ICONS 2025

The Twentieth International Conference on Systems

ISBN: 978-1-68558-278-4

May 18 - 22, 2025

Nice, France

ICONS 2025 Editors

Przemyslaw Pochec, University of New Brunswick, Canada

ICONS 2025

Forward

The Twentieth International Conference on Systems (ICONS 2025), held between May 18-22, 2025 in Nice, France, continued a series of events covering a broad spectrum of topics, including fundamentals on designing, implementing, testing, validating and maintaining various kinds of software and hardware systems.

In the last years, new system concepts have been promoted and partially embedded in new deployments. Anticipative systems, autonomic and autonomous systems, self-adapting systems, or on-demand systems are systems exposing advanced features. These features demand special requirements specification mechanisms, advanced behavioral design patterns, special interaction protocols, and flexible implementation platforms. Additionally, they require new monitoring and management paradigms, as self-protection, self-diagnosing, self-maintenance become core design features.

The design of application-oriented systems is driven by application-specific requirements that have a very large spectrum. Despite the adoption of uniform frameworks and system design methodologies supