score to rank each model, the top five performing models are listed in Table V. A weighting was applied to each metric as follows: Data Quality 20%, Pair Trends 30%, Simulation Time KS Comp 20%, and Feature Importances Correlation 30%. By combining different evaluation metrics, the composite score provides a balanced view of each model's quality.

TABLE V. COMPOSITE METRIC SCORE

Test	*Epochs*	*Batch Size*	*Composite Score*
17	**2000**	**50**	**0.9485**
10	1000	100	0.9473
18	2000	100	0.9429
7	500	250	0.9402
19	**2000**	**250**	**0.9394**

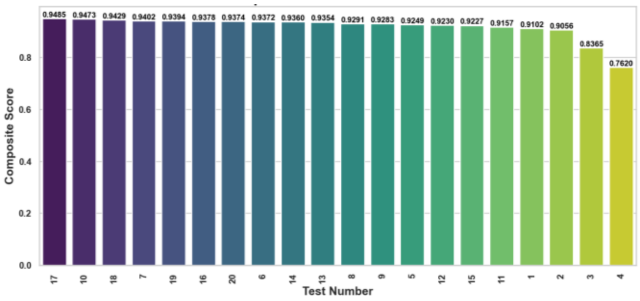


Figure 4. Composite score for 20 CTGAN models.

Model 17 (2000 Epochs, Batch Size 50) produced the highest composite score of 0.9485, this indicates a good similarity between the two datasets. The results suggest that using longer training times with low to moderate Batch Sizes are best for learning the distribution of the data. The poorest performers were models trained with only 100 Epochs, the composite score decreased as the Batch Size increased from 50 to 500. A larger batch size can speed up the training process but it results in fewer updates to the Generator's weights per epoch. Smaller batch sizes allow for more updates per epoch but they also increase training times. The composite scores are visualized in Figure 4, the bar chart shows the results of all 20 models for each of the key metrics. The visualization ranks the models from best to worst, with the best performing model ranked first.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

V. Conclusion

In this paper, we discussed integrating a data generation pipeline into the MAPE-K loop, with the goal of alleviating data scarcity in autonomous space missions. Training a CTGAN often involves trial and error, making the integration of an evaluation component vital. Training parameters can greatly influence the quality of the synthetic data produced so it is important to evaluate the quality of the synthetic data produced. The results demonstrated that synthetic data generated by a CTGAN can closely mimic the real data in terms of feature relationships and distributions. Training the CTGAN for a high number of Epochs combined with a low Batch Size (2000 Epochs, Batch Size 50) produced the highest quality synthetic data. Future work will focus on the data interpolation component to generate new configurations of the swarm not present in the original dataset.

While synthetic data can offer a solution to the problem of data scarcity, it is vital that the practical utility of the data is evaluated. This presents a new challenge that requires a suite of metrics to give a comprehensive evaluation of quality. In addition to post-training evaluation metrics, it is important to pre-process the training dataset to remove errors and biases that could be propagated into the synthetic data.

This paper demonstrates the utility of using synthetic data to increase the size of an existing dataset. However, it does not address the practical constraints associated with equipping craft with hardware capable of performing the data generation. Generating synthetic data requires significant computational resources which may not be practical when operating in a constrained environment. Future work will look at solutions such as distributing computational load across the swarm or extending processing times to simulate hardware limitations, and increase system resilience by distributing the reliance of the Swarm away from one AM in case of damage or malfunction.

References

[1] NASA. *Autonomous Nano Technology Swarm ANTS*, 2010 [Online]. Available from:
https://attic.gsfc.nasa.gov/ants/ArchandAI.html 2025.01.23

[2] S. A. Curtis, W. Truszkowski, M. L. Rilee, and P. E. Clark, “ANTS for Human Exploration and Development of Space,” IEEE Aerosp. Conf. Proc., vol. 1, pp.255–261, June 2003, doi:10.1109/AERO.2003.1235057.

[3] P. A. Oche, G. A. Ewa, and N. Ibekwe, “Applications and Challenges of Artificial Intelligence in Space Missions,” IEEE Access, vol. 12, pp. 44481–44509, Mar. 2024, doi: 10.1109/ACCESS.2021.3132500.

[4] IBM, *An Architectural Blueprint for Autonomic Computing*, IBM, 2003. [Online]. Available from: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=0e99837d9b1e70bb35d516e32ecfc345cd30e795 2025.01.23

[5] R. Sterritt, “Autonomic Computing,” Innov. Syst. Softw. Eng., vol. 1, pp. 79–88, Apr. 2005, doi: 10.1007/s11334-005-0001-5.

[6] J. O. Kephart and D. M. Chess, “The Vision of Autonomic Computing,” vol. 36, no. 1, pp. 41-50, Jan. 2003, doi: 10.1109/MC.2003.1160055.

[7] D. Werner, *Big data, advanced algorithms and new approaches for space missions.* [Online]. Available from: https://spacenews.com/big-data-advanced-algorithms-and-new-approaches-for-space-missions/ 2025.01.23

[8] I. Goodfellow et al., “Generative adversarial networks,” Commun. ACM, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[9] L. Alzubaidi et al., “A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications,” J. Big Data, vol. 10, no. 1, Apr. 2023, doi: 10.1186/s40537-023-00727-2.

[10] M. S. Hedayati, A. Barzegar, and A. Rahimi, “Mitigating Data Scarcity for Satellite Reaction Wheel Fault Diagnosis with Wasserstein Generative Adversarial Networks,” 2024 IEEE Int. Conf. Progn. Heal. Manag. ICPHM 2024, pp. 367–376, June 2024, doi: 10.1109/ICPHM61352.2024.10627589.

[11] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional GAN,” Adv. Neural Inf. Process. Syst. NeurIPS, Dec. 2019.

[12] C. Saunders, R. Sterritt, and G. Wilkie, “Collective communication strategies for space exploration,” JBIS - J. Br. Interplanet. Soc., vol. 72, no. 12, pp. 416–430, 2019.

[13] A. Figueira and B. Vaz, “Survey on Synthetic Data Generation, Evaluation Methods and GANs,” Mathematics, vol. 10, no. 15, pp. 1–41, 2022, doi:10.3390/math10152733.

[14] J. Jordon et al., *Synthetic Data -- what, why and how?*, [Online]. Available from: https://arxiv.org/pdf/2205.03257 2025.01.23

[15] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, “Generation and evaluation of synthetic patient data,” BMC Med. Res. Methodol., vol. 20, no. 1, pp. 1–40, May 2020, doi:10.1186/s12874-020-00977-1.

[16] DataCebo. *The Synthetic Data Vault*, March 2023. [Online]. Available from: https://sdv.dev/SDV/index.html 2025.01.23

[17] A. Paszke, “Pytorch: An imperative style, high-performance deep learning library,” Adv. Neural Inf. Process. Syst., vol. 32, Dec. 2019. doi:10.48550/arXiv.1912.01703

Hard Disk Drive Reliability: A Comparative Study of Supervised Machine Learning Algorithms for Predicting Drive Failure

Alistair McLean, Roy Sterritt
School of Computing and Mathematics
Ulster University
Belfast, Northern Ireland
mclean-a13@ulster.ac.uk | r.sterritt@ulster.ac.uk

Abstract— **Unexpected downtime and IT system outages can cost organisations millions of dollars in lost revenue, loss of opportunity, and negatively impacted reputation. Third party cloud services and infrastructure are commonly used by individuals and organisations as it offers the ability to create highly scalable applications without the huge cost of purchasing and maintaining their own hardware facility. Consequently, cloud service providers are challenged with ensuring that their data centres are reliable, as they have shared responsibility for the applications deployed in them. One of the most common causes of IT system failure in data centres is failing Hard Disk Drives (HDDs). It is proposed that if data centres were able to accurately predict imminent HDD failures, then appropriate action could be taken to prevent potential outages. This paper investigates the relationship between Self-Monitoring, Analysis, and Reporting Technology (SMART) attributes and HDD failure, implementing supervised machine learning methods to predict drive failure at various prediction horizons. Random Forest and XGBoost classifiers are observed to achieve the best prediction performance, with the Area Under the Receiver Operating Characteristic Curve (AUROC) calculated at 0.9185±0.0066 and 0.9162±0.0066 respectively at the shortest prediction horizon (0-24 hours prior to failure). Reallocated sectors count (SMART 5), reported uncorrectable errors (SMART 187), current pending sector count (SMART 197), and uncorrectable sector count (SMART 198) were found to be the most important SMART attributes for HDD failure prediction.**

Keywords-hard disk drive; hdd reliability; machine learning; failure prediction.

LIST OF ABBREVIATIONS

Abbreviation	Definition
AUROC	Area Under the Receiver Operating Characteristic Curve
DT	Decision Tree
FAR	False Alarm Rate
FDR	False Discovery Rate
FPR	False Positive Rate
HDD	Hard Disk Drive
k-NN	K-Nearest Neighbour
LR	Logistic Regression
ML	Machine Learning
MLP	Multi-Layer Perceptron
RF	Random Forest
RUL	Remaining Useful Life
SLA	Service Level Agreement
SMART	Self-Monitoring, Analysis, and Reporting Technology
TPR	True Positive Rate
XGB	XGBoost

I. INTRODUCTION

Unexpected downtime or outages of IT systems can have major consequences for businesses and users. It is reported that a single outage can cost an organisation millions of dollars through the loss of revenue, loss of opportunities, and diminished reputation, and that this cost impact is increasing year on year [1]. Many organisations today use cloud computing as part of their products and services, reducing the need for purchasing and maintaining their own IT infrastructure while improving the scalability of their applications. As cloud adoption continues to grow, cloud service providers have shared responsibility for their users' applications and are tasked with providing highly available services and reliable IT infrastructure. Application downtime or system unavailability can be detrimental for both the cloud provider and the cloud user, with cloud providers liable to fines for breaches of Service Level Agreements (SLAs).

Data centre outages occur from time to time, which may result in application unavailability or system downtime. The most common causes of data centre outages are on-site power-related problems, such as generator or grid failures, followed by network problems and IT system failure [2]. With respect to IT system failures, which include hardware and software failures, Hard Disk Drives (HDDs) are believed to be one of the main offenders of causing problems. HDDs are one of the most replaced hardware components and one of the least reliable, with [3] reporting that 78% of faults or replacements are attributable to hard disks. Another investigation [4] of the data centres of a major internet service organisation observed that 82% of hardware failure tickets were attributable to HDDs. Therefore, data centres could potentially improve their reliability by monitoring the health of HDDs in their IT estate and take appropriate action before a drive failure occurs.

Self-Monitoring, Analysis, and Reporting Technology (SMART) was developed in 1995 and is commonly used by manufacturers today, providing measurements collected by sensors within HDDs to report on various indicators of reliability and drive health. SMART attributes are numbered from 1 to 255 giving raw and normalised values of each measurement. For example, SMART 9 reports the power-on hours, the total count of hours that the drive has been in a power-on state across its operational lifetime. SMART 190 and 194 give measurements of internal temperature within the HDD unit. SMART 240 records the total time in hours

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

that the read-write head has been operating to position itself across the surface of the disk, while SMART 241 and 242 record the lifetime data (in bytes) written to, and read from, the drive respectively. Other attributes count the number of errors, such as SMART 1, which records the rate of read errors between the read-write head and the disk surface.

The aim of this paper is to investigate the relationship between SMART metrics and HDD failure, and to assess if Machine Learning (ML) methods can accurately predict imminent drive failure using reported SMART attribute measurements. Additionally, a comparative study will determine the best ML algorithm for application in this problem domain.

The remainder of the paper is organised as follows: Section II highlights existing work in the field, describing research approaches and their effectiveness for HDD failure prediction. Section III describes the dataset analysis and ML methods used in this paper. Section IV reports the results and performance of the ML implementation. Section V discusses the limitations of the work, highlighting improvements and opportunities for extension. Section VI summarises the research and concludes the paper.

II. EXISTING WORK

“Autonomic systems are examples of accelerated AI automation. They are self-managing physical or software systems, performing domain-bounded tasks that exhibit three fundamental characteristics: autonomy, learning and agency. When traditional AI techniques aren’t able to achieve business adaptability, flexibility and agility, autonomic systems can be successful in helping with implementation. Autonomic systems will take five to ten years until mainstream adoption but will be transformational to organizations” [5]. This work on supervised Machine Learning for HDD failure prediction fits generally with that Autonomic vision [6][7].

The ability to predict HDD failures would allow data centres to mitigate against potential outages by proactively replacing drives before they reach a state of failure. Unsurprisingly, there have been many works of research investigating and attempting to address this problem. Machine learning and probabilistic techniques are popular among researchers, applying traditional ML classification and regression methods, Bayesian networks, deep learning, or combining multiple methods with ensemble learning.

The approach taken in [8] uses SMART attributes to create HDD failure prediction models using classification and regression trees. Their experiments show the classification tree model was able to successfully predict 95% of failures with a False Alarm Rate of less than 0.1% when applied to a real-world data centre containing 25,792 HDDs. Additionally, they propose a regression tree model to evaluate the health status of the drives, where the probability of a fault occurring is predicted. Also using a tree-based model, [9] takes a binary classification approach to predict the health of HDDs in Meta’s Tectonic storage fleet. The SMART metrics of 53,000 failed HDDs were used, alongside a random sample of non-failing drives, to categorise the HDDs as healthy or unhealthy at 1 and 30 days prior to failure. Their XGBoost classifier showed limited prediction performance, achieving low precision when applied to unseen data from a different time window. However, they report noticeable improvements when using the difference, or delta, between SMART measurements over time as opposed to using the singular measurements from a set prediction horizon.

Highlighting the limitation of using SMART attributes with their default thresholds to detect failing HDDs, [10] proposes a failure prediction method using a Bayesian network to provide Remaining Useful Life (RUL) estimates of drives. Using a subset of SMART attributes and their temporal trends, the proposed Bayesian Network for Failure prediction in HDDs (BNFH) was applied to a dataset containing 49,056 drives from Backblaze’s data centres. Their evaluation showed the model outperformed standard reliability-based methods and other Bayesian network-based methods presented in [11]; and achieved similar relative accuracy to a Recurrent Neural Network presented in [12]. The work in [13] utilises ensemble learning to create a Combined Bayesian Network (CBN), where the learning results from four individual classifiers are combined to predict the remaining time before a drive fails. The individual classifiers used backpropagation neural networks, evolutionary neural networks, support vector machines, and classification tree methods. Experimental results indicate the CBN performs similarly to the classification tree model and outperforms the other models. However, the CBN has additional benefit over the classification tree model by indicating when the drive will fail, not just that it will fail.

Other research papers propose deep learning methods for HDD failure prediction. The work in [14] uses bidirectional LSTM models with multi-day lookback periods to learn the temporal progression of key health indicators present in SMART data. The proposed model achieved 96.4% accuracy in predicting HDD failure for a 15-day lookback period, outperforming a standard LSTM implementation. However, due to the inconsistency in SMART measurements recorded by different HDD manufacturers and models, the data used in this work only related to a single Seagate model (ST4000DM000) over the course of 9 months. Another deep learning approach, presented in [15], proposes a model based on Gated Recurrent Unit (GRU) neural networks and TimeGAN adversarial networks to analyse the temporal sequences of SMART attributes in HDDs, while addressing data imbalance issues. Their proposed approach achieved an average failure detection rate of 95% and a false alarm rate of 0.2%. This work also only applies to a single Seagate drive (ST6000DX000).

While existing work has achieved success in HDD failure prediction, it is not always clear which method performs best in this problem domain. The listed works in this section apply numerous ML algorithms to HDD SMART metrics using data centre drives. However, the data centres, drive manufacturers, drive models, and timeframes within the datasets will vary from paper to paper. Therefore, it is not necessarily viable to make direct comparisons. As such, the purpose of this paper is to apply multiple machine learning methods to the same dataset of operational data centre hard drives. The comparative study will derive insight into the best performing methods for HDD failure prediction.

III. METHODOLOGY

The dataset used in this paper was obtained from Backblaze [16], which records the daily SMART metrics of HDDs within their data centres. Each row in the dataset contains the date, serial number, model, capacity (in bytes), and raw and normalised values for various SMART metrics reported by each HDD. Additionally, a failure column records a binary value indicating if the drive is functional (0), or if the entry represents the last operational day before the drive failed (1). This paper uses the reported data across a 10-year period, from 1st January 2014 to 31st December 2023.

A. Data Exploration

Initial analysis of the dataset was conducted to gain a better understanding of the size and complexity of data, the HDD models that are reported on, and the prevalence of drive failures. Over the 10-year period, the dataset contained over 450 million rows, reporting the daily SMART metrics of 388,485 unique hard disk drives (identified by their serial number). In that time, 21,356 rows indicated that an HDD had failed. It appears that the data centre organisation preferred to replace failed drives, rather than repair them, as only a very small proportion of drives (0.005%) were seen to fail more than once. Table I shows that most of the drives (94.5%) did not experience failure, and nearly all failed drives only failed once.

Using the model number provided in the dataset, it was possible to analyse the failure rate with respect to drive manufacturer and drive model. Seagate models accounted for the largest proportion of drives in the data centre throughout the years. Hitachi models also made up a large proportion in the earlier years, but their presence almost entirely disappeared by 2018. Other drive manufacturers present in the data include Toshiba, HGST, and WDC. Of the 193,378 Seagate drives, 16,177 resulted in failure accounting for 75.75% of all failures in the dataset and indicates an 8.37% failure rate for all Seagate HDDs. Table II shows the top 10 models with the highest number of failures, indicating the model failure rate and their proportional contribution to the overall failures present in the data. Models prefixed with ‘ST’ are Seagate drives and Table II shows that 7 of the top 10 most failing drives belong to this manufacturer.

One of the known issues with SMART attributes is that manufacturers do not always use them equally, as mentioned in [14]. The same SMART attribute may be used to report different measurements by different manufacturers. This would make it difficult to train a machine learning model and therefore, for the purposes of this paper, HDDs belonging to a single manufacturer will be used for failure prediction. As Seagate drives are the most prevalent model of HDD in this dataset, and account for the most failures, the Seagate models from Table II were selected. These are the top failing drives and include the following models: ST4000DM000, ST12000NM0007, ST8000NM0055, ST3000DM001, ST12000NM0008, ST8000DM002, and ST14000NM001G.

TABLE I. DRIVE FAILURE COUNTS ACROSS ALL HDDS IN THE DATASET SHOWING PROPORTION OF DRIVES WITH MULTIPLE FAILURES

Number of Failures	Number of Drives	% of HDDs	% of Failures
0	367,147	94.51	-
1	21,320	5.49	99.92
2	18	0.005	0.08

TABLE II. TOP 10 HDD MODELS WITH THE HIGHEST NUMBER OF FAILURES

Model	Total HDDs	Total Failures	% of All Failures	Model Failure %
ST4000DM000	36,983	5,602	26.23	15.15
ST12000NM0007	38,838	2,106	9.86	5.42
ST8000NM0055	15,680	1,718	8.04	10.96
ST3000DM001	4,354	1,454	6.81	33.39
ST12000NM0008	20,836	1,349	6.32	6.47
MG07ACA14TA	39,292	1,173	5.49	2.99
ST8000DM002	10,305	1,037	4.86	10.06
HUH721212ALN604	11,166	600	2.81	5.37
HMS5C4040BLE640	16,349	426	1.99	2.61
ST14000NM001G	11,154	418	1.96	3.75

B. Features

Analysis of the data quality measured the prevalence of null or missing values to determine which SMART columns in the dataset could be used as features for machine learning. The HDD model, capacity (in bytes), and the raw and normalised SMART measurements were used for training and evaluating the ML classifiers.

TABLE III. SMART ATTRIBUTE FEATURES AND THEIR SPEARMAN RANK CORRELATION WITH HDD FAILURE

ID	Attribute Name	Null %	Correlation with Failure
1	Read Error Rate	0.39	-0.001
3	Spin Up Time	1.32	-
4	Start/Stop Count	1.32	0.1015
5	Reallocated Sectors Count	0.38	**0.5352**
7	Seek Error Rate	1.32	0.0584
9	Power-On Hours	0.38	0.0314
10	Spin Retry Count	1.32	-
12	Power Cycle Count	1.32	0.0959
187	Reported Uncorrectable Errors	1.32	**0.6114**
188	Command Timeout	1.32	0.1378
190	Temperature Difference	1.32	0.0429
192	Power-Off Retract Count	1.32	0.0455
193	Load Cycle Count	1.32	0.0448
194	Temperature	0.38	0.0429
197	Current Pending Sector Count	0.38	**0.5056**
198	Uncorrectable Sector Count	1.32	**0.5056**
199	UltraDMA CRC Error Count	1.32	0.0705
240	Head Flying Hours	1.32	-0.002
241	Total LBAs Written	1.33	0.0368
242	Total LBAs Read	1.33	0.0482

Furthermore, the Spearman rank correlation was measured to indicate which SMART metrics were more associated with drive failure. Table III shows the list of SMART metrics used and their Spearman rank correlation coefficients.

C. *Machine Learning Implementation*

The failure prediction in this paper was treated using a classification approach to determine if an HDD will fail or not at specific prediction horizons, or lookahead days. The SMART metrics were collected at 0, 1, 2, and 7 days prior to failure occurrences of the selected Seagate models described previously. As most drives in the dataset do not fail (94.5%), the datasets are highly imbalanced with vastly more examples of non-failing drives than failed ones. As imbalanced data can introduce bias to machine learning models, the non-failing class was under-sampled. A random sample of non-failing drives was collected to create balanced, unbiased training and testing datasets, where each Seagate model had equal representation of failure and non-failure. The models were trained using 80% of the balanced datasets, reserving 20% for testing on unseen data. Any categorical fields, such as the HDD model and capacity, were converted to numerical values using ordinal and one-hot encoders.

The machine learning classifiers implemented in this paper include Random Forests (RF), XGBoost (XGB), Decision Trees (DT), Neural Networks (Multi-Layer Perceptron or MLP), k-Nearest Neighbour (k-NN), and Logistic Regression (LR). These methods were selected as they have shown good performance in other existing works of research. Appropriate hyperparameters were selected for each model using Bayesian optimisation, grid search, or random search with 5-fold cross-validation to measure the combination of parameters that achieved the best mean performance.

Feature importance was assessed for each model, collected if available from the classifier, or measured using permutation. Permutation calculates the decrease in model performance as a result of randomly altering the values of each feature after the model has been trained. If the model performance is not greatly affected by permutations of a feature, then it is assumed that the model does not consider that feature important. Conversely, if the model's performance reduces then the feature is considered important, with larger performance reductions implying a relatively more important feature.

D. *Model Evaluation*

The performance of each model was evaluated by generating failure predictions using the test dataset, and by comparing these predictions to the true failure status of the drives. A confusion matrix of the test predictions allowed for calculation of several evaluation metrics using the True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) values, as shown in Fig. 1.

		Predicted Label	
		0 (Not Failed)	1 (Failed)
True Label	0 (Not Failed)	**TN**	**FP**
	1 (Failed)	**FN**	**TP**

Figure 1. Confusion matrix used to evaluate binary classifiers.

The accuracy and failure detection rate, or True Positive Rate (TPR), were calculated for each classifier at each lookahead window. Accuracy measures the percentage of correct predictions made by the model and the TPR measures the proportion of failed drives that were correctly predicted as failing by the classifier. The accuracy and TPR calculations are as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad TPR = \frac{TP}{TP + FN}$$

In addition to accuracy and TPR, the False Alarm Rate (FAR) was also measured, which calculates the percentage of drives that were incorrectly labelled as failing. Two measurements were used to determine the FAR: the False Positive Rate (FPR), which is the proportion of non-failing drives in the test dataset that were incorrectly labelled as failing; and the False Discovery Rate (FDR), which measures the proportion of predicted drives labelled as failing that are incorrect. The FAR calculations are as follows:

$$FPR = \frac{FP}{FP + TN} \qquad FDR = \frac{FP}{FP + TP}$$

Another measure of performance used to evaluate the models is the Area Under the Receiver Operating Characteristic (AUROC) curve. The ROC curve plots the TPR against the FPR and the area under the curve gives a measure of the model's prediction performance. An AUROC value of 1 represents a perfect classifier, and a value of 0.5 represents the performance obtained by a random classifier. While accuracy is commonly used and important for evaluating the likely real-world benefit of the prediction model, the AUROC represents the goodness of the model. Using the prediction probabilities of belonging to a particular class, rather than the resulting binary label of the classification, the AUROC can give a better indication of model performance and is useful for comparing the performance between different models. Consequently, the AUROC was measured for each classification model, and for each lookahead window, using 5-fold cross-validation of the test datasets to measure the mean prediction score.

IV. RESULTS

The model prediction performance is shown in Table IV, comparing the mean AUROC scores of each of the models and the standard deviation from the 5-fold cross-validation evaluation on test data predictions. The random forest classifier achieved the highest AUROC score of 0.9185±0.0066 at a lookahead window of 0 days. This was followed very closely by the XGBoost classifier, which achieved an AUROC score of 0.9162±0.0066 for the same lookahead window. In all cases, as the prediction horizon increased, the model performance decreased. The worst performing classifier was logistic regression, achieving an

TABLE IV. AUROC AND STANDARD DEVIATION OF FAILURE PREDICTION CLASSIFIERS USING 5-FOLD CROSS-VALIDATION

Method	Lookahead Days (N)			
	0	1	2	7
Random Forest	0.9185±0.0066	0.8976±0.0142	0.8830±0.0092	0.8653±0.0068
XGBoost	0.9162±0.0066	0.8954±0.0126	0.8841±0.0083	0.8653±0.0071
Decision Tree	0.8818±0.0086	0.8648±0.0084	0.8477±0.0132	0.8293±0.0053
Neural Network	0.8721±0.0105	0.8526±0.0132	0.8517±0.0131	0.8254±0.0133
k-NN	0.8617±0.0121	0.8414±0.0111	0.8482±0.0150	0.8176±0.0088
Logistic Regression	0.8484±0.0117	0.8166±0.0135	0.8192±0.0117	0.7871±0.0099

AUROC score of 0.8484±0.0117 at the shortest prediction horizon. The AUROC scores are generally higher in this paper compared to [17], but the prediction performance ranking of classification methods agrees with their findings.

The accuracy of the models generally follows the same trend and rankings as the AUROC scores, as shown in Table V. As accuracy uses the predicted label of the HDDs, and not the prediction probabilities associated with each class, they are lower than the AUROC as expected. Again, random forest and XGBoost performed the best at a lookahead window of 0 days with accuracies of 0.862 and 0.864 respectively.

Although the AUROC and accuracy scores are important evaluators of prediction performance, it is likely that a real-world application would place importance on how well the classifiers predicted failing drives. Therefore, the failure detection rate, also known as the True Positive Rate (TPR), was calculated for each model, and at each prediction horizon, as shown in Table VI. The XGBoost classifier achieved the best failure detection rate at most prediction horizons, successfully identifying 77.6% and 74.8% of failing drives with lookahead windows of 0 and 1 day respectively. It was able to successfully predict the imminent failure of 70.7% of drives 7 days in advance, better than any of the other classifiers. Additionally, any model used in a real-world scenario would require false alarms to be minimised to gain the confidence of users. Also shown in Table VI, the FPR and FDR for each model at each prediction horizon was measured. Random forest and XGBoost show the best ratios between TPR and FAR. At the shortest prediction horizon, the random forest classifier achieves 76.7% TPR with 4.1% FPR, while XGBoost achieves 77.6% TPR with 4.5% FPR.

TABLE V. ACCURACY OF FAILURE PREDICTION CLASSIFIERS ON TESTING DATASETS AT EACH LOOKAHEAD WINDOW (N)

	N	RF	XGB	DT	MLP	k-NN	LR
Accuracy	**0**	0.862	0.864	0.854	0.810	0.801	0.778
	1	0.841	0.844	0.832	0.793	0.788	0.753
	2	0.822	0.822	0.813	0.787	0.786	0.752
	7	0.800	0.804	0.790	0.753	0.753	0.720

The feature importance ranking for each classifier is shown in Table VII, obtained from the model where available, or estimated with feature permutation. Features with smaller values of ranking order indicate more importance to the classification model. Missing values indicate that the feature was not present in the top 5 most important features for that classifier. SMART 187, reported uncorrectable errors, is a key indicator for HDD failure and is the most important feature for almost all classifiers.

TABLE VII. MOST COMMON IMPORTANT FEATURES AMONGST CLASSIFIERS INDICATED BY FEATURE RANKING ORDER

	Feature Ranking Order of Importance if Present in Top 5 Most Important Features					
	RF	DT	XGB	MLP	k-NN	LR
SMART 5	2	3	4	3	**1**	2
SMART 187	**1**	**1**	**1**	**1**	4	**1**
SMART 197	3	2	3	4	2	-
SMART 198	4	-	2	2	3	-
SMART 240	-	5	-	-	-	-
SMART 241	5	-	-	-	-	-
SMART 242	-	4	-	5	-	-

TABLE VI. TRUE POSITIVE RATE (TRP), FALSE POSITIVE RATE (FPR) AND FALSE DISCOVERY RATE (FDR) OF CLASSIFIERS AT EACH LOOKAHEAD WINDOW (N)

	N	RF	XGB	DT	MLP	k-NN	LR
TPR	0	0.767	0.776	0.759	0.761	0.682	0.599
	1	0.738	0.748	0.746	0.728	0.669	0.566
	2	0.707	0.717	0.695	0.726	0.681	0.582
	7	0.689	0.707	0.701	0.689	0.636	0.507
FPR	0	0.041	0.045	0.049	0.139	0.077	0.040
	1	0.052	0.058	0.078	0.141	0.089	0.056
	2	0.061	0.072	0.066	0.150	0.106	0.074
	7	0.085	0.095	0.118	0.182	0.130	0.065
FDR	0	0.049	0.054	0.060	0.153	0.100	0.062
	1	0.064	0.070	0.092	0.160	0.116	0.088
	2	0.078	0.090	0.086	0.168	0.132	0.111
	7	0.107	0.116	0.140	0.208	0.169	0.114

Comparing these results with Table III, the four features with the highest correlation coefficient are the top four most important features in Table VII.

V. DISCUSSION

The classification approach adopted in this paper achieved relatively high levels of prediction performance. It highlighted the most accurate machine learning classifiers for failure prediction using a common dataset of HDDs from an operational data centre. It proved that SMART metrics can be used as an indication of imminent failure, with some more useful than others. However, there are some limitations and weaknesses that need to be highlighted.

As HDDs are constructed of both mechanical and electrical components, there are a number of potential reasons for their failure. Over time the subcomponents within the hard drive unit can degrade, causing problems with reading and writing data, and eventually leading to failure. In this scenario, where the drives fail slowly with accumulated usage and workload, SMART metrics may be a good indicator of HDD health. However, hard disk drives are susceptible to external factors, such as physical disturbances from knocks and vibrations, water damage, and power-related problems, including voltage spikes. It is therefore unlikely that SMART metrics would indicate imminent failure for quick-failing drives due to external factors. With the best TPR of 77.6% achieved in this paper, at least 22.4% of failed drives were not detected. Without knowing the root cause of failure, it could be possible that those drives did not contain any indication of failure in their SMART metrics.

The decision to approach the research as a binary classification problem means that the prediction only has the option to label a drive with a failure status of failing or non-failing. However, the health of HDDs may indicate that the drive is at low, moderate, or high risk of failing, in which case a multi-class classification approach might be better suited. Using the same logic, it may be desirable to predict the probability of drive failure. In the case of the binary classification approach, if a drive is predicted to have a 51% chance of failing it would be labelled as a failing drive (assuming a 50% threshold). However, data centre maintainers may dismiss that drive as a low risk if they were presented with the probability of failing, whereas they would be forced to investigate the drive if the binary classification prediction was presenting its impending failure. Hence, the AUROC was used in this project to better evaluate the classifiers' prediction performance using the prediction probabilities of belonging to a particular class.

A well-known issue with SMART metrics is that the data they report isn't always consistent between various manufacturers and drive models. Some attribute fields may be used to record a particular measurement by one manufacturer, but a completely different measurement by another. And the format or scale used may not be consistent even when reporting the same measurement. Therefore, to mitigate against this, the drive models used in the ML implementation of this work only considered drives of a single manufacturer, Seagate. Consequently, it is not guaranteed that the prediction classifiers would generalise well for predicting failures or other manufacturers' drives.

The prediction horizons examined in this paper use the SMART attribute measurements from 0, 1, 2, and 7 days prior to HDD failure. The prediction performance improved as the lookahead days decreased, with 0 days achieving the best AUROC, accuracy, TPR, and FAR rates. The 0-day lookahead window means that the drives failed sometime in the next 0-24 hours. In a real-world application it is likely that this prediction window is too short, not allowing for enough time to act. Increasing the window would decrease the prediction accuracy, potentially reducing the likelihood for users to trust the classification output.

The features used to train and evaluate the ML classifiers consisted only of the raw and normalised SMART measurements with scaling applied. As indicated in [9], the rate of change of SMART measurements over a given period can provide additional features that potentially improve the prediction performance of HDD failure prediction models. The work presented in this paper would benefit by extending to include temporal disparities of SMART measurements as features for machine learning.

Further work may include extending the scope to compare the prediction performance of the classification methods implemented here with other methodologies, such as time series prediction, and analysing the AUROC, accuracy, TPR, and FAR at varying prediction horizons.

VI. CONCLUSSION

In this paper, the SMART attributes of operational HDDs in a large data centre were analysed with respect to drive failure. SMART attributes 5, 187, 197, and 198 (reallocated sectors count, reported uncorrectable errors, current pending sector count, and uncorrectable sector count) were observed to have the highest correlation with HDD failure.

A subset of the SMART attribute measurements, reported daily by the data centre HDDs, was used to create machine learning classifiers for drive failure prediction. The ML classification models implemented in this work include Random Forest, XGBoost, Decision Tree, Neural Network (Multi-Layer Perceptron), k-Nearest Neighbour, and Logistic Regression methods. The SMART metrics were collected at 0, 1, 2, and 7 days prior to drive failure to evaluate the prediction performance at multiple prediction horizons. It was found that as the prediction horizon decreases, the performance of the failure prediction increased for all classifiers.

Random Forest and XGBoost classifiers achieved the best results, with 86% prediction accuracy and 4-5% False Alarm Rate (FAR) at the shortest prediction horizon. The failure detection rate ranged from 67% when making predictions 7 days prior to HDD failure, to 77% when using the SMART measurements recorded in the last 24 hours before failure. The AUROC was calculated to make better comparisons between the classifiers, which again showed Random Forest and XGBoost as the best performing, with AUROC scores of 0.9185±0.0066 and 0.9162±0.0066 respectively at the shortest prediction horizon.

The relative feature importances of the ML models were obtained, either directly from the classifier or estimated using feature permutation. It was found that, in all but one of the classifiers, SMART 187 was regarded as the most important metric for predicting HDD failure. The top four most important features across all classifiers were those with the highest Spearman rank correlation coefficient relating to failure as described above (SMART 5, 187, 197, and 198).

The classification models generated in this work could benefit from future advancements, and enhanced feature engineering would likely improve the performance of the models. Additionally, SMART attributes do not account for many of the external factors that can affect the health of HDDs, such as physical disturbances. Therefore, it may be valuable to extend the work of this paper by considering other relevant datasets alongside SMART data for predicting failure. For example, force sensor data may indicate knocks or jolts to the HDD, and [18] has shown success in using machine learning to classify force signals and determine if a collision occurred. Other future work may include incorporating the classification models with autonomic computing, where the failure predictions can inform autonomic actions. Such actions may involve pre-emptively backing up data to another storage device to mitigate the risk of data loss.

References

[1] A. Lawrence and L. Simon, “Annual Outage Analysis 2023,” Uptime Institute, New York, NY 10174, 2023. [Online]. Available: https://uptimeinstitute.com/resources/research-and-reports/annual-outage-analysis-2023. [retrieved: Feb. 2025].

[2] J. Davis, D. Bizo, A. Lawrence, O. Rogers, and M. Smolaks, “Global Data Center Survey 2022,” Uptime Institute, New York, NY 10174, 2022. [Online]. Available: https://uptimeinstitute.com/resources/research-and-reports/uptime-institute-global-data-center-survey-results-2022. [retrieved: Feb. 2025].

[3] K. V. Vishwanath and N. Nagappan, “Characterizing Cloud Computing Hardware Reliability,” in Proceedings of the 1st ACM Symposium on Cloud Computing, Indiana, IN, USA, 2010, pp. 193–204.

[4] G. Wang, L. Zhang, and W. Xu, "What Can We Learn from Four Years of Data Center Hardware Failures?," in 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 2017, pp. 25-36.

[5] L. Perri, “What’s New in the 2022 Gartner Hype Cycle for Emerging Technologies,” Gartner, 2022. [Online]. Available: https://www.gartner.com/en/articles/what-s-new-in-the-2022-gartner-hype-cycle-for-emerging-technologies. [retrieved: Feb. 2025].

[6] J. O. Kephart and D. M. Chess, "The vision of autonomic computing", Computer, vol. 36, no. 1, pp. 41-50, Jan. 2003, doi: 10.1109/MC.2003.1160055.

[7] R. Sterritt, (2005), “Autonomic computing”. Innovations Syst Softw Eng: A NASA Journal, 1, 79–88 (2005). doi:10.1007/s11334-005-0001-5

[8] J. Li et al., "Hard Drive Failure Prediction Using Classification and Regression Trees," in 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA, USA, 2014, pp. 383-394.

[9] Z. Miller, O. Medaiyese, M. Ravi, A. Beatty, and F. Lin, "Hard Disk Drive Failure Analysis and Prediction: An Industry View," in 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S), Porto, Portugal, 2023, pp. 21-27.

[10] I. C. Chaves, M. R. P. de Paula, L. G. M. Leite, J. P. P. Gomes, and J. C. Machado, "Hard Disk Drive Failure Prediction Method Based On A Bayesian Network," in 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-7.

[11] I. C. Chaves, M. R. P. de Paula, L. G. M. Leite, L. P. Queiroz, J. P. P. Gomes, and J. C. Machado, "BaNHFaP: A Bayesian Network Based Failure Prediction Approach for Hard Disk Drives," 2016 5th Brazilian Conference on Intelligent Systems (BRACIS), Recife, Brazil, 2016, pp. 427-432.

[12] C. Xu, G. Wang, X. Liu, D. Guo, and T. -Y. Liu, "Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks," in IEEE Transactions on Computers, vol. 65, issue. 11, pp. 3502-3508, 1 Nov. 2016.

[13] S. Pang, Y. Jia, R. Stones, G. Wang, and X. Liu, "A combined Bayesian network method for predicting drive failure times from SMART attributes," in 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 4850-4856.

[14] A. Coursey, G. Nath, S. Prabhu, and S. Sengupta, "Remaining Useful Life Estimation of Hard Disk Drives using Bidirectional LSTM Networks," in 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 4832-4841

[15] Q. Hai, S. Zhang, C. Liu, and G. Han, "Hard Disk Drive Failure Prediction Based on GRU Neural Network," in 2022 IEEE/CIC International Conference on Communications in China (ICCC), Sanshui, Foshan, China, 2022, pp. 696-701.

[16] Backblaze. Hard Drive Data and Stats [Online]. Available: https://www.backblaze.com/cloud-storage/resources/hard-drive-test-data. [retrieved: Jan. 2024].

[17] R. Pinciroli, L. Yang, J. Alter, and E. Smirni, "Lifespan and Failures of SSDs and HDDs: Similarities, Differences, and Prediction Models," in IEEE Transactions on Dependable and Secure Computing, vol. 20, issue. 1, pp. 256-272, 2023.

[18] A.-N. Sharkawy, A. Ma’arif, Furizal, R. Sekhar, and P. Shah, “A comprehensive pattern recognition neural network for collision classification using force sensor signals,” Robotics, vol. 12, issue. 5, p.124, 2023.

Hardware-in-the-loop Experiments on an Active Turning Tool with Robust LPV Control of Chatter Vibration

Ziv Brand
Dept. Mechanical Engineering
Shamoon College of Engineering, SCE
Be'er Sheva, Israel
e-mail: zivbr@sce.ac.il

Matthew O.T. Cole
Dept. Mechanical Engineering
Chiang Mai University, CMU
Chiang Mai, Thailand
e-mail: motcole@dome.eng.cmu.ac.th

Abstract— **This study presents a control design for an active lathe cutting tool aimed at effectively suppressing vibration and chatter during internal turning operations. To address the variability in dynamic behavior caused by uncertainties in cutting parameters, a robust H_∞ controller was developed using a Linear-Parameter-Varying (LPV) model of the machining dynamics. This parametric approach allows real-time tuning of the control system properties in a workshop environment for optimal vibration reduction. The controller manages time-delayed feedback from cutting forces and mitigates spillover instability from unmodeled high-frequency modes. Integrated with piezoelectric actuators and sensors, the active tool holder maintains its original size and shape, ensuring ease of use in industrial applications. Experiments using a laboratory system that emulates chatter showed that the proposed system reduces root-mean-square vibration levels by over 65%, with peak-to-peak values reduced by approximately 47% in unstable cutting regimes. These results demonstrate the system's feasibility for real-world machining environments, enabling higher material removal rates with improved surface finish.**

Keywords*- Active vibration control; Chatter suppression; Internal turning; Linear parameter varying control; H-infinity control.*

I. INTRODUCTION

Machining processes are crucial in the production of high-precision components, with the global machining market projected to reach approximately USD 100 billion by 2025, based on a Compound Annual Growth Rate (CAGR) of 7% between 2019 and 2025. CNC lathes represent around 20% of all machine tools, playing a significant role in modern manufacturing [1]. However, mechanical vibrations, particularly chatter, remain a persistent challenge in these processes. Chatter is a type of self-excited vibration that leads to poor surface finish quality, increased tool wear, and elevated noise levels. In Computer Numerical Control (CNC) lathes, avoiding chatter typically requires setting conservative values of machining parameters such as reduced cutting depth, slower spindle speeds, and decreased feed rates, all of which lower production efficiency [2].

Additionally, the avoidance of vibration problems in internal turning operations imposes limits on tool holder geometry. As the ratio between tool overhang and diameter (L/D, Length-to-Diameter ratio) increases, the stability margins decrease due to reduced bending stiffness [2]. This limitation restricts the length of standard tools, with current solutions ranging from L=4D to 14D depending on what materials and vibration control methods are used. For instance, the length of steel shank tools is limited to 4D, while solid carbide tools can reach 7D, and passive vibration-controlled tools can achieve up to 10D. Solid carbide tool shanks with passive control devices can reach up to 14D [3].

This study explores the significant potential of tool-integrated active vibration control to reduce chatter in internal turning. We focus on the development of a robust LPV feedback control system, which is demonstrated and evaluated through a laboratory emulator system. The experimental setup incorporates a piezoelectric actuator, and a PZT-type vibration sensor mounted to a standard turning tool. The goal is to create a control system that not only enhances vibration suppression but also allows the flexibility of on-the-job tuning to optimize performance according to the setup conditions.

Piezoelectric materials can operate as transducers in a dual manner: the direct piezoelectric effect involves the conversion of mechanical forces into electrical potential, while the inverse piezoelectric effect can convert an applied electric field into mechanical deformation of the material [4]. Piezoelectric components are widely used for active vibration suppression. Examples include vibration reduction of thin-walled rotors [5][6], cables [7], composite laminated plates [8], aerospace applications [9], and research for CNC lathes [10][11].

Various active control approaches for suppressing vibrations in cutting tool holders have been studied and are still actively being researched. These include the use of PID controllers [12], H-infinity state feedback control strategies [10], H-infinity controller design based on Linear Matrix Inequalities (LMI) [13], model-free finite frequency H_∞ control [11], and input shaping control [14], among others. The model-based design of a robust control algorithm for vibration suppression in the turning process presents many challenges. In this work, the effect of parametric uncertainties and time-delay feedback—both inherent in machining processes—are taken into consideration.

To address the uncertain dynamics, a robust parameter-dependent H_∞ controller is synthesized, where the robustness/gain properties can be adjusted in real-time within a workshop environment to achieve optimal vibration reduction, making it more practical in machining processes

ISBN: 978-1-68558-241-8

subject to parametric uncertainty. The controller design considers uncertain time-delayed feedback effects from cutting forces and aims to prevent spillover instability of unmodeled and uncertain higher-order modes. Laboratory experiments conducted using a cutting emulation system demonstrate that the proposed approach can successfully transition the system from unstable to stable operation achieving a vibration reduction of approximately 65%.

Within the paper, Section II defines the dynamic model of the active tool system and presents the methodology for designing the robust H_∞ controller based on the LPV framework. Section III describes the experimental setup, followed by Section IV, which provides the practical evaluations and results from the turning emulation tests conducted in the laboratory. Section V concludes the paper with key findings and recommendations for future work.

II. Dynamic Model

This section presents the mathematical formulation and modeling approach used to describe the tool's dynamic response during internal turning operations.

A. *Turning process model*

The internal turning process involves dynamic interactions between the tool structure and the rotating workpiece at the material removal point. Modeling principles for these dynamics have been established in previous studies [15][16]. For practical approximation, the forced excitation of the tool can be effectively modeled using a single-degree-of-freedom dynamic representation that captures the dominant bending mode of the tool holder. With the inclusion of actuation force, a lumped mass model is found to be suitable:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = Ku(t) + f_c(t) \tag{1}$$

where f_c represents the radial cutting force, u is the control input (voltage), and K is the coupling coefficient for the piezoelectric actuator. The parameters m, c and k denote the effective mass, damping, and stiffness of the dominant mode, respectively. Converting this expression into the Laplace domain gives:

$$X(s) = \Phi(s) \cdot \left(K_1 \cdot U(s) + K_2 \cdot F_c(s)\right) \tag{2}$$

where $\Phi(s) = \frac{1}{s^2+2\zeta\omega_n s+\omega_n^2}$ with $\omega_n = \sqrt{k/m}$ being the natural frequency, and $\zeta = c/\left(2\sqrt{k/m}\right)$ the damping ratio. The static gains are $K_1 = K/m$ and $K_2 = 1/m$.

To predict chatter, the cutting force model must connect the time-varying tool deflection $x(t)$ to the instantaneous chip thickness $h(t)$ and corresponding cutting force $f_c(t)$. In longitudinal cutting, an overlap factor $\Psi = b_d/b$ is used to account for the influence of previous tool deflections on the current chip thickness [17]. This concept is illustrated in Figure 1 for a standard bull-nose end cutter. The effective chip width b is determined by the cutting-edge shape, while the overlap length b_d depends on the feed per revolution. For small radial deflections $x(t)$, the instantaneous chip thickness can be expressed as:

$$h(t) = h_m(t) - [x(t) - \Psi x(t-\tau)] \tag{3}$$

where $h_m(t)$ is the mean chip thickness over the width of the cut, and τ is the time delay from one period of spindle rotation: $\tau = 60/\Omega$ where Ω is the rotational speed in rpm. The radial cutting force can be expressed

$$f_c(t) = f_{tr}(t)\cos\beta + C_s \frac{b}{V}\dot{x}(t), \tag{4}$$

where $f_{tr}(t) = K_s bh(t)$ is the cutting force projected onto the $x - y$ plane, β is the angle between the cutting force and radial direction, C_s is the process damping, and $V = \pi d\Omega/60$ is the cutting velocity, where d is the workpiece diameter, and K_s is the specific cutting force appropriate to the material properties and cutting conditions. The turning process model is represented by the block diagram in Figure 2, where $e^{-\tau s}$ represents the time delay.

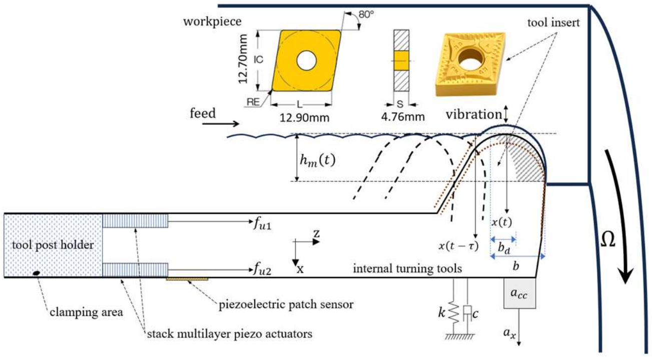


Figure 1. Schematic diagrams of internal turning process.

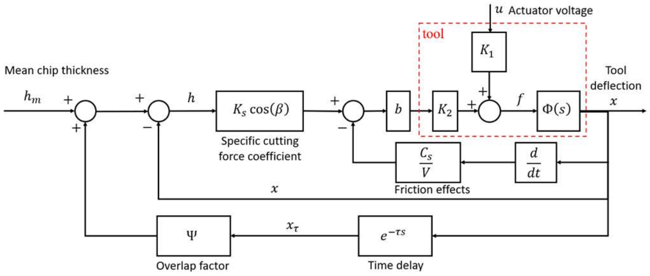


Figure 2. Turning process dynamics with active tool holder.

B. *Robust stability and controller synthesis for cutting processes with delayed feedback*

The optimization of the feedback control algorithm must balance high performance with avoiding excessive controller gain, which could lead to actuator voltage saturation, system instability, or noise amplification. Robust controller design must account for potential differences between the actual system dynamics $G_0(s)$ and the model transfer function $G(s)$. This can be represented by a multiplicative error factor $\Delta_m(s)$

Copyright (c) IARIA, 2025.   ISBN: 978-1-68558-241-8

such that $G_0 = G + \Delta_m G$. When a bound for Δ_m is known or estimated, the small gain theorem can be used to ensure stability and performance of the closed-loop system.

A key challenge is that model accuracy can vary depending on the tool setup, and precise determination of Δ_m, or its bounds, may not always be feasible. Additional uncertainty arises in the cutting force model due to factors like material hardness, cutting speed, and feed rate. Therefore, a systematic approach is needed to handle both model and cutting process uncertainties in the controller design.

The system dynamics for controller synthesis are depicted in Figure 3, where the multiplicative model error (Δ_m) and the time-delayed feedback block ($e^{-\tau s}$) are treated as uncertain external feedback loops. The lumped parameter $\kappa = K_2 K_s b \cos\beta / K_1$ is both uncertain and potentially time-varying, depending on the current cutting conditions. Designing a controller for the maximum expected value of κ may ensure suitability for the maximum depth of cut but could compromise robustness. Instead, a κ-dependent controller is proposed, which can be tuned online to match system setup and cutting conditions. This synthesis is done using the gain-scheduled H_∞ framework, which requires the plant model's state-space matrices to have an affine dependency on the parameters, i.e., be in LPV form. The parameter $\delta = C_s/(V \cos\beta\, K_s)$, representing process damping is also uncertain; however, designing the controller for the smallest value of δ is appropriate as larger values increase damping and improve stability.

The LPV control synthesis problem is defined by considering d and x_τ as exogenous inputs, and f and x as outputs, as shown in Figure 3. The impact of Δ_m and the delayed feedback ($e^{-\tau s}$) are handled using H_∞ norm-bound criteria for robust stability [18, 19]. Three design specifications are defined:

1. Robust Stability under Delayed Feedback:

$$\left\|T_{xx_\tau}(K_c,\kappa)\right\|_\infty < 1 \leftrightarrow \|\kappa T_{xd}(K_c,\kappa)\|_\infty < \Psi^{-1}.$$

2. Forced Disturbance Attenuation:

$$\left\|W_1 T_{xh_m}(K_c,\kappa)\right\|_\infty < 1 \leftrightarrow \|\kappa W_1 T_{xd}(K_c,\kappa)\|_\infty < 1,$$

where $W_1(s)$ is a weighting function representing the disturbance spectrum.

3. Robust Stability under Model Error (Δ_m):

$$\left\|\Delta_m T_{fd}(K_c,\kappa)\right\|_\infty < 1 \leftarrow \|W_2 T_{ud}(K_c,\kappa)\|_\infty < 1,$$

where $W_2(s)$ is chosen to satisfy $|W_2(j\omega)| > |\Delta_m(j\omega)|\ \forall\ \omega$.

Since specification 2 with $|W_{21}(j\omega)| > \Psi\ \forall\ \omega$ is sufficient to achieve specification 1, all three can be unified into a single H_∞ norm-bound criterion. The block diagram for the weighted plant is shown in Figure 4.

The LPV control synthesis can be formulated using LMIs, solvable through convex optimization techniques [20]. This formulation applies to systems with state-space equations in the general form:

$$\dot{x} = A(\kappa)x + B_w(\kappa)w + B_u u$$
$$z = C_z(\kappa)x + D_{zw}(\kappa)w + D_{zu}u \qquad (5)$$
$$y = C_y x + D_{yw}w + D_{yu}u$$

where the matrices are affine in a parameter κ that varies within a fixed interval. The LPV controller solution (A_c, B_c, C_c, D_c) is similarly parameter-dependent:

$$K_c(\kappa) = [A_c(\kappa), B_c(\kappa), C_c(\kappa), D_c(\kappa)] \qquad (6)$$

The controller is synthesized by solving the LMIs for the set of vertex systems, producing vertex controllers to form the parameter-dependent controller matrices. For this problem, the controller can be expressed as:

$$K_c(\kappa) = K_c(\underline{\kappa}) + \alpha\big(K_c(\bar{\kappa}) - K_c(\underline{\kappa})\big) \qquad (7)$$

where the scheduled parameter is $\alpha = \big(\kappa - \underline{\kappa}\big)/\big(\bar{\kappa} - \underline{\kappa}\big) \in (0{,}1)$. Note that α will be the adjustable parameter for the controller.

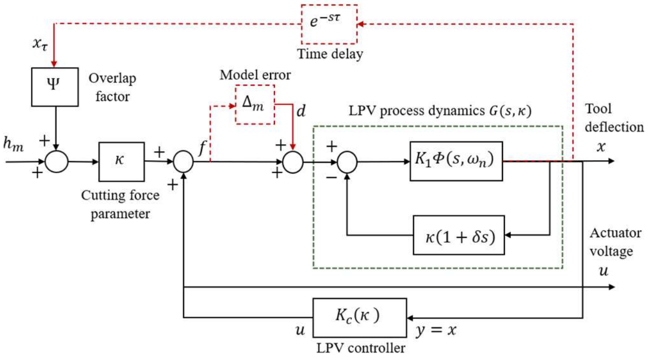


Figure 3. System definition for robust LPV control synthesis.

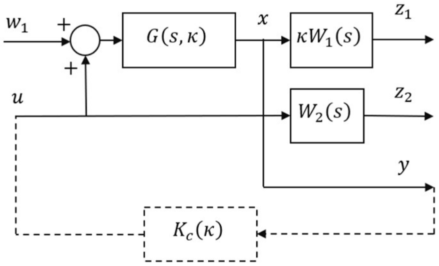


Figure 4. LPV plant definition with weighting functions.

The multiplicative model error (as shown in Figure 5) was calculated based on the error between the single-degree-of-freedom dynamic model (Eq. 1), obtained using the peak picking method for identifying the modal parameters of the first mode, and the FRF results from frequency response testing. Simultaneously, sensitivity calibration was performed around the first resonance frequency, converting from voltage units to meters by matching the piezoelectric bending sensor with the accelerometer data. The results of this stage are presented in Figure 6. The transfer function from the PEA input (in volts) to the PES measurement (in µm) was thus determined as:

$$K_1\Phi(s) = \frac{X(s)}{U(s)} = \frac{K_1}{s^2+12s+(2\pi\times 400)^2} \qquad (6)$$

where $K_1 = 4.161\,\mu m/\mathrm{Vs}^2$ and $K_2 = 2.575\,\mu m/\mathrm{Ns}^2$. For the tested design case, the maximum value of the force coefficient was set to $\overline{\kappa} = K_2 K_s b \cos\beta / K_1 = 0.22\ \mathrm{V}/\mu m$ (based on $K_s b = (7.5 \times 10^8) \times (1 \times 10^{-3})\ \mathrm{N/m}$ and $\beta = 15°$).

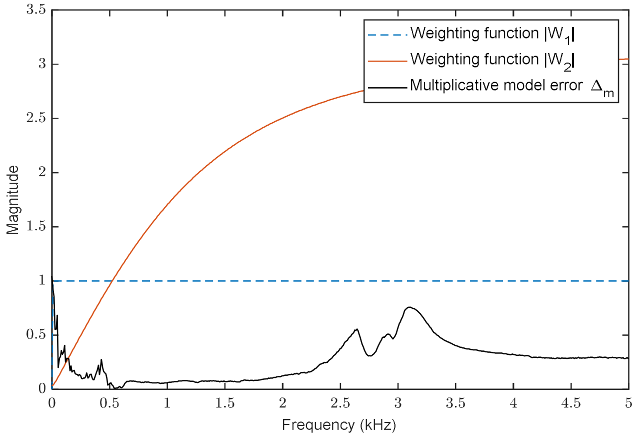


Figure 5. Design weighting functions and model error.

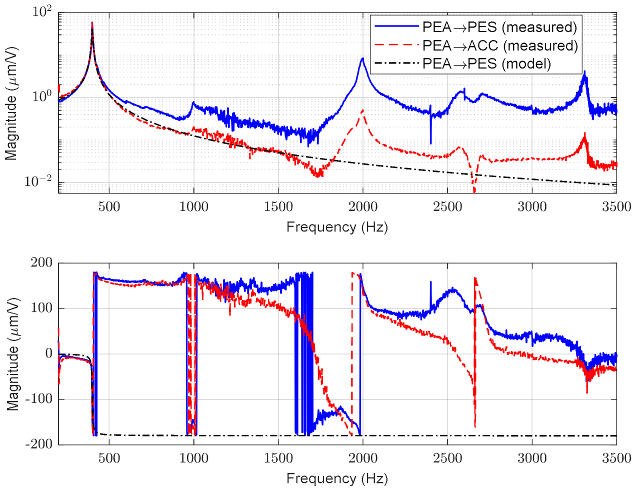


Figure 6. Frequency response measurements from sine-sweep tests with PEA excitation, alongside model-based data for comparison (after tuning).

To align with the robust stability condition outlined in design specification 1 and to reduce the complexity of the synthesis process, $W_1(s)$ was selected as a flat weighting with a unity gain scaling. In contrast, $W_2(s)$ was chosen to bound the model error for $G(s)$, based on the FRF measurement results.

The transfer functions used for robust LPV control synthesis, along with the vertex LPV plant models, are shown in Figure 7. The design results demonstrate that the approach enables real-time tuning of a critical parameter, allowing adjustment of the trade-off between achieving high performance and avoiding excessively large controller gains.

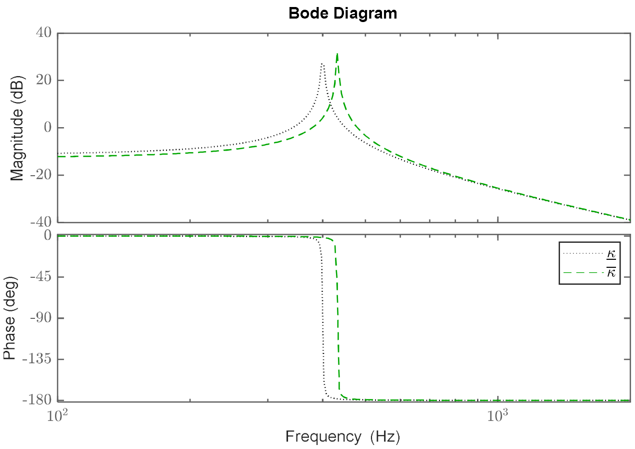


Figure 7. Transfer functions for robust LPV control with vertex plant models.

III. Experimental System

The active tool system, shown in Figure 8, was constructed using a commercially available internal turning tool holder (model S32T-MCLNL12 from K. CUTTER), measuring 300 mm in length with a 32 mm diameter. Actuation was achieved through two multilayer piezoceramic stacks (model P-888.31 from PI Ceramic GmbH), each measuring 10 × 10 × 17 mm, with a nominal displacement range of 11 µm and a blocking force of 3500 N. To ensure proper dynamic operation, a preload of 15 MPa was applied via a wedge mechanism with a preload screw. The actuator slots were machined near the clamping area, resulting in an overhang of 180 mm and an L/D ratio of 5.625.

A piezoelectric patch sensor (model P-876.SP1 from Physik Instrumente (PI) GmbH) was installed to measure longitudinal strain, which correlates with radial bending of the tool holder, and was positioned near the actuators. For sensor calibration, an IEPE accelerometer (model CA-YD-1181 from Sinocera) was mounted near the tool insert in the radial direction.

The experiments utilized a cutting force emulator system in a laboratory hardware setup of the tool, as shown in Figure 8. The control block diagram is depicted in Figure 9. The setup included two main real-time subsystems: one for chatter force emulation and the other for active vibration control. Chatter emulation was performed using an electromagnetic actuator powered by an Escon 24/2 amplifier from Maxon Motor, operating in current control mode. The pair of piezoelectric actuators were driven differentially using two piezo amplifiers (model E-617 from PI Ceramic GmbH).

Data acquisition and control were managed using a real-time target machine from Speedgoat, equipped with an IO397 analog input/output card. Both the control algorithm and chatter emulator were implemented in Simulink and MATLAB. A High-Pass Filter (HPF) with a cutoff frequency of 30 Hz was applied to prevent the controller from compensating for quasi-static measurements. Additionally, a real-time user interface enabled online adjustment of controller parameters. The upper part of Figure 9 shows the chatter emulator, which incorporates chip processing parameters and simulates the chatter force using the electromagnetic force driven by the current amplifier.

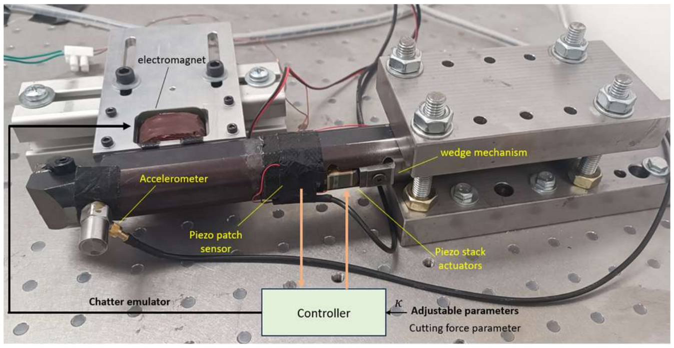


Figure 8. Experimental setup.

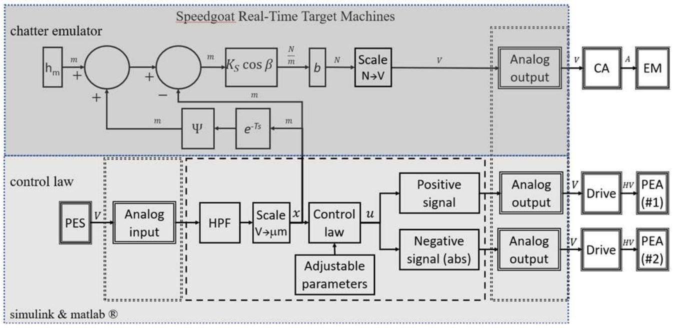


Figure 9. Block diagram of experimental implementation hardware.

IV. Experimental Results

Turning emulation experiments were conducted using the experimental setup (Figure 8). The gain-scheduled H_∞ control law was synthesized using MATLAB. To demonstrate the control effectiveness in a laboratory setting, several key results are presented here. Figure 10 shows the measured FRF of the internal cutting tool for the uncontrolled case, along with three cases using robust LPV control with the two vertex models and one interpolation case. The results support the design and demonstrate that high damping was achieved in the system's resonant frequency range.

Time-domain results from the chatter emulator are shown in Figure 11 for three different cases: Case 1 (red) and Case 2 (black) represent unstable machining conditions without and with control, respectively. Case 3 (blue) represents stable machining parameters without control. Table 1 presents quantitative measures for these cases and an additional Case 4 (without the chatter emulator), which yields results similar to Case 3, thereby confirming the absence of chatter in Case 3. The results show approximately a 65% reduction in chatter vibration (in terms of Root-Mean-Square, RMS value) and about a 47% reduction in the peak value. The value of the dynamic chatter force (f_c) also decreases from Case 1 to Case 2 due to the reduction in the chip thickness parameter (emulated), where PEA is a piezoelectric actuator and PES is a piezoelectric sensor. Quantitative and comparative results with and without control in an internal turning machine were validated in a subsequent study [21], demonstrating improvements consistent with the trends observed here under laboratory conditions with the emulator.

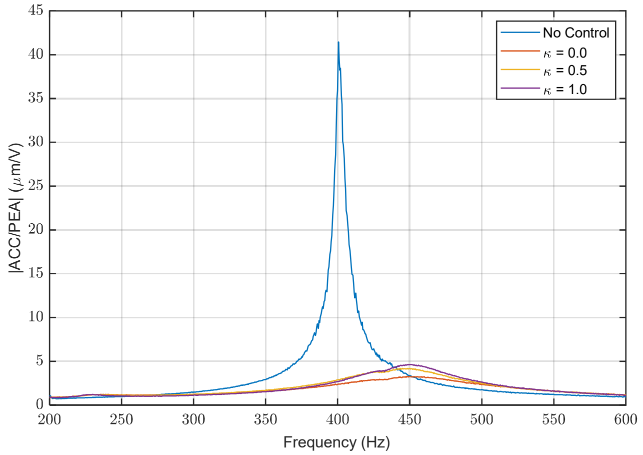


Figure 10. Measured FRF for LPV control with vertex models.

TABLE I. Quantitative Measures

Parameter	*Case 1*	*Case 2*	*Case 3*	*Case 4*	*Units*
PES(rms)	1.82	0.63	0.383	0.379	μm
PES(max)	5.88	3.10	0.910	0.855	μm
f_c (rms)	9.34	2.87	0.157	0.000	N
f_c(max)	29.68	13.77	0.633	0.000	N
Control(rms)	0.00	0.32	0.000	0.000	V
Control(max)	0.00	1.19	0.000	0.000	V

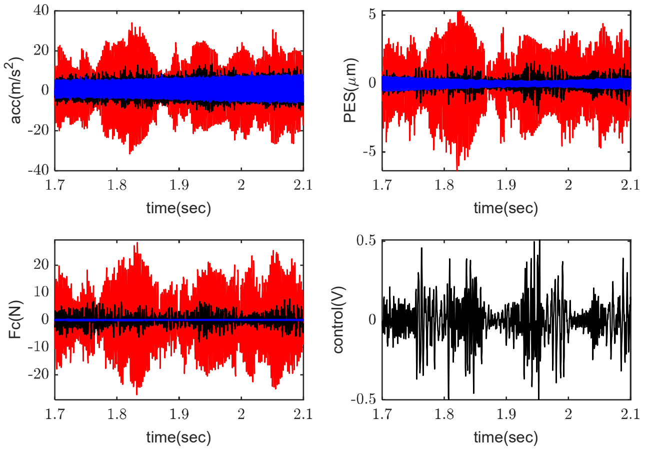


Figure 11. Experimental results for emulator turning: (red) case 1: unstable cutting regime without control, (black) case 2: control applied in unstable cutting regime, (blue) case 3: stable cutting regime withought control.

V. Conclusions

This study investigated the effectiveness of a robust gain-scheduled H_∞ control strategy for suppressing chatter in internal turning processes. Experimental validation using an emulator setup showed that the proposed controller significantly reduced vibrations in the system's unstable operating region, achieving a reduction of approximately 65% in RMS values and 47% in peak values of chatter forces. These results confirm the practical feasibility of the tunable real-time controller in machining environments, including adaptation to varying cutting conditions.

The research emphasizes the importance of addressing both model uncertainties and time-delayed feedback in controller design. By employing LPV control synthesis with vertex models, the system could maintain robust performance across different machining setups, effectively mitigating instability without compromising operational efficiency

Future work will focus on validating the control design through practical turning experiments in a workshop environment. Additionally, the control algorithm will be extended to more realistic scenarios, such as when the turning tool is connected to CNC machines at varying lengths, causing changes in resonance frequency. This extension will further enhance the adaptability and effectiveness of the control strategy in industrial applications.

Acknowledgment

This research was supported by Israel Innovation Authority Fund 2024, Israel [83647] and by Chiang Mai University Fundamental Fund 2024 [FF041/2567].

References

[1] R. Ye, "CNC Machining industry trends," 3ERP, Jun 2019. [Online]. Available from: https://www.3erp.com/blog/cnc-machining-industry-trends-2019 [retrieved: Jan, 2025].

[2] R. P. Siddhpura, "A review of chatter vibration research in turning," International Journal of Machine Tools and Manufacture, vol. 61, pp. 27-47, 2012.

[3] ISCAR, "Anti-vibration boring bars with exchangeable turning heads," [Online]. Available from: https://www.iscar.se/newarticles.aspx/countryid/15/newarticleid/2344 [retrieved: Jan, 2025].

[4] A. Preumont, "Mechatronics: dynamics of electromechanical and piezoelectric systems," Springer Netherlands, Germany, 2006.

[5] Z. Brand and M.O.T. Cole, "Piezo-based flexural vibration suppression for an annular rotor via rotating-frame H2 control optimization,". Journal of Intelligent Material Systems and Structures, vol. 33, no. 4, pp. 572-589, 2022.

[6] Z. Brand and M.O.T. Cole, "Mini-max optimization of actuator/sensor placement for flexural vibration control of a rotating thin-walled cylinder over a range of speeds," Journal of Sound and Vibration, vol 506, p. 116105, 2021.

[7] Y. Wu, Y. Zhang, X. Guo, et al., "Active control of cable vibration using piezoelectric actuators considering strong electric field nonlinearity," Journal of Vibration Engineering and Technologies, vol. 12, pp. 935-947, 2024.

[8] T. Liu, C. Liu, and Z. Zhang, "Adaptive active vibration control for composite laminated plate: Theory and experiments," Mechanical Systems and Signal Processing, vol 206, p. 110876, 2024.

[9] A. Preumont, "Active damping, vibration isolation, and shape control of space structures: A Tutorial," Actuators, vol. 12, p. 122, 2023.

[10] J. Chen, H. Ma, Z. Liu and Z. Liu, "Finite-frequency H_∞ control for active chatter suppression in turning,". International Journal of Advanced Manufacturing Technology, vol. 129, pp. 5075–5088, 2023.

[11] G. Han, H. Ma, Y. Liu, Z. Liu, and Q. Song, "Model-free finite frequency H_∞ control for active chatter suppression in turning," Journal of Sound and Vibration, vol. 577, p. 118342, 2024.

[12] M. Chen and C.R. Knospe, "Control approaches to the suppression of machining chatter using active magnetic bearings," IEEE Transactions on Control Systems Technology, vol. 15, no. 2, pp. 220–232, Mar. 2007.

[13] E. Mizrachi, S. Basovich, and S. Arogeti, "Robust time-delayed H_∞ synthesis for active control of chatter in internal turning," International Journal of Machine Tools and Manufacture, vol. 158, p. 103612, 2020.

[14] M. Kasprowiak, A. Parus, and M. Hoffmann, "Vibration suppression with use of input shaping control in machining," Sensors, vol. 22, p. 2186, 2022.

[15] Y. Altintas and A.A. Ber, "Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design," Applied Mechanics Reviews, vol. 54, no. 5, p. B84, 2001.

[16] T. Schmitz and K. Smith, "Machining dynamics: frequency response to improved productivity," 2nd ed., 2019.

[17] L. Zhang, X. Wang, and S. Liu, "Analysis of dynamic stability in a turning process based on a 2-DOFs model with overlap factor," Journal of Mechanical Science and Technology, vol. 26, no. 6, pp. 1891–1899, 2012.

[18] M. Chen and C.R. Knospe, "Control approaches to the suppression of machining chatter using active magnetic bearings," IEEE Transactions on Control Systems Technology, vol. 15, no. 2, pp. 200–232, 2007.

[19] P. Ruttanatri, M.O.T. Cole, R. Pongvuthithum, and S. Huyanan, "H-infinity controller design for chatter suppression in machining based on integrated cutting and flexible structure model," Automatica, vol. 130, p. 109643, 2021.

[20] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory," SIAM, 1994.

[21] Z. Brand, M.O.T. Cole, and N. Razoronov, "An active tool holder and robust LPV control design for practical vibration suppression in internal turning," Control Engineering Practice, vol. 156, p. 106215, 2025.

Pipe Climbing Robot TAOYAKA VII

Simplified Control of Grasping Force Using a Current Sensor

Saya Eguchi, Kazuyuki Ito
Dept. of Electrical and Electronic Engineering
Hosei University
Tokyo, Japan
email: ito@hosei.ac.jp

***Abstract*— This paper presents a bioinspired pipe-climbing robot that can grasp various columnar objects with unknown shapes by utilizing a passive mechanism. Through this proposed passive mechanism, the arm replicates octopus-like behavior, allowing it to conform to unknown objects without sensing their shape. In this paper, we improved our previous robot and installed current sensors to measure the motor current and estimate the grasping force. Using the current information, the robot controls the grasping force and climbs various columnar objects with different shapes. To demonstrate the effectiveness of the proposed robot, an improved version was developed, and experiments were conducted. As a result, we confirmed that the proposed system works well and that the robot could grasp various unknown-shaped objects using a preset grasping motor current.**

Keywords-climbing robot; flexible mechanism; passive mechanism; bio-inspired robot; octopus-like robot.

I. INTRODUCTION

Autonomous systems operating in unknown, complex environments—such as autonomous vehicles, domestic robots, agricultural robots, construction robots, and rescue robots—have attracted great attention, and various research efforts are being conducted in this area. In general, controlling these robots in complex environments is very challenging because, in unknown environments, robots need various sensors to gather extensive information about their surroundings, and significant computational resources are required to process this information. Additionally, controlling the robots' many degrees of freedom also demands considerable computational power.

On the other hand, even lower organisms in nature can exhibit intelligent behavior in natural environments. In particular, some of these organisms do not possess enough processing ability to control the many degrees of freedom of their bodies in complex natural surroundings. Nevertheless, their behavior remains adaptive.

The mechanism by which they achieve adaptive behavior is not yet fully understood. However, it is thought that the interaction between the body and environment plays an important role, and that flexible or passive mechanisms are crucial for making use of this interaction. Based on this concept, various soft robots have been developed [1]-[14].

In our previous work, we developed various bio-inspired robots with flexible or passive mechanisms, among which the pipe-climbing robot TAOYAKA [1]-[3] is one example. TAOYAKA's arms have multiple passive joints, and by simply pulling a string with a motor, octopus-like movements are achieved, allowing the arm to hold pipes of unknown shapes. In the earlier version of this robot [1], motion control was successfully realized through the interaction between the robot and the pipe. However, controlling the grasping force remained a challenge.

In summary, the biggest challenge in our projects is how to control the many degrees of freedom of a robot in a complex environment. The solution lies in utilizing the interaction between the body and the environment by mimicking the strategies of real creatures.

In particular, this research focuses on grasping force. We improved the previous robot by adding a function to control the grasping force using a simplified method that employs motor current sensors.

The remainder of this paper is organized as follows. Section II introduces the strategy of the octopus. Section III reviews related work and previous research. Section IV proposes an improved robot design. In Section V, we present an experiment to demonstrate the effectiveness of the proposed approach. Finally, Section VI concludes the paper.

II. RERATED WORK AND PREVIOUS WORK

We introduce the behavior of octopuses that inspired us to develop our robot, and we explain earlier and related robots.

A. *Strategy of octopus*

The octopus can grasp various objects of unknown shapes without seeing them. To achieve this intelligent behavior, the octopus uses a strategy in which it gradually contacts its flexible arm with the unknown object from the root to the tip, as shown in Figure 1. Through this strategy, the shape of the arm is determined by the interaction with the object, allowing it to cover the object without sensing its shape.

Our robot, TAOYAKA, is designed to replicate this strategy and can climb pipes or pillars of unknown shapes.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

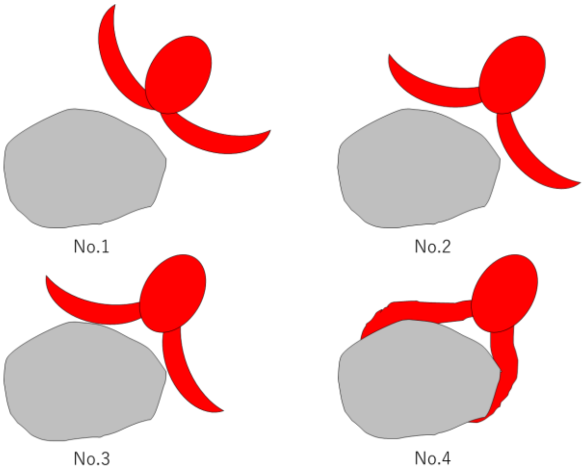


Figure 1. Grasping strategy of octopus

B. Soft robots

Recently, various soft or flexible robots have been developed because they can adapt to complex environments through interaction [4]-[8]. Many conventional soft robots are driven by air, and a pushing force is used to move the body [4]-[8]. However, high pressure reduces softness, and adaptability decreases when generating large forces.

In our previous study [9], we took this disadvantage into account and proposed a novel mechanism driven by the pulling force of a string. In another study [10], we confirmed the advantages of the pulling mechanism through experiments. Using this mechanism, we developed an octopus-like manipulator, as shown in Figure 2 [9][10]. This manipulator can easily pick up objects of unknown shapes by replicating the octopus's strategy. In [10], we proposed utilizing the dynamics of a soft body to reproduce octopus-like behavior and demonstrated its effectiveness through experiments. Subsequently, we applied this grasping mechanism to a pipe-climbing robot, TAOYAKA (Figure 3) [1]-[3]. This robot can grasp pipes or pillars of unknown shapes and climb them. TAOYAKA has two versions based on size: in [1][2], we developed larger versions made with an aluminum frame for structural support and passive rubber joints for flexible movement. In [3], we developed a smaller version using silicone rubber, which can climb real tree branches thanks to its flexibility (Figure 4).

These robots were able to achieve adaptive behaviors despite their simple mechanisms. This provides a significant advantage over conventional complex control mechanisms. However, the grasping force was not adjustable and remained constant. As a result, it was necessary to set an appropriate grasping force in advance, and autonomous control of the grasping hand was anticipated.

In this paper, we introduce a simple controller for regulating the grasping force, utilizing a current sensor to enhance the adaptability of TAOYAKA VI.

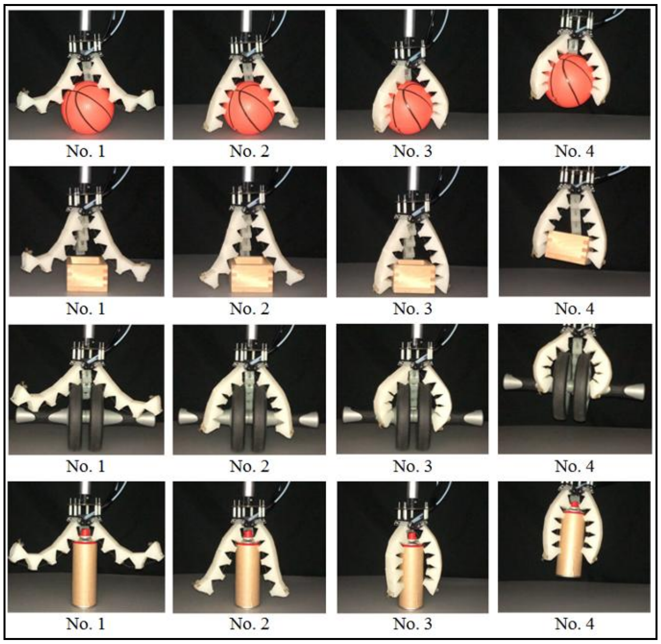


Figure 2. Octopus-like manipulator

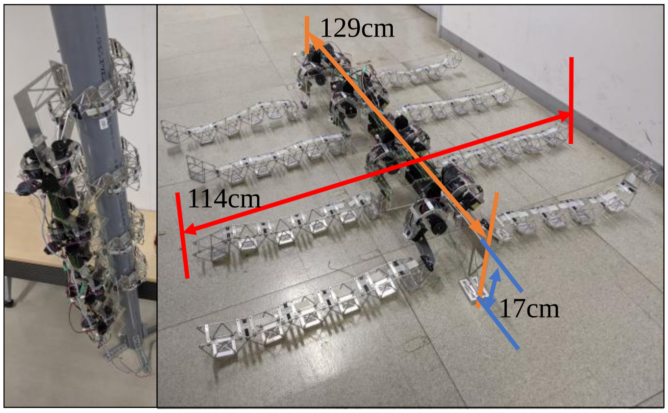


Figure 3. TAOYAKA V

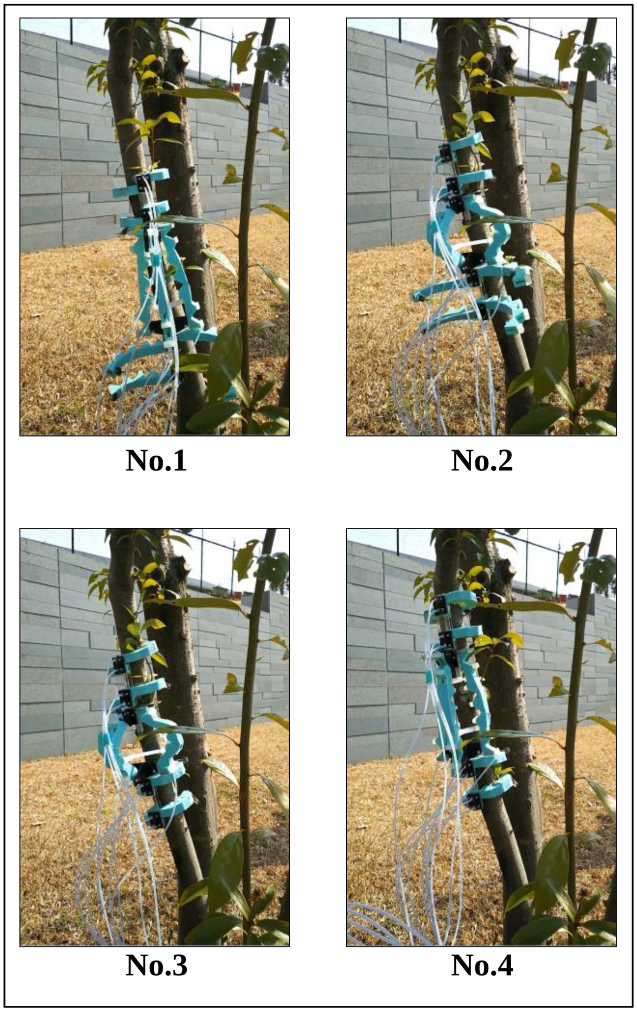


Figure 4. TAOYAKA-SII

III. TAOYAKA VII

We develop TAOYAKA VII by improving previous robot. Figure 5 shows the robot that we are developing, and Table 1 shows its specification.

TABLE I. SPECIFICATION OF THE TAOYAKA VII

Weight	3.0 kg
Length	105 cm
Height	149 cm
Servomotor	KRS-2572HV ICS
Worm Gear	TAMIYA Woem Gear Box HE
Microcomputer	Arduino Mega 2560
Power supply	DC power supply 12V25A

To reduce the weight, the size of the robot is slightly smaller than the previous one. The passive mechanism and moving motion are the same as the previous one, but in this version, we install grasping force control. Details are as follows.

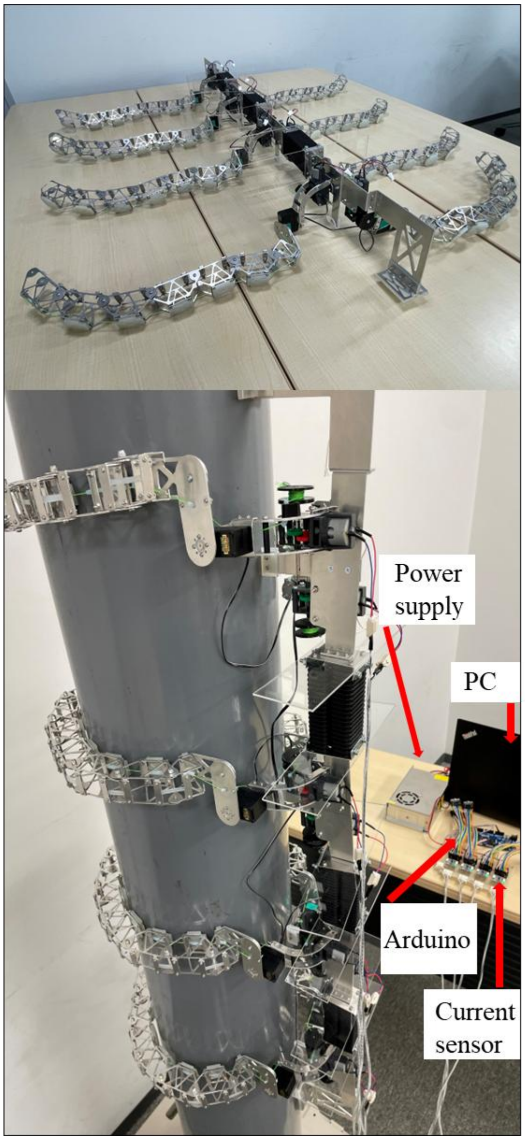


Figure 5. TAOYAKA VII

A. *Mechanisn of the arm*

Figure 6 shows the proposed arm mechanism. The arm is composed of seven passive joints and is driven by one motor with worm gear. A string is installed through the arm, and by pulling the string, the arm is closed. Rubber bands are attached between links, and the rubber strength gradually increases from the root to the tip. Owing to the difference of rubber tension, the arm moves from the root to the tip by just pulling the string, and the octopus-like movement is replicated. Figure 7 shows the octopus-like movement.

At TAOYAKA VII, we install current sensors on the motor that pulls the string. By monitoring these sensors, we can detect excessive torque and cut off the current. Since the motor has a worm gear, even if the electrical power is turned off, the arm remains locked, and the robot can keep holding.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

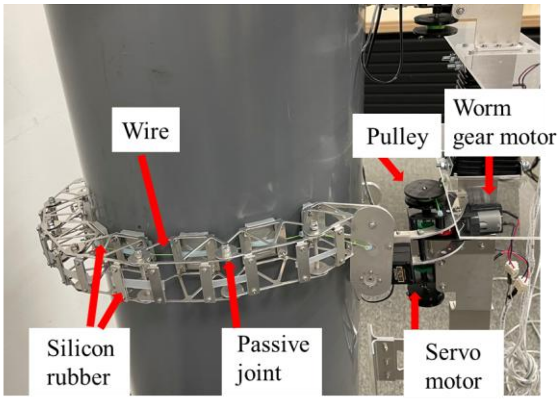
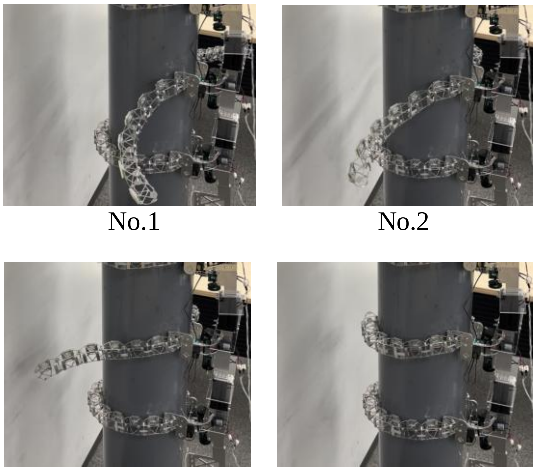


Figure 6. Arm mechanism


No.1 No.2

No.3 No.4

Figure 7. Octopus-like movement

B. Control method and motion pattern for Climbing

The robot consists of four segments, and each segment has two arms, as shown in Figure 5. Each arm is connected to the trunk via a servomotor. When the servomotor rotates, the trunk rises, as illustrated in Figure 8. Figure 9 shows the flowchart for controlling the robot, and Figure 10 shows the climbing pattern.

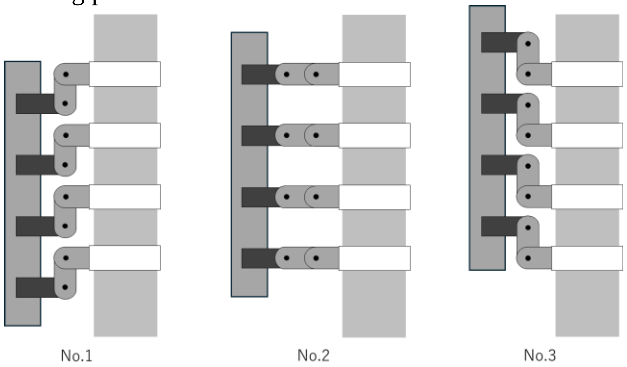


Figure 8. Rising motion

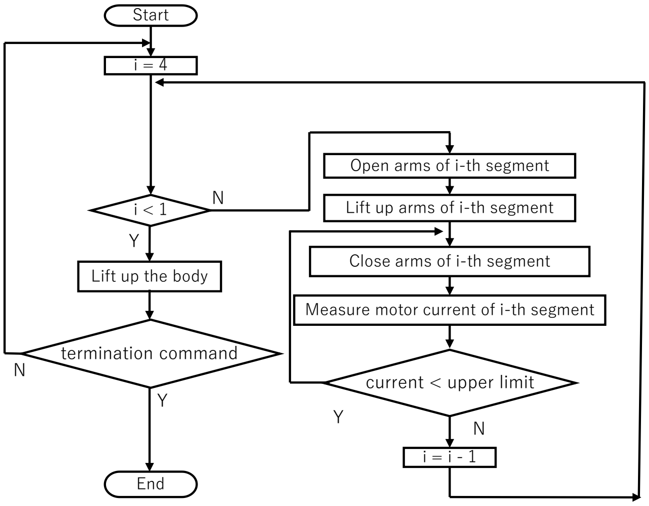

Start
i = 4
i < 1
N
Y
Open arms of i-th segment
Lift up arms of i-th segment
Close arms of i-th segment
Measure motor current of i-th segment
Lift up the body
termination command
N
Y
current < upper limit
Y
N
i = i - 1
End

Figure 9. Flowchart for controlling the robot

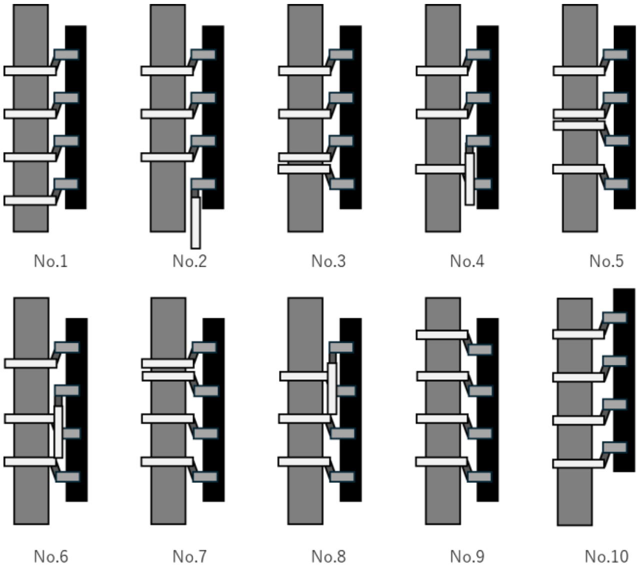


Figure 10. Climbing Pattern

By repeating the pattern of Figure 10, the robot ascends the pipe.

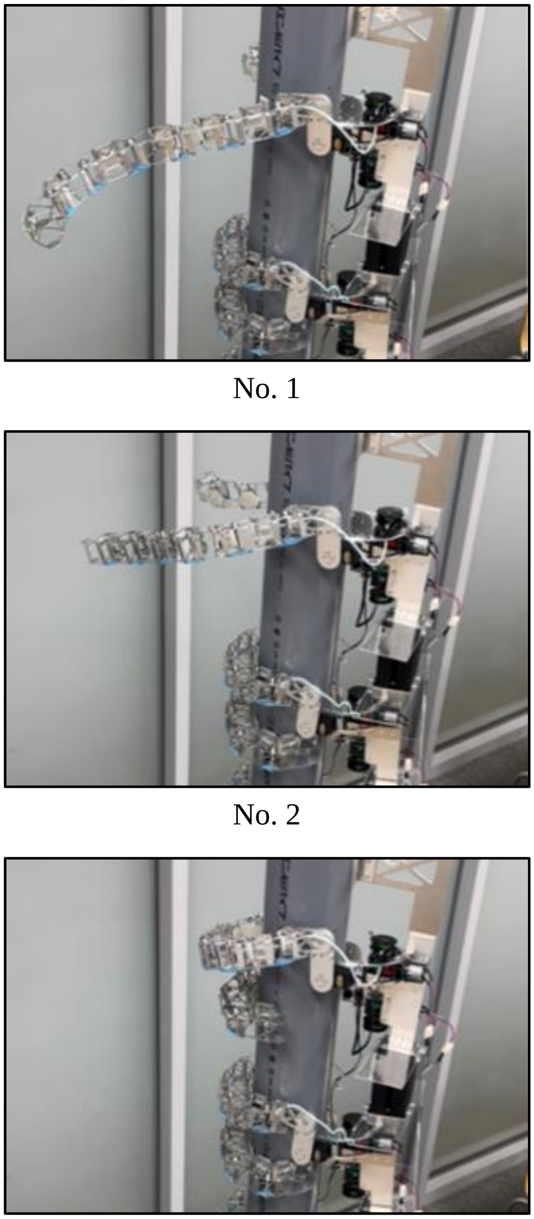
IV. EXPERIMENT

We conducted experiments to validate the proposed control scheme for grasping force. To control the grasping force, we used the motor current. We applied the same control algorithm to three pipes with different diameters. Figure 11 presents a typical behavior for one arm, and Figure 12 shows the motor current.

As shown in Figure 12, in the no-control case, the current was saturated at a high level, indicating that the grasping force was too high to grasp soft objects. In contrast, in the case of the proposed control, the current was cut off around the desired value.

In each experiment, the maximum current exceeded 1500 mA even though the desired value was set to 1500 mA. This is because the sampling time in this experiment was set to 0.2 seconds, meaning that the current value was measured every 0.2 seconds. The motor power was cut off only when the current first exceeded 1500 mA. Therefore, if more precise control is required, this can be addressed by shortening the sampling time.

Since the proposed mechanism employs a worm gear, as shown in Figure 6, the arm can maintain its grasp even when the motor current is cut off. Thus, the maximum grasping force can be easily controlled by the simple control scheme shown in Figure 9.



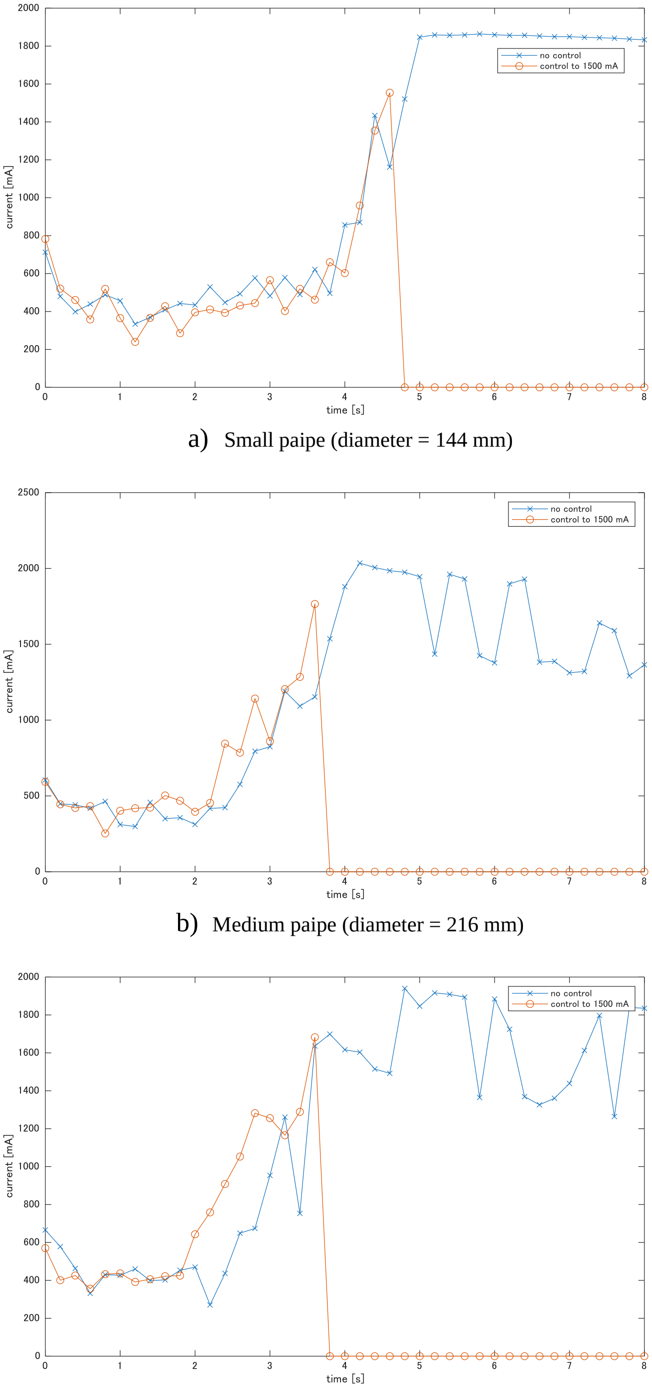
No. 1

No. 2

No. 3

Figure 11. Realized movement

Our next step is to apply the control scheme to all the arms and conduct climbing experiments on various soft columnar objects.



a) Small paipe (diameter = 144 mm)

b) Medium paipe (diameter = 216 mm)

c) Large pipe (diameter = 257 mm)

Figure 12. Experimental result of the current control

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

V. Conclusion

In this paper, we focus on the adaptive mechanisms of animals and apply a passive and flexible mechanism to a pipe-climbing robot. We improved the previous robot by adding a function to control the grasping force using a simplified method that employs motor current sensors.

With the proposed mechanism, the arm replicates octopus-like behavior and can grip the pipe using a preset grasping motor current. Our future work is to apply this robot to columnar objects with unknown dents and bumps to demonstrate the effectiveness of the proposed mechanism.

Acknowledgment

This research was partially supported by the Japan Society for the Promotion of Science through the Grant-in-Aid for Scientific Research [22K12155]

References

[1] H. Egawa and K. Ito, “A Multi-Legged Robot Capable Of Moving From Pipe To Pipe: TAOYAKA-VI,” Proc. of International Symposium on Socially and Technically Symbiotic Systems(STSS), CDROM, 2021.

[2] K. Ito and Y. Ninomiya, “TAOYAKA V: a multi-legged robot, successfully combining walking and climbing mechanisms,” Artif Life Robotics, Vol. 26, pp. 97–102, 2021.

[3] S. Takaku, K. Mamiya, and K. Ito, “TAOYAKA-S III : Soft multi-legged robot for climbing unknown columnar objects : Realization of self-contained mechanism,” International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 236-241, 2021.

[4] F. Chen and M. Y. Wang, “Design Optimization of Soft Robots: A Review of the State of the Art,” in IEEE Robotics & Automation Magazine, vol. 27, No. 4, pp. 27-43, 2020.

[5] F. Stroppa, F. J. Majeed, J. Batiya, E. Baran, and M. Sarac, “Optimizing soft robot design and tracking with and without evolutionary computation: an intensive survey,” Robotica. Published online, pp.1-27, 2024.

[6] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, “Control Strategies for Soft Robotic Manipulators: A Survey,” Soft Robotics Vol.5, No. 2, pp. 149-163, 2018.

[7] L. Schiller, A. Seibel, and J. Schlattmann, “Toward a gecko-inspired, climbing soft robot,” Front Neurorobot Vol. 13, pp. 106, 2019.

[8] G. Phanomchoeng, P. Pitchayawetwongsa, N. Boonchumanee, S. Lin, and R. Chancharoen, “Grasping profile control of a soft pneumatic robotic gripper for delicate gripping,” Robotics, Vol. 12, No. 4, pp. 107, 2023.

[9] K. Ito and S. Hagimori, “Flexible manipulator inspired by octopus: development of soft arms using sponge and experiment for grasping various objects,” Artif Life Robotics Vol. 22, pp. 283–288, 2017.

[10] K. Ito and T. Mukai, “Flexible manipulator inspired by octopi: advantages of the pulling mechanism,” Artificial Life and Robotics, Vol. 25, pp. 167–172, 2020.

[11] Y. Hiramaki and K. Ito, "URARAKAVII: Multi-Legged wall climbing robot, Improving of the mobility to travel from a horizontal floor to a vertical wall,” IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1-6, 2023.

[12] S. Yamashita and K. Ito, “Six-legged crawling soft robot: NOBIYAKA,” Artifcial Life and Robotics, Vol. 28, pp. 530-539, 2023.

[13] K. Ito, Y. Homma, H. Shimizu, and Y. Sakuhara, “Introduction to Octopus-Inspired Soft Robots: Pipe-Climbing Robot TAOYAKA-S II and Ladder-Climbing Robot MAMEYAKA,” Advances in Intelligent Systems Research and Innovation, Studies in Systems, Decision and Control, Vol. 379, pp. 287-313, 2021.

[14] Y. Aso, K. Aihara, and K. Ito, “Multi-legged robot for rough terrain: SHINAYAKA-L VI,” International Conference on Advanced Mechatronic Systems (ICAMechS), pp. 136-141, 2019.

Next-Generation Automotive Radar: Pioneering Safety and Driving Autonomy

Sanaz Asgarifar
Bosch Car Multimedia
Braga, Portugal
email: Sanaz.asgarifar@pt.bosch.com

Tiago Afonso
Bosch Car Multimedia
Braga, Portugal
email: Tiago.afonso@pt.bosch.com

***Abstract*— Automotive radar technology has rapidly advanced, providing increasingly sophisticated functionalities that are essential to achieving the vision of fully autonomous vehicles. Modern radar sensors are equipped with enhanced detection ranges that now reach up to 300–400 meters, improved angle separability within 0.1°–1°, and Artificial Intelligence (AI)-driven object recognition capabilities, making them indispensable for automated vehicle systems. These improvements enable radar systems to detect and differentiate multiple objects, including vulnerable road users, even in complex urban settings and at high speeds. Automotive radar operates reliably in various environmental conditions, including rain, fog, and low light, thus providing stability and accuracy critical to safe vehicle autonomy. This paper reviews the latest innovations in automotive radar and explores its essential role in the automotive industry's journey toward achieving fully autonomous vehicles. By examining cutting-edge radar sensor features and applications, this study demonstrates how automotive radar technology effectively addresses perception and safety challenges, ensuring robust and consistent performance even under adverse conditions.**

Keywords - Automotive radar; Autonomous driving; Perception systems; Enhanced detection range; Waveguide antenna.

I. INTRODUCTION

Radar systems have long been integrated into vehicles for applications like Adaptive Cruise Control (ACC) and collision avoidance. However, traditional radar technology has limitations in resolution, interference, and robustness, impacting performance in urban environments. With the rise of autonomous driving, demand has grown for high-precision radar systems [1].

Modern automotive radar plays a key role in driver assistance and automation, excelling in detecting objects, measuring distances, and tracking velocity in real time. These capabilities ensure reliable operation in challenging conditions such as poor visibility and dense traffic. Recent advancements, including extended range, higher resolution, and improved object classification, further support autonomous vehicle needs [2].

Despite these improvements, challenges remain, such as interference from other sensors, difficulty in distinguishing stationary objects, and resolution limitations in complex environments. Integrating radar with Light Detection and Ranging (LiDAR) and camera systems introduces challenges related to cost, computational power, and sensor fusion accuracy [3].

This paper is structured as follows: Section II discusses radar's role in vehicle autonomy. Section III covers recent radar advancements, focusing on signal processing and sensor integration. Section IV examines radar's impact on safety and industry regulations. Section V presents performance insights from the latest radar models. Section VI concludes with key findings and future research directions.

II. ROLE OF RADAR TECHNOLOGY IN ACHIEVING FULL AUTONOMY

Fully autonomous vehicles rely on precise and reliable sensing in various environments. Radar sensors enable key automation functions such as ACC, Automatic Emergency Braking (AEB), and lane-keeping assistance [4]. Their capability to detect range, speed, and object positions enhances perception system redundancy, essential for safe operation.

Modern radar can track multiple objects simultaneously, crucial for navigating intersections and high-traffic areas. It contributes to high-definition mapping and real-time updates via cloud-connected databases, improving situational awareness and hazard response [5]. Figure 1 illustrates improvements in sensitivity and accuracy in handling complex scenarios.

III. ADVANCED AUTOMOTIVE RADAR TECHNOLOGY

Recent advancements in automotive radar technology have led to significant improvements in signal processing, sensor integration, and radar architecture, which contribute to better detection and resolution. Contemporary radar sensors, such as 4D imaging radar, operate in the 76-81 GHz frequency band, providing detection ranges up to 391 meters. This high-frequency range enhances range sensitivity, angle separability, and speed detection, which are essential for accurately identifying vehicles, pedestrians, and other obstacles in complex traffic environments.

A significant innovation in automotive radar technology is the use of waveguide antennas, specifically designed to improve radar performance in terms of directivity and efficiency. Waveguide antennas are highly efficient in transmitting and receiving high-frequency signals, thus enabling better accuracy and range in radar applications. They are also less susceptible to interference, which improves signal clarity and reduces noise. When used in radar systems, waveguide antennas facilitate sharper beamforming, allowing for precise detection of objects even

ISBN: 978-1-68558-241-8

in densely packed traffic scenarios. With these characteristics, waveguide antennas play an essential role in enhancing radar performance and contribute to more reliable detection and tracking in autonomous driving [4].

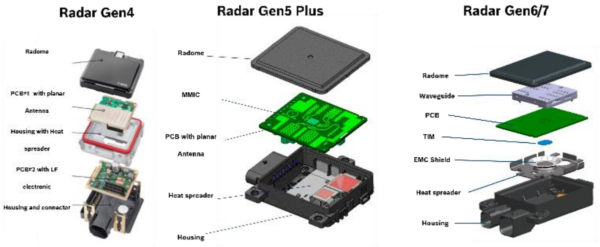


Figure.1 Radar sensor hardware architecture.

IV. POTENTIAL AND IMPACT OF ADVANCED RADAR ON AUTOMOTIVE INDUSTRY

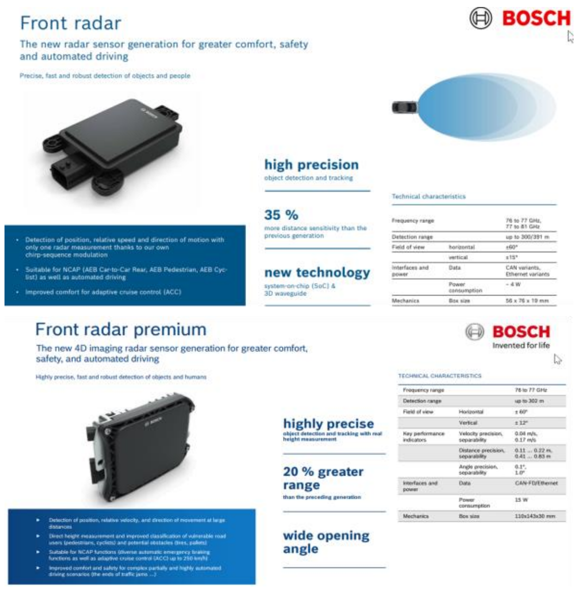
As autonomous driving technology matures, the role of radar in perception systems becomes increasingly indispensable. Radar's resilience to challenging weather conditions, along with its high accuracy, ensures that autonomous systems can perform reliably in various adverse scenarios, such as rain, fog, or low-light environments. Additionally, AI-enhanced object recognition in radar technology allows sensors to continuously adapt and improve, making radar a scalable solution for different levels of vehicle automation [5].

The implementation of advanced radar systems in fully autonomous vehicles has the potential to improve road safety by significantly reducing human error, which accounts for most traffic accidents. As radar technologies progress, they will be pivotal in meeting safety and performance standards set by regulatory organizations like the New Car Assessment Program (NCAP). The integration of automotive radar not only aligns with regulatory demands but also serves as a cost-effective, reliable, and scalable solution, aligning with the industry's vision for future mobility.

V. RESULTS AND DISCUSSIONS

The latest advancements in automotive radar technology exhibit remarkable improvements in detection accuracy, range, and operational resilience. Bosch's front radar model offers a 35% increase in range sensitivity compared to its predecessor, improving object detection, tracking, and angle separability for ACC and AEB applications. Meanwhile, the premium 4D imaging radar delivers a 20% increase in detection range and enhanced object identification, with angular precision of approximately 0.1° (Figure 2).

Future work will focus on integrating advanced AI algorithms to enhance real-time decision-making and adapting radar systems for higher automation levels. Additionally, optimization for power efficiency and size reduction remains a priority, particularly for electric and compact vehicles. Continued innovation in waveguide antennas could further improve directional accuracy, reduce interference, and enable multi-beam capabilities, enhancing robustness in complex traffic conditions. Increased data integration across radar, LiDAR, and camera systems may further refine autonomous navigation, advancing the industry toward fully autonomous vehicles.



Front radar

The new radar sensor generation for greater comfort, safety and automated driving

Precise, fast and robust detection of objects and people

- Detection of position, relative speed and direction of motion with only one radar measurement thanks to our own chirp-sequence modulation
- Suitable for NCAP (AEB Car-to-Car Rear, AEB Pedestrian, AEB Cyclist) as well as automated driving
- Improved comfort for adaptive cruise control (ACC)

high precision
object detection and tracking

35 %
more distance sensitivity than the previous generation

new technology
system-on-chip (SoC) & 3D waveguide

Technical characteristics

Frequency range		76 to 77 GHz, 77 to 81 GHz
Detection range		up to 300/391 m
Field of view	horizontal	±60°
	vertical	±15°
Interfaces and power	Data	CAN variants, Ethernet variants
	Power consumption	~ 4 W
Mechanics	Box size	56 x 76 x 19 mm

Front radar premium

The new 4D imaging radar sensor generation for greater comfort, safety, and automated driving

Highly precise, fast and robust detection of objects and humans

- Detection of position, relative velocity, and direction of movement at large distances
- Direct height measurement and improved classification of vulnerable road users (pedestrians, cyclists) and potential obstacles (tires, pallets)
- Suitable for NCAP functions (diverse automatic emergency braking functions as well as adaptive cruise control (ACC) up to 250 km/h)
- Improved comfort and safety for complex partially and highly automated driving scenarios (the ends of traffic jams ...)

highly precise
object detection and tracking with real height measurement

20 % greater range
than the preceding generation

wide opening angle

TECHNICAL CHARACTERISTICS

Frequency range		76 to 77 GHz
Detection range		up to 302 m
Field of view	Horizontal	± 60°
	Vertical	± 12°
Key performance indicators	Velocity precision, separability	0.04 m/s, 0.17 m/s
	Distance precision, separability	0.11 ... 0.22 m, 0.41 ... 0.83 m
	Angle precision, separability	0.1°, 1.0°
Interfaces and power	Data	CAN-FD/Ethernet
	Power consumption	15 W
Mechanics	Box size	110x143x30 mm

Figure. 2 Results from latest generation of Bosch Radar.

VI. CONCLUSION

Automotive radar technology has achieved impressive advancements, bringing the automotive industry closer to the vision of fully autonomous vehicles. Enhanced radar sensors enable vehicles to navigate complex environments, ensure safety, and maintain high operational efficiency. The critical role radar plays in object detection, target tracking, and environmental perception makes it an essential component of autonomous vehicle design. With ongoing improvements in radar architecture, such as waveguide antenna integration, and AI-driven decision-making, radar technology will be instrumental in shaping the future of autonomous driving. As radar systems continue to evolve, their integration into vehicle automation will be a defining factor in the automotive industry's journey toward fully autonomous mobility.

REFERENCES

[1] J. Levinson et al., *Towards fully autonomous driving: Systems and algorithms*, IEEE Intelligent Vehicles Symposium, 2011

[2] R. Lin and M. Dehring, "4D Imaging Radar: Transforming Automotive Perception," IEEE Transactions on Vehicular Technology, vol. 69, no. 12, [15145]-[15157], Dec. 2021. doi:10.1109/TVT.2021.3115229.

[3] J. H. Lee and D. H. Kim, "Sub-Terahertz Radar Technology for Enhanced ADAS Capabilities," IEEE Access, vol. 9, [80492]-[80506], 2021. doi:10.1109/ACCESS.2021.3092251.

[4] S. Schertler et al., "Advances in RFCMOS Radar Technology for Autonomous Driving," Microwave Journal, vol. 65, no. 11, 2022.

[5] V. Patel, "Software-Defined Radar for Enhanced Safety in Vehicles," FutureBridge Automotive Technology Insight, Aug. 2023.

Applying Q-Learning Agents to Distributed Graph Problems

Jeffrey McCrea and Munehiro Fukuda
Division of Computing and Software Systems
University of Washington Bothell
Bothell, WA 98011, U.S.A.
e-mail: {jeffmccr | mfukuda}@uw.edu

***Abstract*—Breadth-first search is used as a brute-force approach to parallelizing graph computations over a distributed graph structure, such as the shortest path, closeness centrality, and betweenness centrality search. As a smart alternative, we integrate Q-learning capabilities into agents, dispatch them over a distributed graph, have them populate the Q-table, and accelerate their graph computations. We developed the Q-learning agents with the Multi-Agent Spatial Simulation (MASS) library and measured their parallel performance when running over a distributed graph with 16K or more vertices. This paper identifies the graph scalability, static/dynamic graph structures, application types, and Q-learning hyperparameters that take advantage of Q-learning agents for parallel graph computing.**

***Keywords*-Q learning; agent-based modeling; cluster computing; graph algorithms.**

I. INTRODUCTION

Graphs play a critical role in data representation across diverse domains, such as social networks, transportation systems, and biological networks. The exponential growth of graph sizes and complexity necessitates scalable solutions to handle vast datasets. Distributed frameworks, such as Google's Pregel [1] and Apache Spark GraphX [2], address this need by partitioning graph data across multiple computing nodes for parallel processing. While these solutions provide robust support for large-scale static graphs, their predefined computation models fall short when adapting to dynamic graph environments.

In contrast, reinforcement learning, particularly Q-learning, introduces a paradigm shift by enabling agents to learn graph structures and adapt their strategies dynamically. Q-learning's flexibility demonstrates a promising approach to handling static and dynamic graphs, especially in scenarios with frequent and unpredictable topology changes. To evaluate the Q-learning agents' adaptability, flexibility, and parallel performance, we implemented them within the Multi-Agent Spatial Simulation (MASS) Java library [3].

Our work applied MASS-enabled Q-learning agents to three fundamental graph problems: shortest path, closeness centrality, and betweenness centrality. Experimental results demonstrated the efficiency of this approach, including performance gains of 66% to 99% and a 190% reduction in training time when executed on eight computing nodes. These improvements highlight the potential of combining distributed Q-learning agents to effectively tackle graph analysis.

The remainder of this paper is structured as follows. Section II reviews related work on parallel graph computing, reinforcement learning, and dynamic graph processing. Section III details the design and implementation of Q-learning agents within the MASS framework. Section IV presents the experimental setup and results, comparing Q-learning-based solutions with existing MASS implementations. Finally, Section V concludes our work on distributed graph computing using Q-learning agents.

II. RELATED WORK

This section reviews some libraries and algorithms related to our MASS-parallelized Q-learning agents from the following three viewpoints: (1) parallel graph computing, (2) Q-learning graph computing, and (3) tolerance to dynamic graphs.

A. Parallel graph computing

The exponential growth of graph sizes has necessitated distributed platforms, such as Google's Pregel [1] and Spark GraphX [2], to handle large-scale graph computations. In Pregel, the master computing node partitions a graph dataset across worker computing nodes for parallel computation. Pregel simulates message propagation or breadth-first search over the graph in iterative supersteps, where each vertex exchanges messages with its neighbors. GraphX utilizes a similar vertex-centric message propagation approach, facilitated by the Pregel API and enhanced through Spark's Resilient Distributed Datasets, enabling efficient in-memory computation and fault tolerance.

While GraphX and Pregel are robust for static graphs, they struggle with dynamic environments where changes in the graph necessitate complete recomputation. Depending on the size of the graph, this can result in significant computational costs as the frameworks have no native mechanism to respond incrementally to these changes. In this way, adaptive solutions have a significant advantage over purely static implementations.

B. Q-Learning Graph Computing

Q-learning [4] has seen wide adoption in reinforcement learning due to its ability to learn and adapt to unknown environments and derive optimal policies for navigating them, particularly in the shortest path problem. In this context, graph vertices represent states, edges represent possible actions, and rewards are based on the edge weights. As the Q-learning agent navigates the graph, it iteratively populates the Q-table, capturing the optimal policies for efficient traversal.

ISBN: 978-1-68558-241-8

In literature, Q-learning has successfully addressed the shortest path problem. Sun [5] utilized Q-learning to navigate a static city graph, demonstrating its effectiveness in road network-like graph structures. Wang et al. [6], applied Q-learning to autonomous robots in a grid-like environment, prioritizing the shortest path and obstacle avoidance. The study highlighted Q-learning's ability to enable intelligent decision-making in uncertain environments driven by learned agent intelligence. Nannapaneni [7] introduced a novel path-routing algorithm based on a modified Q-learning algorithm for dynamic network packet routing, underscoring its adaptability to evolving environments. However, these studies often focus on small graphs with fixed hyperparameters on single-machine setups.

Two extensions of the shortest path search are closeness and betweenness centrality computation [8]. The former computes the shortest path from each vertex *u* to all other vertices *v*, sums all the path lengths, and takes its reciprocal with $C(u) = \frac{N-1}{sum(u)}$. The latter determines all shortest paths between each pair of vertices *u* and *v* in a graph, counts how many paths pass through a given vertex, and computes the centrality with $B(s) = \sum_{u \neq s \neq v} \frac{\sigma_{uv}(s)}{\sigma_{uv}}$, where σ_{uv} is the total number of the shortest paths from vertex *u* to vertex *v*, and $\sigma_{uv}(s)$ is the number of those paths passing through s. For dynamic graphs, recalculating these metrics with each change can be computationally expensive. Parallel approaches, such as using landmarking techniques [9] or GPU acceleration [10], have been proposed to address this complexity. However, these methods often require extensive preprocessing or specialized hardware, limiting their practicality.

C. Dynamic graphs

Dynamic graphs, where changes to the graph's vertices or edges occur over time, better emulate real-world situations like those in traffic networks. Changes to the graph can be incremental, adding vertices and edges or increasing weights, or decremental, removing vertices and edges or decreasing weights [11]. Traditional algorithms, such as Dijkstra's or Bellman-Ford, are designed for static graphs as they require complete recomputation for each change. This recomputation becomes prohibitively expensive when handling frequent changes.

To address the challenges associated with dynamic graphs, several approaches have been explored. Early efforts adapted traditional algorithms like A* [12] for incremental changes. Techniques like those employed in Dynamic Dijkstra's [13] and D* [14] utilized incremental search strategies that recompute only affected portions of the graph. Subsequent methods used landmark-based techniques [15] for real-time shortest path approximation and distributed techniques [16] to partition the graph data and subgraph computation across a distributed cluster of computing nodes. While these approaches improve performance, they often rely on knowledge of where changes occur, which is impractical for unpredictable, real-world environments. Additionally, landmarking requires extensive preprocessing to select landmark vertices accurately. Our work focuses on measuring dynamic performance without prior knowledge of graph changes or the underlying structure of the graph.

D. Challenges and Agent-based Solutions

Our project seeks to expand on these studies by exploring a diverse range of graph sizes, assessing the performance of Q-learning on dynamic graphs using the MASS framework, and improving the algorithm's efficiency through performance-based hyperparameter tuning and MASS-enabled agent features. For this purpose, we take the following four strategies:

1) MASS constructs and maintains a distributed graph over a cluster system for repetitive graph computing [17].
2) MASS agents build a Q-table over a distributed graph and dynamically tune hyperparameters during training.
3) MASS facilitates dynamic graphs by adding or deleting vertices and their edges.
4) MASS agents adapt to graph changes by reaching semi-optimal solutions when the initial graph structure is altered.

III. An Implementation of Q-Learning Agents

This section details the Q-learning algorithm for the shortest path problem and its application to closeness and betweenness centrality. It also covers the implementation details for the Q-learning shortest path, closeness centrality, and betweenness centrality in MASS, including MASS-specific performance improvements.

A. MASS Library

The MASS Java Library [3] uses two primary components: *Places* and *Agents*. *Places* represent individual data members of a larger dataset, whereas *Agents* comprise individual agent execution entities that traverse *Place* objects to perform computation. Individual *Place* objects are allocated across a specific number of computing nodes within a cluster and can be referenced using a universally recognized set of coordinates.

After mapping each *Place* within the cluster, they are assigned to threads, allowing them to communicate with other *Place* objects and the agents residing on them. *Agents* are organized into groups on each computing node and can move between nodes using serialization and TCP communication.

The MASS library has been extended to support explicit graph structures in a multi-node cluster. *GraphPlaces*, an extension of the original *Places* class, consists of *VertexPlace* objects that store graph vertex information. Graphs can be manually created by invoking the `addVertex()` and `addEdge()` methods, or users can load graph data from a supported graph format, including Hippe, Matsim, or a few propriety formats [17]. MASS enables parallel file processing, making graph creation far more efficient than serial methods. Keeping with its tradition of distributed cluster computing, vertices are balanced across the entire cluster in a round-robin fashion to ensure an even distribution of graph data between all the cluster nodes.

B. Algorithm Design

The Q-learning shortest path algorithm models the environment as a graph $G = (V, E)$, where vertices V are states and edges E are actions connecting them. An agent learns the optimal path by iteratively updating the Q-table $Q(s, a)$, which stores the value of taking action a in state s. The Q-value update rule is given by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a} Q(s',a) - Q(s,a) \right]$$

Here, α is the learning rate, γ is the discount factor, and r is the reward received for the state transition. Rewards guide the agent by assigning penalties for non-terminal states as the edge cost, a positive reward for reaching the destination, and a penalty for encountering dead ends. The parameter ϵ controls the exploration-exploitation tradeoff in an epsilon-greedy action selection process (see Figures 1 and 2), where agents explore new paths with probability ϵ and exploit the highest Q-value action otherwise. Training continues until the Q-values converge or the maximum number of episodes is met.

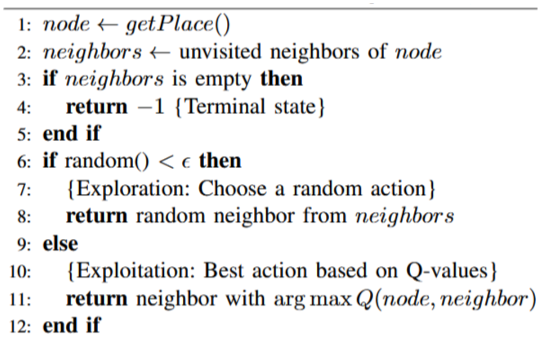
To improve on traditional Q-learning, dynamic hyperparameter tuning is utilized to enhance learning efficiency and adaptiveness. This approach uses a distributed reward window, a linked list that stores cumulative rewards received during training episodes. If the current average reward in the window decreases relative to the previous average, α and ϵ are increased to promote exploration. In contrast, if the reward improves, these parameters are decreased to stabilize learning and encourage exploitation. This dynamic adjustment enables agents to adapt to changing performance, improving convergence and efficiency compared to fixed parameter decay.

Beyond finding the shortest path, the Q-table also enables suboptimal path enumeration. By defining a Q-value and path length threshold, agents can explore alternative paths within a certain percentage of the optimal Q-value and path length. This flexibility allows the agent to identify viable alternative routes in the event of changes in the graph structure, such as blocked paths or updated edge weights. The Q-table's comprehensive knowledge provides a robust basis for exploring these alternative paths without requiring additional training.

The Q-learning shortest path algorithm naturally extends to calculating graph centrality metrics like closeness and betweenness. In MASS, this is achieved by assigning a Q-learning agent to each source-destination pair, with Q-tables organized in a two-dimensional array of hashmaps at each vertex. For betweenness centrality, agents are further tasked with enumerating all optimal paths between vertex pairs.

C. Algorithm Implementation with MASS

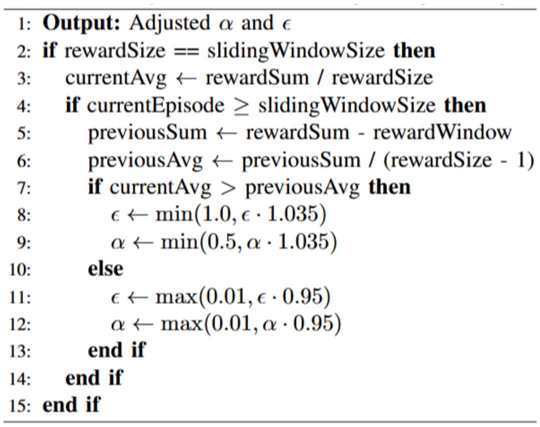
The MASS library streamlines the application of the Q-learning shortest path algorithm by distributing computation and facilitating multi-agent collaboration. The algorithm is built upon the fundamental abstractions of MASS: *Places* and *Agents*, utilizing the graph-specific extension of *Places*, *GraphPlaces*, to enable easy-to-use MASS graph functionality. Furthermore, the *Node class*, representing individual graph nodes and an extension of *VertexPlace*, is created to incorporate Q-learning functionality related to distributed Q-table storage and dynamic hyperparameter tuning.



```
node ← getPlace()
neighbors ← unvisited neighbors of node
if neighbors is empty then
    return −1 {Terminal state}
end if
if random() < $\epsilon$ then
    {Exploration: Choose a random action}
    return random neighbor from neighbors
else
    {Exploitation: Best action based on Q-values}
    return neighbor with $\arg\max Q(node, neighbor)$
end if
```

Figure 1. Action Selection algorithm in the Q-learning process.

The primary agent, *QLAgent*, navigates the graph, updates Q-values, and dynamically adjusts hyperparameters based on feedback from the reward window. During training, the *QLAgent* iteratively explores the graph, storing the learned values in the distributed Q-tables. The action selection process is handled by the `chooseAction` function, detailed in Figure 1, which implements the epsilon-greedy policy to balance exploring new actions and exploiting the highest-value actions stored in the Q-table. At the end of each episode, the *QLAgent* evaluates its performance and fine-tunes its hyperparameters using the `adjustAlphaEpsilon` function, described in Figure 2. This adaptive function increases exploration and learning rates when performance declines and decreases when performance improves, ensuring efficient learning throughout the training process. The adjustment factors of 1.035 and 0.95 for α and ϵ were identified through manual testing as values that provide balanced yet gradual change without excessive oscillation. Once training is complete, the *QLAgent* uses the learned values from the Q-table to identify the optimal path.



```
Output: Adjusted $\alpha$ and $\epsilon$
if rewardSize == slidingWindowSize then
    currentAvg ← rewardSum / rewardSize
    if currentEpisode ≥ slidingWindowSize then
        previousSum ← rewardSum - rewardWindow
        previousAvg ← previousSum / (rewardSize - 1)
        if currentAvg > previousAvg then
            $\epsilon \leftarrow \min(1.0, \epsilon \cdot 1.035)$
            $\alpha \leftarrow \min(0.5, \alpha \cdot 1.035)$
        else
            $\epsilon \leftarrow \max(0.01, \epsilon \cdot 0.95)$
            $\alpha \leftarrow \max(0.01, \alpha \cdot 0.95)$
        end if
    end if
end if
```

Figure 2. Dynamic α and ϵ tuning algorithm.

Suboptimal paths are explored using the *PathAgent*, a simplified version of the *QLAgent* designed for path enumeration. The *PathAgent* identifies routes within user-defined Q-value and path length thresholds, enabling the discovery of both optimal and viable alternative paths.

MASS significantly enhances Q-learning performance on large-scale graphs by optimizing resource utilization and scalability. The distributed architecture efficiently shares memory and computational load across computing nodes and enables the sparse Q-table representation focused on only relevant state-action pairs. Moreover, MASS's multi-agent training support allows numerous agents to explore the graph simultaneously, accelerating the training process and ensuring comprehensive graph coverage.

IV. Evaluation

This section presents the experimental results for the Q-learning shortest path, closeness, and betweenness centrality applications, comparing their performance against current MASS implementations. Experiments were conducted on the CSSMPI cluster at the University of Washington Bothell, comprising eight virtual machines allocated four cores at 2.10GHz from an Intel Xeon Gold 6130 processor and 16GB of RAM. The MASS Q-learning applications utilized MASS Java Core version 1.4.3, enhanced by Fastutil version 8.5.8 for optimized data structures. Performance metrics focus on training time, adaptability for dynamic graph changes, and efficiency improvements through multi-agent training.

TABLE I. Graph Datasets Properties

Dataset	Vertices	Edges	Avg Deg	Max Deg
Synthetic (Shortest)	500-16K	63K-64M	126-4038	189-5833
Road Networks	1861-19K	4K-50K	2.35-5.3	5-10
Synthetic (Centrality)	8-256	16-16K	2-64	3-97

Table I summarizes the datasets used for evaluation, including synthetic graphs for shortest path testing, real-world road network graphs for dynamic performance analysis, and synthetic centrality graphs for closeness and betweenness centrality testing.

A. Shortest Path

Figure 3 demonstrates the training performance of the Q-learning shortest-path application on synthetic graphs. The application performs best on single-node executions for graphs up to 8000 vertices, as the need for communication between nodes in the cluster is minimized. For the 16,000-vertex graph, multi-node execution becomes necessary, with eight computing nodes achieving optimal performance due to distributed agent coordination and parallel execution.

After training, the Q-learning shortest path algorithm significantly outperforms the current MASS shortest path implementation in output speed, as shown in Figure 4. The trained *QLAgent* efficiently navigates the graph using the trained Q-table to produce the shortest path, demonstrating the inherent advantage of pre-trained Q-learning models.

Dynamic benchmarking evaluates the adaptability of the Q-learning application to graph changes. Figure 5 compares Q-learning performance with the current MASS shortest path application when faced with the removal of a single vertex. The Q-learning implementation excels in well-connected synthetic graphs, quickly adapting to minor changes due to the knowledge stored in the Q-table. More substantial disruptions, such as the removal of five vertices, resulted in increased retraining times, thereby making rerunning the current MASS implementation more efficient for significant changes.

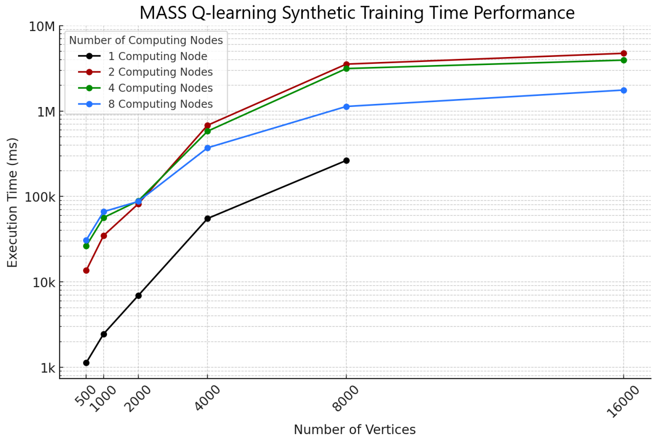


Figure 3. MASS Q-learning Shortest Path Synthetic Training Time

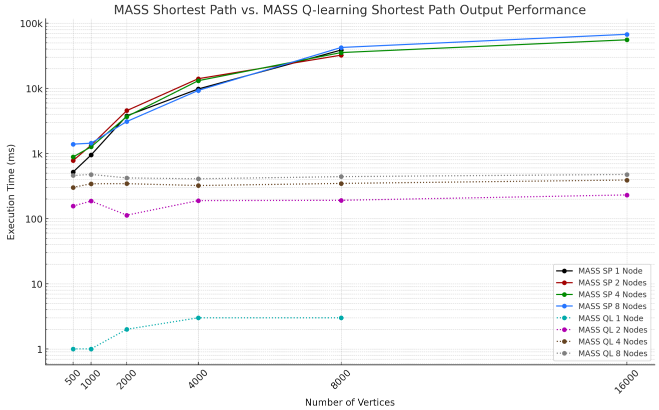


Figure 4. MASS Shortest Path vs. MASS Q-learning Shortest Path

B. Closeness and Betweenness Centrality

The Q-learning approach for centrality analysis demonstrates potential but faces scalability challenges. For closeness centrality, training individual Q-tables for all vertex pairs resulted in quadratic scaling with the number of vertices, making the approach inefficient for larger graphs as shown in Figure 6. Similarly, betweenness centrality incurred additional overhead from enumerating all shortest paths between vertex pairs, further amplifying scalability issues. The scalability issues and the need to retrain extensively for dynamic graph changes made the Q-learning approach less competitive compared to the current MASS implementation.

Overall, while the Q-learning-based centrality method showed promise for small, static graphs, its scalability and

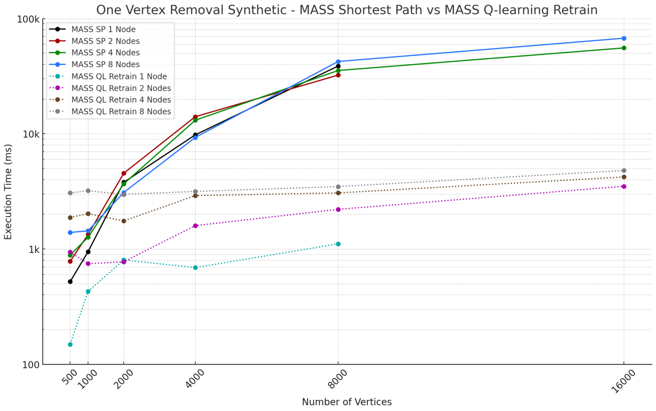


Figure 5. MASS vs. MASS Q-learning Shortest Path One Vertex Removal

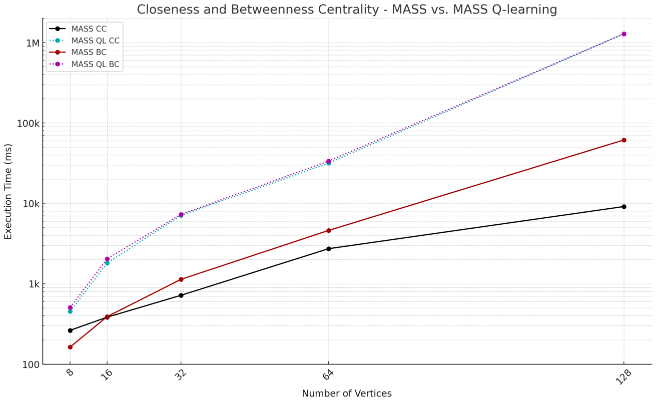


Figure 6. MASS vs. MASS Q-learning Closeness and Betweenness Centrality

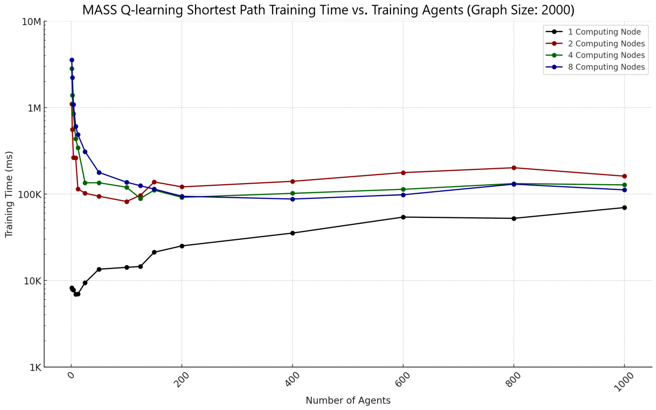


Figure 7. MASS Q-learning Shortest Path Multi-Agent Training Time

adaptability fell short of the success in the shortest path application. Future work could explore hybrid approaches or optimizations, such as combining Q-learning with heuristics or leveraging pre-computed graph partitions, to address these limitations and improve centrality analysis performance.

C. MASS Enabled Q-learning Performance Improvements

The Q-learning shortest path application incorporates key innovations enabled by the MASS framework: multi-agent training, distributed reward window, and dynamic hyperparameter tuning. Multi-agent training, facilitated by MASS's distributed multi-agent architecture, significantly reduces training times by allowing agents to operate in parallel across all computing nodes. As shown in Figure 7, this approach achieved a 190% reduction in training time on an 8-node cluster with 1750 training agents compared to a single agent execution. The MASS framework facilitates this improvement by efficiently managing agent migration, synchronization, and data sharing across distributed nodes, enabling seamless multi-agent parallelism.

In multi-agent Q-learning scenarios, a considerable challenge is the need for agents to share the knowledge gained during training, often requiring costly synchronization methods. Our implementation of a distributed reward window, enabled by the MASS framework, was a key enhancement to the Q-learning applications. Figure 8 illustrates that the distributed reward window allows agents to share their cumulative episodic rewards without costly synchronization methods, enabling efficient hyperparameter tuning and reducing training times. MASS's distributed architecture provides a simple and effective mechanism for aggregating these rewards. By streamlining communication and eliminating costly synchronization overhead, the distributed reward window significantly increases the scalability and efficiency of Q-learning as applied to large-scale graph analysis.

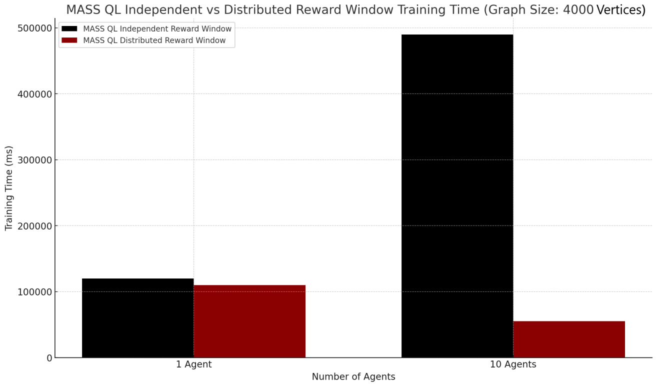


Figure 8. MASS Q-learning Independent vs. Distributed Reward Window

Dynamic hyperparameter tuning further enhances the training process by adjusting exploration and learning rates based on the aggregated rewards from the distributed reward window. Figure 9 demonstrates how this capability minimizes unnecessary computations, reducing overall training time without compromising accuracy. By leveraging MASS's robust distributed communication and state management features, agents can adapt their hyperparameters dynamically, ensuring a balanced and efficient approach between exploration and exploitation.

These improvements underscore the critical role of the MASS framework in enhancing the Q-learning implementa-

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

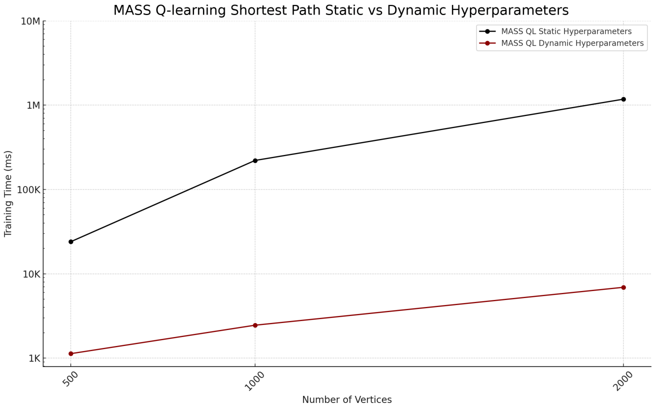


Figure 9. MASS Q-learning Static vs. Dynamic Hyperparameter Tuning

tions. By providing a robust distributed parallel environment, MASS enables efficient multi-agent operations, adaptive learning, and scalable performance for large-scale graph analysis.

V. Conclusion and Future Work

We successfully demonstrated the capability of MASS to accelerate reinforcement learning algorithms like Q-learning through its distributed and parallel multi-agent programming paradigm. In our experimentation, MASS's multi-agent faculties resulted in up to a 190% reduction in training time compared to a single agent, and the distributed architecture enabled Q-learning for large-scale graph analysis. MASS's ability to support distributed graph computing and facilitate adaptive learning presents a promising framework for future multi-agent machine learning applications. Furthermore, the Q-learning applications, particularly the shortest path implementation, exhibited promising performance, especially in their adaptability to dynamic data, and served to highlight where MASS can be improved to better align with machine learning workloads.

Our future work will focus on expanding the scope of machine learning in MASS by exploring advanced graph-based techniques, such as Graph Convolutional Networks and embedding techniques like FastRP. Additionally, research into efficient agent communication, reduced agent overhead, and more dynamic graph modifications could unlock new capabilities and improve MASS's adaptability for a broader range of applications.

Finally, the MASS library and the applications featured in this project are available at: http://depts.washington.edu/dslab/MASS/.

References

[1] G. Malewicz *et al.*, "Pregel: A system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, Association for Computing Machinery, 2010, pp. 135–146.

[2] R. S. Xin *et al.*, *Graphx: Unifying data-parallel and graph-parallel analytics*, 2014. arXiv: 1402.2394 [cs.DB].

[3] M. Fukuda, C. Gordon, U. Mert, and M. Sell, "Agent-based computational framework for distributed analysis," *IEEE Computer*, vol. 53, no. 4, pp. 16–25, 2020.

[4] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[5] Z. Sun, "Applying reinforcement learning for shortest path problem," in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, IEEE Computer Society, 2022, pp. 514–518. DOI: 10.1109/BDICN55575.2022.00100.

[6] X. Wang, L. Jin, and H. Wei, "The shortest path planning based on reinforcement learning," *Journal of Physics: Conference Series*, vol. 1584, no. 1, 012006: 1–6, 2020. DOI: 10.1088/1742-6596/1584/1/012006.

[7] R. Nannapaneni, "Optimal path routing using reinforcement learning," Dell Inc., Tech. Rep., 2020, Accessed: May 03, 2024.

[8] J. Golbeck, "Chapter 3 - network structure and measures," in *Analyzing the Social Web*, Morgan Kaufmann, 2013, pp. 25–44. DOI: 10.1016/B978-0-12-405531-5.00003-1.

[9] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Computing classic closeness centrality, at scale," in *Proceedings of the Second ACM Conference on Online Social Networks*, Association for Computing Machinery, 2014, pp. 37–50. DOI: 10.1145/2660460.2660465.

[10] K. Shukla, S. C. Regunta, S. H. Tondomker, and K. Kothapalli, "Efficient parallel algorithms for betweenness- and closeness-centrality in dynamic graphs," in *Proceedings of the 34th ACM International Conference on Supercomputing*, Association for Computing Machinery, 2020, pp. 10–21. DOI: 10 . 1145 / 3392717.3392743.

[11] D. Ferone, P. Festa, A. Napoletano, and T. Pastore, "Shortest paths on dynamic graphs: A survey," *Pesquisa Operacional*, vol. 37, no. 3, pp. 487–508, 2017. DOI: 10.1590/0101-7438.2017.037.03.0487.

[12] J. Doran and D. Michie, "Experiments with the graph traverser program," *Proceedings of the Royal Society A*, vol. 294, no. 1437, pp. 235–259, 1966. DOI: 10.1098/rspa.1966.0205.

[13] Y. Yang and E. P. Chan, "Shortest path tree computation in dynamic graphs," *IEEE Transactions on Computers*, vol. 58, no. 4, pp. 541–557, 2009. DOI: 10.1109/TC.2008.198.

[14] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 4, 1994, pp. 3310–3317. DOI: 10.1109/ROBOT.1994.351061.

[15] T. Akiba, Y. Iwata, and Y. Yoshida, "Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling," in *23rd International World Wide Web Conference*, Association for Computing Machinery, 2014, pp. 237–248. DOI: 10.1145/2566486.2568007.

[16] H. Cui, R. Liu, S. Xu, and C. Zhou, "Dmga: A distributed shortest path algorithm for multistage graph," *Scientific Programming*, vol. 2021, no. 1, p. 6 639 008, 2021. DOI: 10.1155/2021/6639008.

[17] J. Gilroy, S. Paronyan, J. Acoltzi, and M. Fukuda, "Agent-navigable dynamic graph construction and visualization over distributed memory," in *7th Int'l Workshop on BigGraphs'20*, IEEE, 2020, pp. 2957–2966.

Hybrid Visual SLAM for Multi-session Precise Localization: Application to a Coastal Cliff in Normandy

Diego Navarro [1,2], Ezio Malis [2], Raphael Antoine [1], and Philippe Martinet [2]
[1]ENDSUM team. Cerema Normandie-Centre, France
[2]ACENTAURI team. Centre Inria d'Université Côte d'Azur, France
e-mail: diego-navarro.tellez@inria.fr

***Abstract*—In the context of smart structure maintenance, the positioning of data measurements (e.d. radar, thermal camera, etc..) is crucial. Most of this data is meant to be collected in the proximity of the structure. In this paper, we address the challenge of accurately localizing a drone within GPS-deprived environments. This issue arises particularly near large structures, where the GPS signal can be significantly distorted or entirely absent. One of the most common solutions to the problem is to use a vision sensor and a Simultaneous Localization and Mapping (SLAM) system to reconstruct the environment and localize the drone. However, existing SLAM approaches may not be robust and precise enough, especially when the cameras lose perspective due to the proximity of the structure. In this paper, we propose a novel framework that computes a dense map of the environment to exploit it using direct odometry which excels in precise localization. The main contribution of this paper is the use of dense maps that enable localization in scenarios with narrow perspective. Experiments in realistic simulated environments demonstrate the capability of the system to localize the drone within 16 centimeter accuracy and to outperform existing State Of The Art approaches.**

***Keywords*-robotics; autonomous vehicles; vision and scene understanding; volumetric image representation.**

I. Introduction

Coastal cliff monitoring has become increasingly critical in the face of climate change, as rising sea levels and more frequent extreme weather events accelerate coastal erosion [1]. Cliffs are particularly vulnerable to processes such as landslides and collapses, which threaten both natural ecosystems and human settlements. Monitoring using Unmanned Aerial Vehicles (UAVs) may allows for early detection of instability, helping to mitigate the risks to infrastructure and communities while informing coastal management strategies to adapt to a changing environment. Commonly used onboard monitoring instruments include cameras and LiDAR (Light Detection and Ranging) systems. Recently, geophysical systems have been developed to enable subsurface imaging directly from UAVs. Among them, Ground-penetrating radar (GPR) has proven to be a powerful non-invasive tool for rocky cliffs, enabling 2D or 3D visualization of subsurface conditions and the detection of internal structural weaknesses [2]. For optimal results, the antennas must be positioned very close to the surface, typically within a few centimeters, to maximize signal-to-noise ratio, without compromising the security of the flight. Performing close-range measurements poses a major challenge for conventional localization methods that rely on GNSS systems, especially for UAVs. This scenario introduces several issues regarding the accuracy of the measurements: on one hand, satellite signal occlusion caused by proximity to surfaces or multi-path interference can compromise GPS (Global Positioning System) positioning accuracy. On the other hand, onboard applications are limited by the vehicle's weight capacity and energy consumption, reducing operational efficiency. Among the different tools used for localization in GPS-denied environments, SLAM (Visual Simultaneous Localization and Mapping) methods, especially feature-based sparse approaches, have gained considerable popularity [3]. Notable examples include methods like ORB-SLAM [4], which are particularly advantageous when computational resources are limited. However, these methods face challenges when dealing with repetitive textures or low-textured images.

Prior work has explored the creation of dense maps from sparse key-frames, as demonstrated by Zhang's work on stereo mapping [5]. Zhang and Shu introduced a dense mapping module into the ORB-SLAM2 workflow, enabling the generation of dense maps from the key-frame database. This method explored the post-processing of the mapping session, but it did not address the problem of localization in close-range inspection scenarios. Kerb et al. [6] introduced a novel approach to modeling and rendering dense maps using Gaussian splatting. This work revisited an older rasterization technique, now more feasible due to advancements in hardware architectures and rendering methods. As a result, it reignited interest in dense mapping approaches, leading to the emergence of new SLAM techniques inspired by [6] within the open-source community.

In this paper, we present a V-SLAM framework tailored for close structure inspection using UAVs. To overcome UAV constraints, we employ a two-step system. In the initial step, a stereo camera captures images during a mapping flight, using a feature-based method for ego-pose estimation [7]. The system's modular design allows for flexibility in choosing various ego-localization methods. The resulting dense map serves dual purposes: aiding mission planning and enhancing localization in the second step. In this subsequent stage, precise localization is achieved using a monocular camera, addressing weight and energy constraints posed by embedded measurement equipment such as radar and thermal cameras.

The rest of the paper is structured as follows: In Section II, we will describe the proposed method, beginning by the mapping workflow and then describing the scanning functioning. Then, in section III the results of the method will be presented, as well as the conditions of test. Finally, the

ISBN: 978-1-68558-241-8

performance of the method and future work will be discussed in section IV.

II. Proposed Method

In this section, we present a novel Visual SLAM (V-SLAM) method that integrates feature-based and direct methods to generate a dense global map useful for odometry and navigation tasks. The generated dense map plays a pivotal role in localizing the agent when the SLAM system faces challenges, such as loss of perspective or low-texture environments. Its richness offers an additional layer of information to estimate poses in situations where traditional methods may fail.

During the localization step in our target use case, as the agent approaches structures, the descriptors generated by feature-based methods during the mapping session may become invisible. This typically occurs due to a change in scale of the incoming image flow, resulting in low density data. Our dense map overcomes this obstacle by storing maximal information, enabling the agent to formulate better pose hypotheses.

A. Mapping

1) Input: During the mapping flight the ego-localization system is expected to select a set of key-frames composed of a pair of stereo images $\mathbf{I}_k$ and an estimated pose ${}^{0}\hat{\mathbf{T}}_k$. The selection criteria may vary, but the frames shall respect some conditions for photogrammetry reconstruction such as the overlap and the coverage of the target area. Once the key-frames have been collected, the system carries the point-cloud registration process.

The registration process (see Figure 1) is carried out by aligning the incoming image $\mathbf{I}_k$ with a render of the global map $\mathbf{I}_v$ and its depth map $\mathbf{D}_v$ at the current estimated pose ${}^{0}\hat{\mathbf{T}}_k$. The alignment procedure leverages the map data observable at the current pose $\mathbf{M}_v$.

2) Rasterizer: Inspired by the work of [6], the framework uses the EWA (Elliptical Weighted Average) volume splatting method [8] to render (the process of converting the point-cloud into an image) $\mathbf{I}_v$ and $\mathbf{D}_v$ without need of a mesh. To do so, the EWA method propose a projection of the covariance matrices by means of a projection model linearized around the center of the ellipsoid $\mathbf{u}_j$.

$$\mathbf{J}_j = \begin{pmatrix} \frac{f_{px}}{(\mathbf{u}_j)_z} & 0 & -\frac{f_{px}(\mathbf{u}_j)_x}{(\mathbf{u}_j)_z^2} \\ 0 & \frac{f_{px}}{(\mathbf{u}_j)_z} & -\frac{f_{px}(\mathbf{u}_j)_y}{(\mathbf{u}_j)_z^2} \\ \frac{(\mathbf{u}_j)_x}{|\mathbf{u}_j|} & \frac{(\mathbf{u}_j)_y}{|\mathbf{u}_j|} & \frac{(\mathbf{u}_j)_z}{|\mathbf{u}_j|} \end{pmatrix} \tag{1}$$

Specifically, we use an implementation of the EWA method that suppose an irregular volume which implies non-circular kernels (see Figure 2). As proposed in the original EWA method, the weights are gathered from a lookup table using a radial distance function defined by the conic matrix $\mathbf{Q}_v$ (inverse of the covariance $\mathbf{V}_v$). The radial distance is computed using the ellipsoid equation and the difference in pixels between the center of the ellipsoid projected in the image plane $\mathbf{p}_v$ and the cells inside the kernel $\mathbf{p}_i$ (see Figure 3).

$$r_{vi}(\mathbf{d}_{vi}) = \mathbf{d}_{vi}^T \mathbf{Q} \mathbf{d}_{vi} \quad \text{where} \quad \mathbf{d}_{vi} = \mathbf{p}_v - \mathbf{p}_i \tag{2}$$

The radial function is scaled by the lookup table size so every pixel in the kernel corresponds to a given integer index in the lookup table (r_{vi} in Figure 3).

Nevertheless, as our scenario contemplates a decrease in the point density, instead of deforming the kernel with a low-pass filter (described by an identity matrix), we change the shape of the lookup weights for an exponential decay function in the table w_{tab} (see Figure 3). This allows the rasterizer to fill the unpopulated areas with the information of the neighbors without excessively blurring the points in the center of close kernels.

As the first key-frame is used as reference and therefore directly saved into the map, no rendering is needed. However, for the next frames the map manager will collect the visible map points $\mathbf{M}_v$, their colors $\mathbf{C}_v$ and its respective covariance matrices $\mathbf{V}_v$ into a *virtual* surface that will serve for rendering. The surface is then projected into the image space to generate a new image and depth-map $(\mathbf{I}_v, \mathbf{D}_v)$. This process involves selecting a weight from the look-up table w_{tab} based on the radial distance index derived from the ellipsoid equation r_{vj}.Each color value $\mathbf{c}_v$ is then aggregated, weighted by the selected weight w_{vj}.

$$\mathbf{c}_j = \frac{1}{\sum^v w_{vj}} \sum^v w_{vj} \mathbf{c}_v \quad \text{where} \quad w_{vj} = w_{tab}(r_{vj}(\mathbf{d}_{vj}))) \tag{3}$$

3) Dense visual odometry (DVO): Then, the DVO module [9] computes the pose ${}^{v}\tilde{\mathbf{T}}_k$ that represents the estimated error between the estimated pose and the real pose. The corrected pose of the Keyframe can then be composed from the estimated pose and the error estimation.

The corrected pose is then used to align the incoming point-cloud data with the global map data. This map is build as a graph of *surfaces* $\mathbf{S}_s$, composed of a set of 3D points $\mathbf{M}_s$, a set of colors $\mathbf{C}_s$, a set of covariance matrices $\mathbf{V}_s$ and a pose ${}^{0}\hat{\mathbf{T}}_s^*$. This serves to model a fictitious camera that organizes the data in a grid corresponding to an image plane $\mathbf{I}_s$. This simplifies the process of updating the map and the selection of points to render.

4) Point-cloud computation: Once the pose is corrected, a new data-cloud is computed and filtered. In the current implementation, the depth-maps are computed by two algorithms chosen depending on the scenario. The Stereo Block Matching algorithm of the OpenCV [10] library for unstructured environments and the CREStereo pre-trained model for structured environments. The new depth-map is compared with the existing data in the map, if this information is too different, that region of the map is considered an anomaly. Abnormal depth estimations can appear for many reasons, specially with StereoBM, errors in the depth estimation or unveiling of occluded objects. Spurious information in known regions of the map tends to be visually similar but spatially incoherent. In contrast, revealed object are both visual and

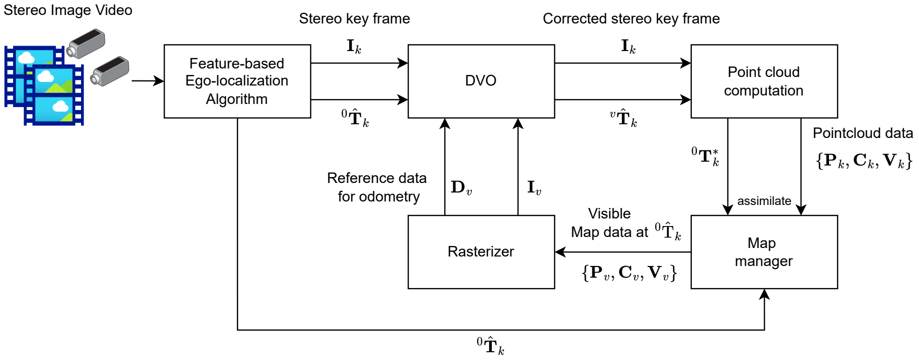

Stereo Image Video
Feature-based Ego-localization Algorithm
Stereo key frame
$\mathbf{I}_k$
${}^{0}\hat{\mathbf{T}}_k$
DVO
Corrected stereo key frame
$\mathbf{I}_k$
${}^{v}\hat{\mathbf{T}}_k$
Point cloud computation
Pointcloud data
$\{\mathbf{P}_k, \mathbf{C}_k, \mathbf{V}_k\}$
${}^{0}\mathbf{T}^{*}_k$
assimilate
Reference data for odometry
$\mathbf{D}_v$
$\mathbf{I}_v$
Visible Map data at ${}^{0}\hat{\mathbf{T}}_k$
Rasterizer
$\{\mathbf{P}_v, \mathbf{C}_v, \mathbf{V}_v\}$
Map manager
${}^{0}\hat{\mathbf{T}}_k$

Figure 1. Workflow for the mapping phase. Most computations are currently computed offline due to constraints related to the current implementation. One frame takes 300 seconds to be treated, execution time profiles indicate that 90% of the time is expended in object instances creations and single-threaded operations that could be parallelized.

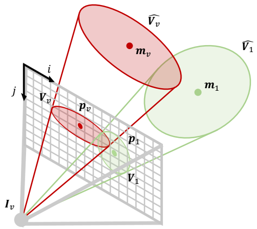


Figure 2. The 3D covariance matrices in camera frame $\hat{\mathbf{V}}_1, \ldots, \hat{\mathbf{V}}_v$ are mapped as ellipsoid regions in the image space $\mathbf{V}_1, \ldots, \mathbf{V}_n$ through perspective projection linearized around the center of the ellipsoids $\mathbf{m}_1, \ldots, \mathbf{m}_v$ whose coordinates in the image plane are $\mathbf{p}_1, \ldots, \mathbf{p}_v$. Then the colors are merged by means of a weighted average.

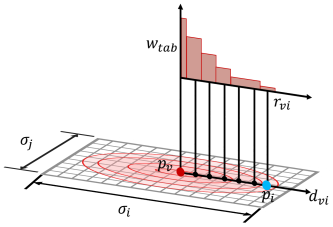


Figure 3. The weights are gathered from a lookup table using a radial function defined by the covariance matrix. This distance is computed for each pixel covered by the ellipsoid of the kernel $\mathbf{p}_i$ around the projected center $\mathbf{p}_v$. The size of the kernel is determined by the covariance matrix.

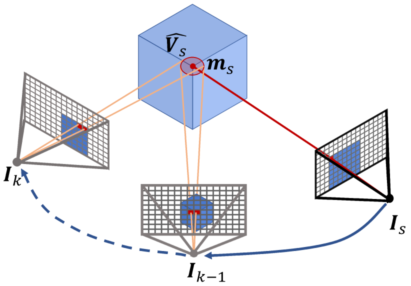


Figure 4. point-cloud alignment is achieved through the minimization of distance between the renderings at the estimated positions $\mathbf{I}_v$ and the key-frame images $\mathbf{I}_k$. The set of points that fall in the same pixels of $\mathbf{I}_s$ are used to update the covariance matrix of the map ellipsoids $\mathbf{m}_s$. The projection done is similar to the rendering process (linearization around the center of the ellipse).

spatially incoherent. This can be exploited to reject defective stereo images of noisy depth computations at the same time that legitimate new information is kept.

5) Map manager: Accepted data-clouds are then processed by the map manager which decides either to generate a new surface or update an existing one. This decision is done based in multiple criteria. The first criterion is the distance from the last reference surface, a straightforward measure ensuring proximity for alignment in image space. The second one is the overlap of the reference surface and the incoming point-cloud. If visible area of the key-frames changes significantly, a new surface is generated to maximize information gathering. The third criterion is a custom measure based on the coverage of interesting areas. Even with substantial overlap, is still possible to neglect some interesting areas observed just once or twice during the mapping session. The current implementation uses saliency maps computed by the Static Saliency module of OpenCV, a human-centered measure of local image relevance, but objective relevance metrics will be tested to enhance the probability of convergence in localization algorithms. At the moment, the system computes a salient map of the incoming point-cloud and compare it with the reference surface saliency map projected in the same image space. If interesting areas are not covered by the reference surface, the system will generate a new one.

Finally, there is a pixel size check. Lower depth values indicate that the camera is closer to the surface. When this occurs, a new node is created and initialized with data from a neighboring node. The covariance matrices for the new node are adjusted based on the characteristics of the incoming image, ensuring that the map keeps all available information to enhance performance during the next phase.

New key surfaces $\mathbf{S}_s$ are initialized with the information

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

from the surrounding surfaces and the incoming point-cloud. The matches of points are obtained through alignment in the image space. By projecting the 3D ellipsoids into the same image plane, the system can compare the points of the surfaces with the incoming point-cloud(see Figure 4). In the empty cells, the grid assimilates the values of the neighbors or the incoming point-cloud. In populated cells, the new value is averaged with the current one if the distance between them is within a certain threshold. Otherwise, the system overrides the existing point with the new one. This update process mimics the splatting method for rendering so the misalignment of image pixels and reference pixels do not affect the quality of the update. The covariance matrix $\mathbf{V}_s$ is merged with a new one with the shape of the current pixel size. The center of the ellipsoid is updated as the result of an iterative mean computation from all the point observations after alignment. Currently, the fusion of the color information $\mathbf{C}_s$ is done with a mean computation in the LAB color space with the intention to better capture the perceptual changes in the color. After processing all key-frames, the system has generated a surface graph that will be used in the localization step.

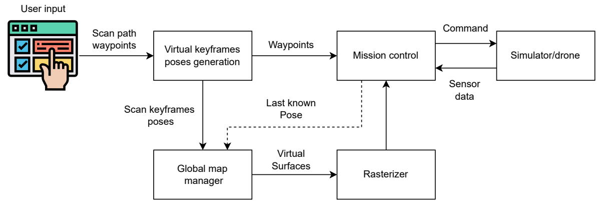


Figure 5. The global map manager will take any pose and generate a virtual surface that will be used to generate a depth and color images necessary for the odometry.

B. Localization

1) Input: In this step, a scanning path is defined either by an algorithm or by the user. In the current state of the framework, the user shall choose the shape of the scan path. After defining the way-points of the scanning path, the mission control will generate a set of virtual key-frames to assist the scanning operation.

2) Virtual frames generation: First, the map produced by the offline registration is loaded. Then, the system generates a set of frames with a co-visibility criteria (see Figure 5). By projecting the scanning path's lines onto the virtual image plane, the visible distance covered by the virtual camera's FOV (field of view) is computed. With this information, the next pose of the virtual camera is chosen by overlap criterion. In the current implementation, the path generator computes the portion of the parametric line between two way-points that is covered by the virtual camera FOV. The selection of the point along this line is based on achieving the specified percentage of overlap. This chosen point becomes the image center for the next virtual image frame. The path generator maps this pixel to the 3D space and adds an offset to get the next pose of the virtual camera. This offset places the camera at a distance behind the scan trajectory, relative to the expected position, allowing the image to cover a larger area than the real camera will capture. This approach ensures that incoming images along the scan path can be reliably tracked from at least one virtual key-frame.

3) Frame management: During the scan path, the global map manager module picks the closest virtual key-frame for localization. The selection is based on an estimation of the current pose got by the mission control module. If the DVO module fails to converge, a new virtual key-frame is generated by the rasterizer at the last known pose to restart the tracker. Since the DVO module estimates the pose from the incoming images to the reference virtual key-frames, the system can recover the global pose without accumulating any errors. The localization is carried out with the same direct odometry method used during the global map generation. However, it is worth noting that the DVO module can localize the agent using different camera configurations than those used during mapping. This can be done by configuring the rasterizer to emulate the camera settings from the incoming images.

III. Results

This section presents the results of the experiments to characterize the performance of the method. First, the test conditions and the baseline for comparison are described, followed by a discussion of the method's performance.

A. Simulations with realistic data

The data used for the experiments was collected in a simulated environment in Airsim [11] Since our color profile for the color fusion is too simple, we decided to carry the simulation in a lighting-controlled environment. The observed site is the Sainte-Marguerite-sur-Mer cliff (Normandy), monitored in the framework of the Defhy3geo project [12]. The scene is composed by a segment of a cliff model reconstructed using the Agisoft [13] metashape software with centimeter-lever precision. A drone acquired geo-referenced images, allowing the generation of a 3D model of the cliff. The effective visible area is equivalent to a 60x20 meters wall with non-structured texture. This map demonstrates the performance of the system in low texture environments.

To benchmark the performance of our method, we use ORB-SLAM3 [7] in localization mode as a baseline. In this mode, the system focuses solely on agent localization, sacrificing mapping capabilities for computational efficiency. Both methods are submitted to the following test. First, a set of key-frames is selected during a *mapping flight*, a path far from the surface to inspect and done in an arbitrary way to simulate human pilot behavior. Then, the drone performs a "scanning flight", a path close to the surface with a structured path (see Figure 9) to simulate a measuring mission. We first tried to perform the scan mission with the ORB-SLAM3 in monocular mode, but the system gets lost when the agent is too close to the mapping trajectory. Then, we tried to perform the mission in stereo mode. Figure 6 displays the absolute translation error (ATE) in meters during the scan mission in this mode. Despite

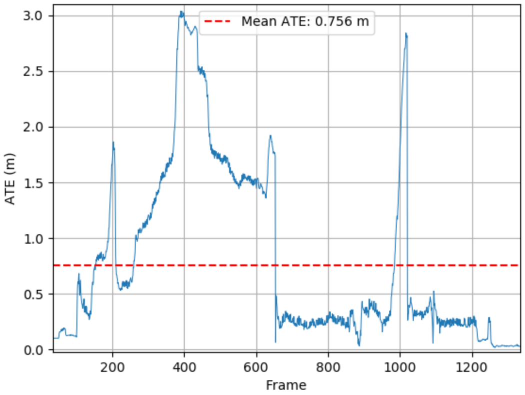


Figure 6. ORB-SLAM3 ATE (absolute translation error) in meters during the scan path with stereo camera. Being stereo the best configuration, the closest ORB-SLAM3 could get to the structure is 5 m from the structure, i.e. 8 m from the mapping path.

being the best configuration, the closest ORB-SLAM3 could get to the structure was 5m from the structure (8 meters from the mapping path). Prior to establishing correspondence with the map, localization error may escalate to 3m with possibility of recovery. Upon successful alignment with the map, the error returns to 15cm. Although the system is capable of maintaining this level of accuracy post-alignment, its performance falls short for close-range structure inspection.

The hardware used for the experiments is a custom-built computer with a Xeon processor and an Nvidia Quadro P2200 GPU. For the moment, the mapping is too slow for online use. The time of execution can be improved once the system is implemented with optimized code architecture for parallel and image processing. Our framework is built upon the following software components:

- Airsim as the simulation environment [11].
- ROS for communication between different software components.
- A Python-based offline registration node for map generation.
- A direct odometry algorithm from the OpenROX library (developed by the ACENTAURI team).
- A mission control module to load the map, generate the virtual frames and control the drone.

B. Performance

The first outcome of our method is the dense map. Figure 7 depicts the dense map generated by our approach alongside the reference ground-truth model. This map effectively captures the structural details of the environment as well as color information. Comparing the generated dense map with the ground truth, 50% of the points have less than 10cm of error, 36% are under 20cm and 10% under 30cm. In summary, the mean error of the points in the map is 11cm. The second result of our method is the capability to generate virtual frames.

In Figure 8, one can see a virtual frame generated by our method. As shown, our method is capable of generating useful images even with relative low density point-clouds. Our method's third contribution is global pose estimation, demonstrating the system's resilience in adverse conditions. Figure 9 illustrate the dense map with the set of virtual reference poses. The Figure 10 shows that the agent was able to maintain the global pose estimation with mean ATE (absolute translation error) of 0.16m at 2m from the structure.

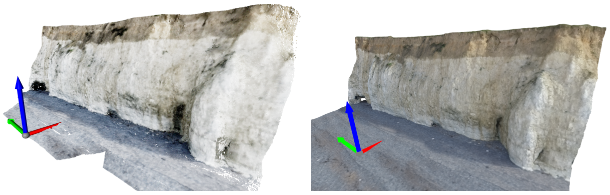


Figure 7. The map generated during the offline registration can be used to provide information for subsequent navigation tasks. The left image shows the dense map generated by our method, the right image shows the ground truth model.

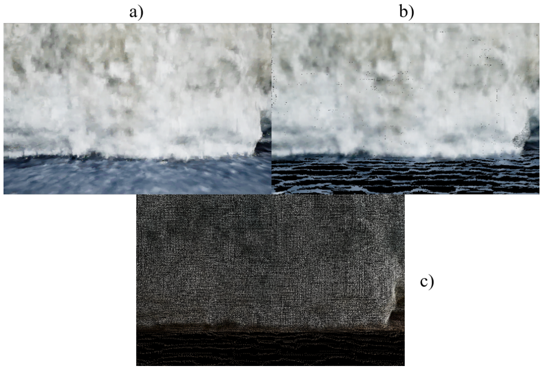


Figure 8. Virtual frames. a) Image seen by the camera, b) image rendered from the dense model and c) centers of the kernels used for rendering.

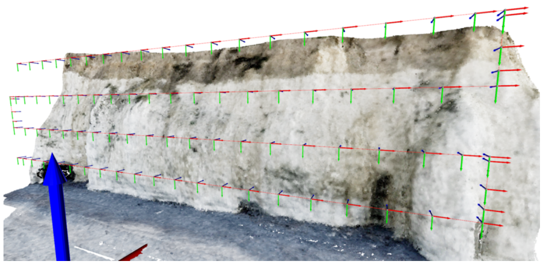


Figure 9. Path followed by the agent during the scan mission, a set of virtual poses is generated and a virtual image is rendered.

Pose estimation accuracy depends on the stereo pair-derived point cloud quality and dense odometry module accuracy. While odometry is fairly precise, depth estimation can be improved. The significant error between frames 1300 to 1500 in Figure 10 results from poor map quality in that area. This region has not been correctly modeled during mapping, which shows as under-performance of the registration module. Despite this, our method exhibits a accuracy similar to that of ORB-SLAM3 but operates at a distance 4 meters closer to the structure (2m

ISBN: 978-1-68558-241-8

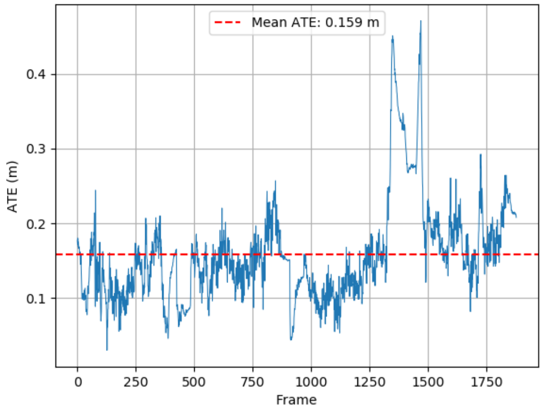


Figure 10. Our method exhibits consistent ATE behavior over 1900 frames, showing stability at various distances across multiple experiments. However, a notable shift occurs when the model is observed in regions with extremely low-texture conditions.

from the structure and 11m from the mapping path). It is worth mentioning that the position estimation results showed are not treated by any kind of filter, the error metrics were computed only with the output of the odometry module which computes an instantaneous pose estimation from the last estimated pose but does not keep track of the dynamic evolution of the images.

IV. Conclusion and Future Work

In this paper, we introduced an innovative framework designed to tackle the challenge of localization close to structures where traditional methods may fail. Our approach focuses on creating and using a dense environmental map to enhance accuracy, particularly in scenarios involving narrow-perspective cameras and low-texture images. We achieve this by accurately estimating camera ego poses and applying the projection model for efficient matching of dense point clouds and generation of synthetic images.

This versatile map proves invaluable not only for localization but also for mission control and cost-effective creation of digital twins for multi-session autonomous navigation. In the current version of our method, we seek virtual key-frames based on pose estimations, which may deviate from actual surface positions. To further strengthen our system, we plan to integrate sensor fusion techniques, such as Ultra-Wideband (UWB) and Inertial Measurement Units (IMU), streamlining surface identification and simplifying the odometry initialization process. Likewise, ongoing work aims to improve the characterization of the surface modeling data-structures and data aggregation.

Despite these advances, several challenges remain. Although a dense map augments the robustness of the system across sessions, this map representation could benefit from richer visual data-representations. First, color information modeling can be improved by considering the dynamic nature of outdoors illumination across sessions. Secondly, the registration method could be formulated in a more effective way, instead of an optimistic computation of the covariance matrices with aligned point-clouds, the map could capture this information with a more realistic representation and less susceptible to errors of depth and pose estimation.

The results show that the quality of the localization depends heavily on the quality and coverture of the map. This implies that the criteria to create new map nodes are critical in the creation of a good quality map. Ongoing work focuses on the detection of relevant information in the surfaces, detail-centered registration and memory optimization. Likewise, the creation of reference of key-frames during localization present a research venue worth exploring. So far, the main challenge is to formulate data-wise criteria to efficiently manage the map registration and the reference rendering.

While the results presented here are based on simulations with realistic data, ongoing work has shown promising results in real-world scenarios, which align with the findings presented in this study.

References

[1] J. R. Shadrick *et al.*, “Sea-level rise will likely accelerate rock coast cliff retreat rates”, *Nature Communications*, vol. 13, no. 1, p. 7005, 2022.

[2] A. D. Switzer, C. Gouramanis, C. S. Bristow, and A. R. Simms, “Ground-penetrating radar (gpr) in coastal hazard studies”, in *Geological Records of Tsunamis and Other Extreme Waves*, Elsevier, 2020, pp. 143–168.

[3] Y. Dai, J. Wu, and D. Wang, “A review of common techniques for visual simultaneous localization and mapping”, *Journal of Robotics*, vol. 2023, no. 1, p. 8 872 822, 2023.

[4] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”, *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2016. DOI: 10.1109/tro.2017.2705103.

[5] B. Zhang and D. Zhu, “A Stereo SLAM System With Dense Mapping”, *IEEE Access*, vol. 9, pp. 151 888–151 896, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3126837.

[6] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering”, *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, vol. 42, no. 4, Jul. 2023.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM”, *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, ISSN: 1941-0468. DOI: 10.1109/TRO.2021.3075644.

[8] M. Zwicker, H. Pfister, J. van Baar, and M. Gross, “Ewa volume splatting”, in *Proceedings Visualization, 2001. VIS ’01.*, 2001, pp. 29–538. DOI: 10.1109/VISUAL.2001.964490.

[9] A. I. Comport, E. Malis, and P. Rives, “Real-time Quadrifocal Visual Odometry”, *The International Journal of Robotics Research*, vol. 29, no. 2, pp. 245–266, 2010. DOI: 10.1177/0278364909356601.

[10] G. Bradski, “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.

[11] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles”, *CoRR*, vol. abs/1705.05065, 2017. arXiv: 1705.05065.

[12] T. Junique *et al.*, “Investigation of the geological and hydrogeological structure of chalk cliffs with visible, thermal infrared and electrical resistivity imaging”, *Journal of Hydrology*, vol. 630, p. 130 642, 2024.

[13] Agisoft LLC, *Agisoft metashape*, 2019.

Automated Coupling of Freight Wagons using a Road-rail Vehicle and Innovative Communication and Positioning Technologies

Robert Richter, Oliver Michler
Technische Universität Dresden, "Friedrich List" Faculty of Transport and Traffic Sciences, Chair of Transport Systems Information Technology
Dresden, Germany
email: robert.richter@tu-dresden.de
email: oliver.michler@tu-dresden.de

***Abstract*—The focus of this work is the development of a comprehensive system for the automation of shunting processes in marshalling yards, based on an electrically driven dual-mode (Road-rail) vehicle as an autonomous shunting vehicle (ASV) and a highly available localization and communication system. Possible areas of application are shunting locations where a low volume of traffic does not justify large shunting yards. In addition, individual subsystems can also be used across different modes of transport, as key innovation pillars - secure communication and reliable localization - are also essential prerequisites for highly automated driving in road transport. The associated research project AZubiG lays the foundation for electromobile logistics in rail freight transport. AZubiG enables automatic and highly flexible 24/7 operation in shunting and rail operations. This strengthens the competitiveness of rail transport by improving the "last mile" and exploiting its ecological and economic advantages. The main requirements were a system architecture for reliable localization and secure communication, system design, implementation and integration of the electronic components for the implementation of automated shunting operations. In functional terms, this affects the shunting vehicle on the one hand and the freight car itself during the coupling process on the other. In specific terms, the design of the components of the positioning system based on a radio sensor network (WSN) consisting of permanently installed anchors and mobile tags is explained in this context. The TDoA (Time Difference of Arrival) method proved to be suitable for dynamic deployment scenarios. However, this requires highly synchronized infrastructure points, which are usually implemented using wired solutions. In the railroad sector, cabling of this type is difficult to implement, as infrastructure points are required on both sides of tracks and subsequent cabling is therefore not feasible or economical. For this reason, a concept was developed that is completely wirelessly synchronized and uses ultrawideband (UWB) runtime signals between the anchors in order to synchronize them with each other continuously. In the end, the work presents laboratory and field tests of varying granularity (railway laboratory, railway research hall, research operating yard) and their evaluations in terms of feasibility of an autonomous shunting vehicle.**

Keywords- Railway; automated shunting; marshalling yard; shunting yard; automation; intelligent transport systems; autonomous shunting vehicle (ASV); UWB; TDoA

I. INTRODUCTION

Today's transportation systems face a multitude of complex social, economic, environmental and technological challenges. While innovative and disruptive technologies, such as the use of Unmanned Aerial Vehicles (UAVs) in transportation, are opening up new opportunities, historically evolved transport infrastructures continue to be of central importance. An outstanding example of this is rail transport, whose roots go back to the early 19th century. A significant milestone was the commissioning of the first steam locomotive with 36 attached wagons for public rail transport between Stockton and Darlington in 1825 - an event that laid the foundations for the modern rail industry and continues to shape global passenger and freight transport today [1], [5]. While rail freight transport makes an important contribution to sustainable mobility thanks to its high energy efficiency and comparatively low emissions, many processes still face considerable challenges. Particularly in the areas of shunting and train composition, numerous procedures are still characterized by manual processes that are both time-consuming and inefficient. The digitalization and automation of these workflows offer enormous potential for optimizing logistics chains, increasing transport capacities and reducing operating costs. The increased use of modern technologies such as artificial intelligence, automated coupling systems and digital control systems could help to make rail freight transport more efficient, economical and environmentally friendly. In this regard, the Section II of the article examines the current status of shunting in a marshalling yard and discusses the shortcomings of current shunting systems. Building on this, Section III introduces the autonomous shunting vehicle (ASV) and explains the components relevant for automation, as shown in Fig. 1. In addition, Section IV takes a closer look at concepts for secure, high-availability localization and communication. The article concludes with a discussion of current laboratory and field tests in a railroad research depot (City of Dresden, marshalling yard, Germany) in Section V and further work in Section VI.

II. STATE OF ART - AUTOMATION OF SHUNTING PROCESSES IN MARSHALLING YARDS

As highlighted in [2], [3], current shunting systems face significant limitations that hinder their efficiency and

ISBN: 978-1-68558-241-8

adaptability. This is particularly critical given the increasing demand for freight transport, with rail freight playing a crucial role as a low-emission alternative. Additionally, the ongoing digitalization of rail transport and the integration of smart freight wagons exert growing pressure on shunting yard systems to incorporate digital solutions. The presence of media discontinuities within these systems would compromise the feasibility of an end-to-end digital transport chain.

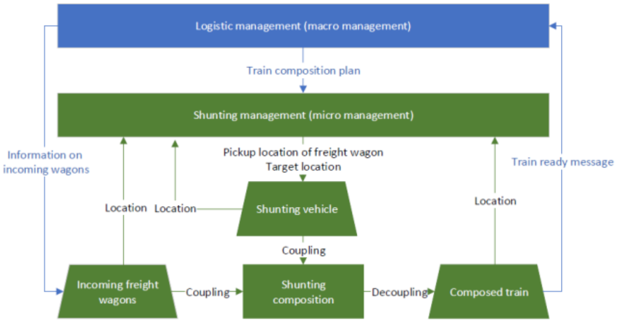


Figure 1. Abstraction of the automated shunting process [2].

A. *Overview of relevant components for shunting*

We have the following components for shunting in this order:

- Shunting management process
- Localization especially multisensor data fusion of Realtime Kinematic Satellite based localization (RTK GNSS), Inertial Navigation System (INS) and Wireless Sensor Network (WSN) for seamless localization in-/outdoor
- Communication Systems especially leaky coaxial cables
- Obstacle and railtrack detection by AI-camera
- Digital Automated coupling
- Smart infrastructure and onboard systems
- Human machine interface
- Safety and security concept
- …

In the following, we will focus on the explanation of components for understanding and specifically on scientific and technical innovations. The systematic shunting process with its individual stages is shown below. In this regard, the coupling process requires a particularly high degree of accuracy in the position and alignment of the ASV

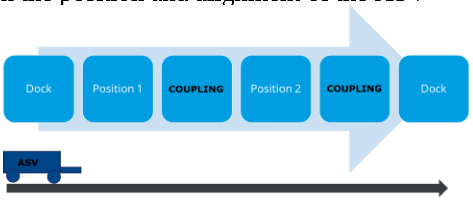


Figure 2. Shunting process

B. *Introduction of Wireless Sensor Networks for Positioning and Navigation*

In order to realize an autonomous coupling process indoor and outdoor in freight yards, a WSN is adapted for positioning in combination with RTK GNSS and an INS combined in an Onboard Unit (OBU) [6]. The technical basis for this was the recent increase in miniaturization and the progressive development of microsystems technology for the construction of highly complex sensor nodes, which are combined as individual systems in an infrastructure-free and self-configuring WSN based on Ultrawideband (UWB).

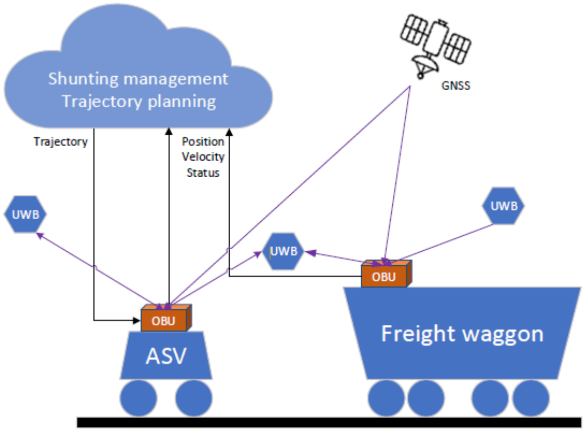
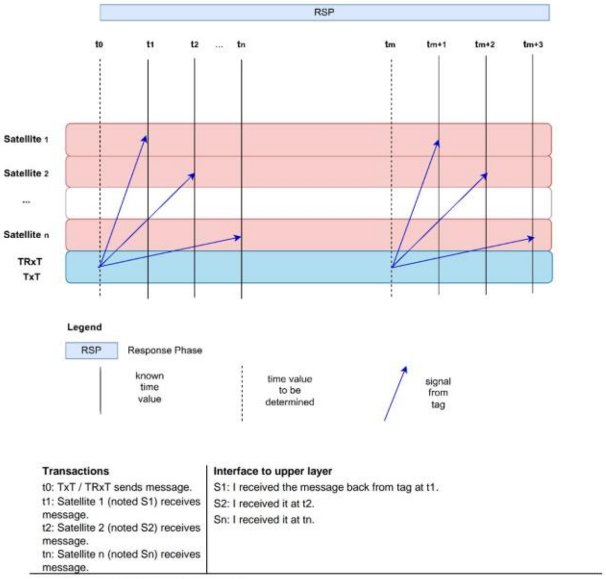


Figure 3. System overview for the ASV and freight wagon technologies required for an autonomous coupling process [2]

In this context, the focus in the following will be on positioning and navigation with radio sensor networks and not on RTK-GNSS (a sufficiently well-known and established method). In generally broadband radio systems can be used for providing distance relations in the form of distance and angle measurements in corresponding real-time performance e.g. on the basis of the licensed ultra-wideband spectrum (UWB). Within a WSN typically with at least four permanently installed anchor nodes, an ASV equipped with a tag can autonomously run a corresponding necessary trajectory to the track and then to the wagon. The anchor nodes are statically attached in pairs to appropriate infrastructure in buildings or outside in the marshalling yard, e.g. to lanterns (see Figure 4). A total of 6 indoor anchors and 12 mobile outdoor anchors were used for the test area. These are PoE-capable and time-synchronized to a central unit.

Figure 4. One of six installed UWB indoor Anchors in the test area

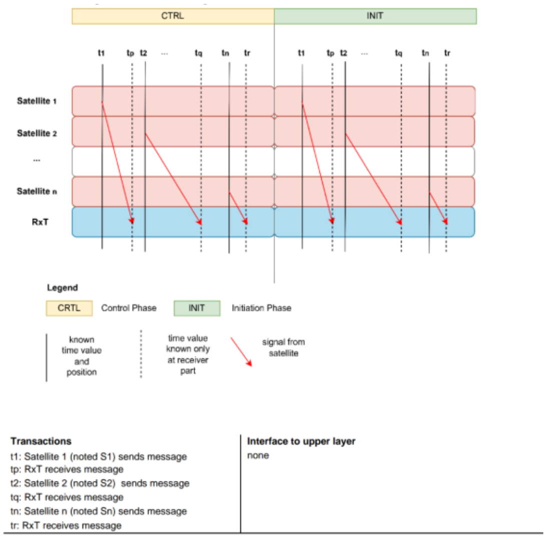
In principle, the tags fitted to vehicles can also change their role to an anchor under appropriate boundary conditions, such as when a vehicle is stationary. nodes can be integrated, for example, in the clothing of forest workers or directly on mobile devices such as drones, chainsaws. Operationally and algorithmically, static and mobile nodes can switch properties [2]. The shunting vehicle can move freely and must therefore be located freely in 2-dimensional space regardless of track layouts and environments. The accuracy and the update interval should therefore be as high as possible. During rerailing, it is necessary to localize in the cm range. The control system runs on the shunting vehicle itself and therefore the position must also be available on the vehicle itself. A direct calculation of the position on the vehicle would be useful so that external influences and interference can be excluded and the fusion of wireless localization with intertial sensors etc. can be carried out with very low latency. The up-link time-difference-of-arrival method is used to localize the wagons, while the shunting vehicle is to be located using the down-link time-difference-of-arrival method. Both methods are described in more detail below. With UL-TdoA (see Figure 5: Uplink TdoA in CoreZone Specification), a tag (usually battery-operated) sends a message via UWB, which is received by the infrastructure. A single packet is sufficient for localization, which makes the implementation very energy-efficient. The position is then determined by algorithms in the server, which evaluate the runtime differences in the installed satellites (infrastructure on the site, which is permanently installed and whose positions do not change). The position calculation itself therefore takes place centrally and can be saved directly on the server.



Transactions	Interface to upper layer
t0: TxT / TRxT sends message. t1: Satellite 1 (noted S1) receives message. t2: Satellite 2 (noted S2) receives message. tn: Satellite n (noted Sn) receives message.	S1: I received the message back from tag at t1. S2: I received it at t2. Sn: I received it at tn.

Figure 5. Uplink TdoA in CoreZone Specification

With the DL-Tdoa method, on the other hand (see Figure 6: DL-TdoA in CoreZone Specification), the individual components of the infrastructure send their own signals permanently (every 125ms) with additional information on position and time synchronization. These signals are used among each other to keep the entire network time-synchronized on a UWB basis. At the same time, by receiving these signals at an end point, the own position can be calculated using the runtime differences. However, this process requires more energy, as a UWB transceiver in the receiver module has an increased energy requirement and the reception of such messages must be permanently guaranteed. The shunting vehicle provides a receiver with sufficient energy, as the energy required to move a wagon or the vehicle itself is much higher and the vehicle must therefore generally be charged regularly. The onboard calculation can therefore be carried out centrally on the vehicle and made available in combination with inertial sensors without high latencies.



Transactions	Interface to upper layer
t1: Satellite 1 (noted S1) sends message tp: RxT receives message t2: Satellite 2 (noted S2) sends message tq: RxT receives message tn: Satellite n (noted Sn) sends message tr: RxT receives message	none

Figure 6. Downlink TdoA in CoreZone Specification

This technique enables energy-efficient, highly dynamic and flexible 2D positioning of the shunting vehicle in appropriately equipped areas of marshalling yards and buildings. In this context a special feature is realized, the challenge of seamlessly locating the ASV from indoors to outdoors and vice versa. These areas are as shown in the following Figure 7, for example roll-up doors.

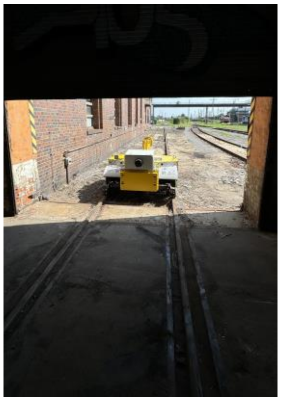


Figure 7. indoor/outdoor transition

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

III. AUTONOMOUS SHUNTING VEHICLE (ASV)

In generally, the shunting process with an ASV is illustrated in the following subtasks (see Figure 8). First, a preliminary positioning of the ASV is initialized on the basis of existing positioning signals. This is already based on integrity approaches based on technologies such as GNSS, LCX, WSN), some of which are independent of each other. In this phase, an operator also specifies a destination point for rerailing on the track plan and a travel trajectory is then computed. The ASV then travels autonomously along this trajectory to the destination point and a new phase of fine positioning on the track begins.

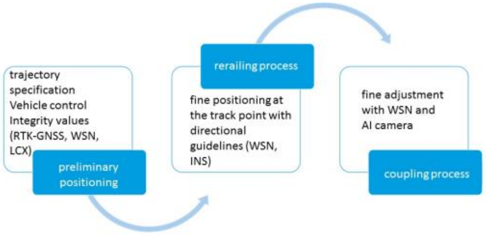

trajectory specification
Vehicle control
Integrity values (RTK-GNSS, WSN, LCX)
preliminary positioning
rerailing process
fine positioning at the track point with directional guidelines (WSN, INS)
fine adjustment with WSN and AI camera
coupling process

Figure 8. Shunting process scheme

This is essential so that the rerailing process can be realized. A very precise angle-based alignment in +/- 1 angle accuracy of the ASV is absolutely necessary here. This is achieved through the combination of INS and an AI-supported camera that recognizes the tracks. Once again it is made clear that in this context it is not the absolute position accuracy on the track that is relevant but only the exact orientation of the ASV at a track point. Once the ASV has rerailed, the coupling process can begin.

A. Technologies Used on ASV

The two-way vehicle used is a vehicle with so-called tank steering, i.e., no axle is steered. Steering movements are achieved by speed differences between the left and right wheels, which also enables the vehicle to turn on the vertical axis. According to a literature search, it was found that a Stanley controller [8] can be used for these vehicles under certain conditions [9] (see Figure 10).

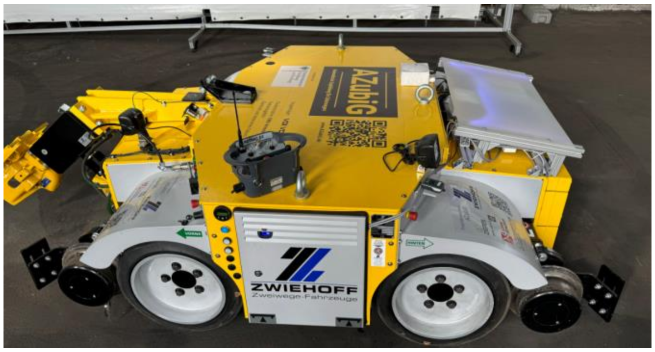


Figure 9. ASV side view

To control the vehicle, the Stanley controller had to be integrated into the vehicle in such a way that it can control the vehicle in accordance with the tractor specifications. Due to the system, this was realized via the technology of the control unit, specifically the radio remote control (FFB), which an operator uses to send driving commands to the vehicle (see Figure 9). The receiver unit of this FFB on the vehicle converts the specified control commands into signals on the vehicle bus (CAN bus). It is at this point that a CAN gateway is installed, which reads the CAN bus coming from the receiver unit and transmits a modified bus with the same messages to the vehicle. This allows the control commands for steering and driving to be manipulated or specified by a controller.

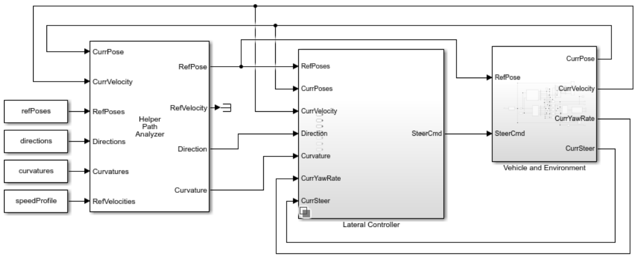


Figure 10. Simulation model of the control loop consisting of Stanley controller and kinematic vehicle model of a tank steering system.

The control center sends the desired movement trajectory via WLAN. This data must be received and processed so that it can finally be used by the controller to control the vehicle. Real-time capable hardware and software was used for this purpose. Specifically, this involves a CarPC “Mexcom VTC7230-BK”, which receives the data from the railroad area via a standard wireless LAN (WLAN). In addition, the CarPC runs software that processes the data received from the control center and makes it available to the real-time capable prototype control unit “dSpace MicroAutoBox II” (MABX) via a UDP Ethernet connection. A CAN gateway was implemented on the MABX. In addition, the Stanley controller runs on the MABX so that it can manipulate the control signals on the CAN directly in order to keep the latency times as low as possible. The laboratory tests and the tests in the railroad environment were successful in terms of real-time capability and control of the ASV (see Figure 11).

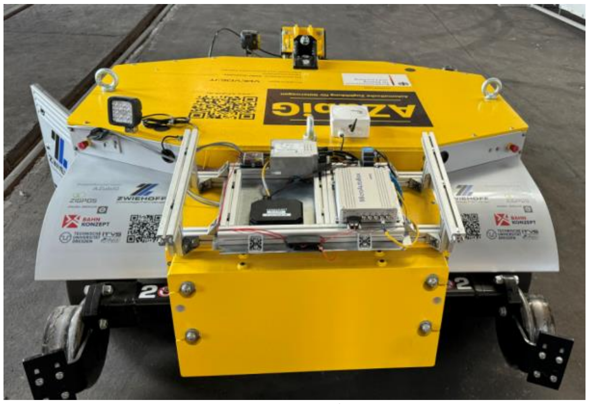


Figure 11 ASV integrated hardware for autonomous driving

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

B. *Technical Realizing of the Rerailing Procedure*

The system interfaces for communication, localization and control were used to create a corresponding configuration and control interface for a control center.

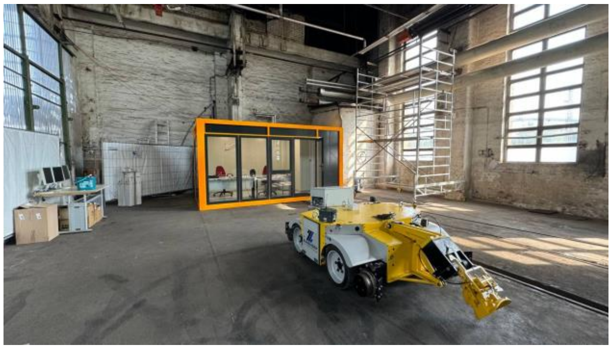


Figure 12. Operation control center and ASV

In addition to entering the basic information and parameters for the shunting area, other areas such as the single-track area were also implemented for driving operations. This area is specially designed (like a railroad crossing) so that the vehicle can drive into it in road mode and then access the track. In addition to the rerailing points, there are other important points for the operation of a vehicle, e.g. the shelter or the loading point. Everything the operator needs to enter such points was also implemented in the control center and tested using the defined test sites in Dresden and Wustermark. Once the system has the basic information for route planning, it implements the path planning using the implemented controller so that destination points can be approached.

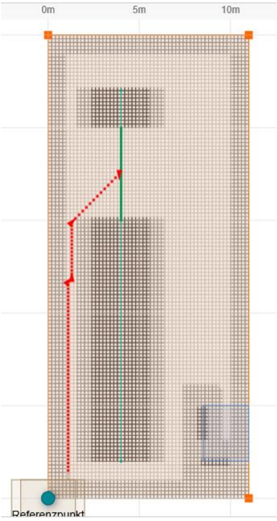


Figure 13. Control Center dashboard with trajectory computing

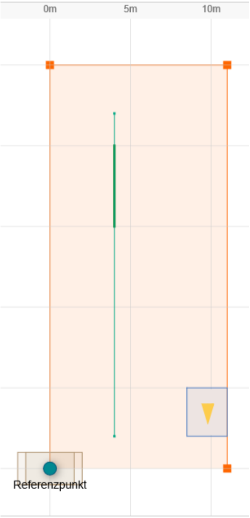


Figure 14. Control Center dashboard with ASV position and heading (MAP view)

The red arrows in Figure 13 clearly show that the path is planned towards the middle of the rerailing area, and that the vehicle on this area should already have the same longitudinal alignment as the track (green) itself. The heading points in the direction of the wagon. The system has automatically defined all dark shaded areas as passable from the data entered and has planned the path only in the passable areas, thus providing the prerequisites for executing complex driving maneuvers. Typical destination points are the rerailing point and the loading station; in rail mode, the wagon. As soon as the ASV is connected to the system, a valid position and an initial heading are automatically determined. Position and direction are indicated by an arrow - similar to navigation systems (see Figure 14). From the system's point of view, there are three main steps in a driving process. Driving in road mode, rerailing and driving in rail mode. Different algorithms are required in all three steps. When driving on the track, for example, no real path planning is required, as the vehicle cannot leave the track and therefore has perfect longitudinal guidance. However, the speed is important in this step, e.g. to carry out a successful coupling process without moving the wagon far beyond the shunting position. The ASV also reacts differently in rail mode than in road mode and also moves a significant load when a wagon is coupled. This changes the braking distances in particular and the speed controller has a significantly longer distance to travel before coming to a standstill.

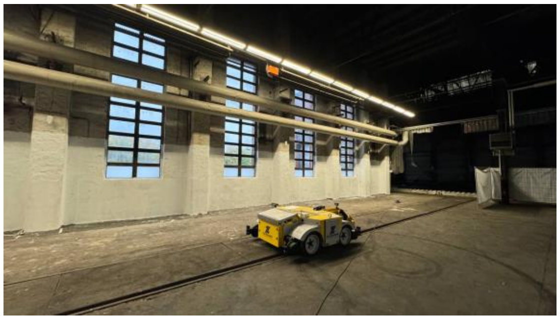


Figure 15. Finished ASV indoor rerailing process

However, it is technically irrelevant on the system side whether a shunting operation is aborted in the middle or not. The system only needs to ensure that the ASV can still safely leave the track. The system can check this via the positions and the map data and then abort the operation if the ASV has taken up a safe position. It is precisely this check that takes place in all steps of a shunting process. If, for example, the selected wagon fails during the operation, e.g. during rerailing, the process would be aborted after rerailing, as there would then be no target position for the next step. In principle, the system always performs the same steps:

- Find the best rerailing position for the operation
- Guide the ASV to the rerailing position
- Rerail (in the correct direction)
- Drive to the wagon and couple
- Check/wait that the path to the next shunting position is clear
- Drive with the wagon to the next shunting position

- Repeat the two previous steps until the destination position of the wagon is reached
- Uncouple
- Drive to the rerailing point
- Rerail

Once the process is complete, the ASV is available for the next process (see Figure 15). If another operation is stored in the system for the vehicle, it is started. The system can operate several ASVs in parallel. When coupling a wagon, in principle only the coupling of the wagon is driven against. This then engages and locks automatically. The vehicle is then steered to the target position with the coupled wagon. Automatic uncoupling then takes place by pneumatically opening the coupling (see Figure 16)

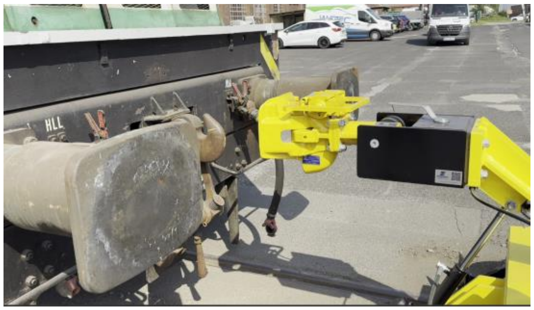


Figure 16. Coupling process

IV. HIGHLY AVAILABLE LOCALIZATION AND COMMUNICATION SYSTEM

For the goal of autonomous shunting, a system concept “integrity of localization and communication” was developed, see Fig. 17. A distinction is made between statistical integrity and systematic integrity. In the case of statistical integrity, the sensor measurement data is evaluated on the basis of the available sensor measurement values; this procedure is used to evaluate the UWB and Bluetooth Low Energy (BLE) data. A quality factor (DQF, Distance Quality Factor) returns an evaluation of the current measured value and can be integrated into the calculation. RAIM (Receiver Autonomous Integrity Monitoring) can be integrated into the satellite position data determined. This procedure can intervene to support the acquisition of measured values and sort out faulty satellite data before a position is calculated. In systematic integrity monitoring, all available information from the systems used is used and merged into a position solution.

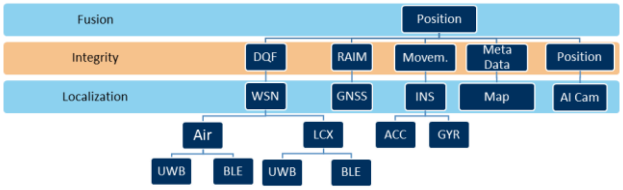


Figure 17. Concept Integrity Localization

If the overall solution deviates significantly from the solutions of the individual systems, as in the case of a threshold-based deviation of the ASV position of 10% for three seconds in the longitudinal and lateral directions, a warning can be sent to the control system of the ASV in order to initiate emergency braking of the freight wagon. In order to ensure the permanent transmission of a position to the central control component of the UAV, the development of a redundant communication infrastructure is proposed. The system concept provides for the transmission of data via the UWB (IEEE802.15.4z) and BLE (IEEE802.15.1) radio standards used for positioning. In the event of interference via the air interface, the use of other transmission media is planned. In the approach investigated, leaky waveguides and slitted coaxial antennas were used.

Figure 18 shows the basic use of LCX for locating and transmitting information.

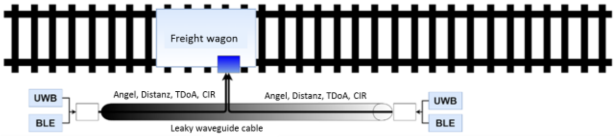


Figure 18. Basic use of leaky waveguide cable for localization and information transmission

Initial investigations with leaky waveguides (LCX) as a communication medium under laboratory and test side conditions have shown that this medium is suitable as a redundancy system for the transmission of communication data (Figures 19 and 20).

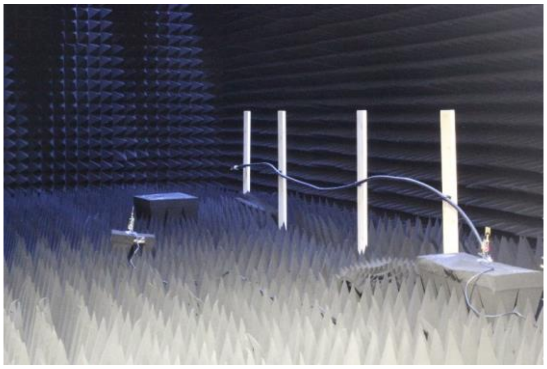


Figure 19. Laboratory tests on the use of leaky waveguide cables for localization and information transmission

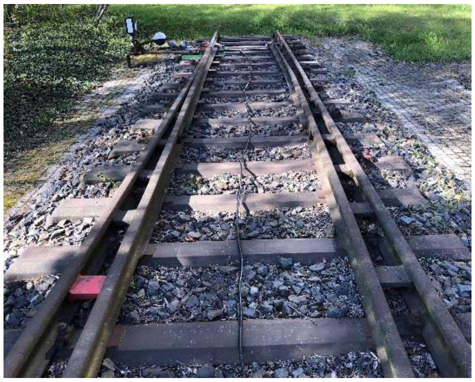


Figure 20. Real-world test investigations into the use of leaky waveguide cables for localization and information transmission in the rail sector

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-241-8

The methods developed for multisensor-based integrity monitoring were combined on the basis of the individual algorithms tested in the conceptual phase on the basis of synthetic data. In this context, the radio channel in particular was implemented using software with regard to a “control loop” for integrity monitoring (see Figure 21) based on the above-mentioned radio parameters.

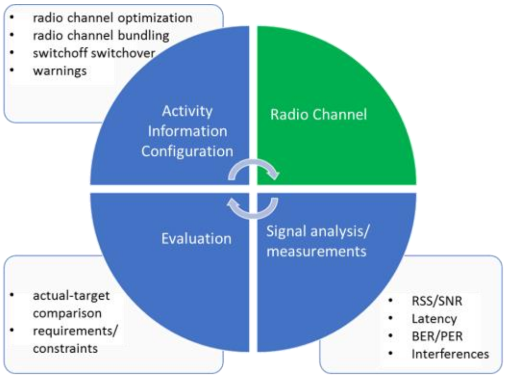


Figure 21. Representation of the “control loop” of integrity monitoring for radio communication [4]

V. Laboratory Based and field tests in railroad research depot

The following approach has proven itself scientifically and practically as a way of finding objectives. First, computer-aided simulative investigations with modeling and simulation. Secondly, laboratory-based tests of selected promising components (e.g. UWB radio sensor network, LCX cable). Thirdly, after systematic laboratory tests, the transfer to the real test environment, in this case the Railroad Research Depot takes place. In the following, individual results that are considered important are shown. Figure 22 shows a simulative evaluation of the distance estimates of the UWB radio system actually used on the basis of a multipath-free and a multipath-strong 3D modeled railroad operating area. It is clearly visible that a good position estimate can be derived despite the strong multipath propagation [7].

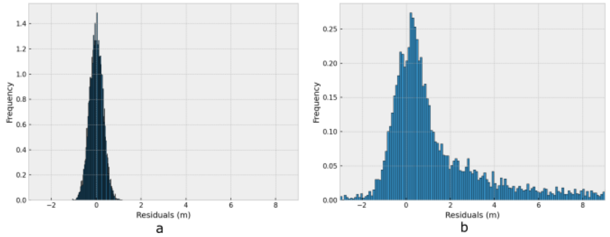


Figure 22. UWB distance estimates. a) free space propagation, b) strong multipath influence

Furthermore, the possible use of an LCX cable was also investigated, which was initially evaluated in the laboratory (multipath-free room) under best-case conditions (see Figure 19). The results are shown below in Figure 23.

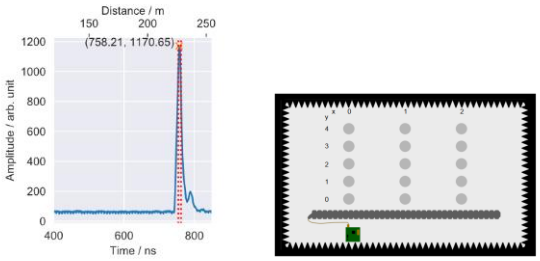


Figure 23. Measurement result of the channel impulse response at point 2-2 (marked in red) for determining the position along an LCX in a multipath-free room

VI. Conclusion and further Work

In In this work, we have shown the implementation of an automatic shunting system with an ASV and dealt with the most important system components for the realization, in particular with the implemented system concepts and components for localization and communication based on radio sensor networks and leaky waveguides cables. It can be stated that the complexity of autonomous navigation of ASVs in densely occupied marshalling yards was underestimated. For this reason, ongoing research and development work is focusing on evaluating and fine-tuning the system components for control, communication and localization. This will focus in particular on improving the object recognition of the rails to initialize the rerailing process using AI-supported image processing, as this has so far proven difficult for different marshalling yard environments.

Acknowledgment

This work was partially funded by the Federal Ministry of Education and Research (BMBF, FKZ: 16ME0054, AZubiG). The work on the research project was carried out jointly within various chairs of Technische Universität Dresden (ITVS, FZM, EB) and by the project partners Zigpos GmbH, Bahnkonzept GmbH Deutschland, Dresden Elektronik Ingenieurtechnik GmbH and G. Zwiehoff GmbH Further investigations are ongoing.

References

[1] A. N. Mackay (Hrsg.): A History of North Eastern Railway Signalling. North Eastern Railway Association, ISBN 978-1-873513-99-6, 2016

[2] A. Michler et al: Conceptualization of Communication and Localization Components for Automated Shunting, MT-ITS, 2023

[3] R. C. Ramírez, I. Adin, J. Goya, U. Alvarado, A. Brazalez, and J. Mendizabal, “Freight train in the age of self-driving vehicles - a taxonomy review,” IEEE Access, vol. 10, pp. 9750–9762, 2022. [Online]. available: https://doi.org/10.1109/ACCESS.2022.3144602

[4] “AZubiG - Electronic system for automatic train formation for freight cars, https://tud.link/wey1”

[5] A. Auer, S. Feese, S. Lockwood, and B. A. Hamilton, “History of intelligent transportation systems.” United States. Department of Transportation. Intelligent Transportation, Tech. Rep., 2016. [Online]. Available: http://www.itsga.org/wp-content/uploads/2016/08/ITS-JPO-History-of-ITS.pdf

[6] P. Papadimitratos, A. L. Fortelle, K. Evenssen, R. Brignolo, “Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation,” IEEE Communications Magazine, vol. 47

[7] P. Schwarzbach, A. Michler, and O. Michler, “Tight integration of GNSS and WSN ranging based on spatial map data enhancing localization in urban environments,” in 2020 ICL-GNSS. IEEE, Jun. 2020. [Online]. Available: https://doi.org/10.1109/icl-gnss49876.2020.9115519

[8] Hoffmann, Gabriel M., Claire J. Tomlin, Michael, Montemerlo, and Sebastian Thrun. "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing." American Control, Conference. 2007, doi:10.1109/ACC.2007.4282788

[9] Akhimullah, Muhammad & Hudha, Khisbullah & Kadir, Zulkiffli & Syed Mohd Dardin, Syed Mohd Fairuz & Amer, Noor. (2020). Development of path tracking control of a tracked vehicle for an unmanned ground vehicle. International Journal of Advanced Mechatronic Systems. 8. 136. DOI: 10.1504/IJAMECHS.2020.112628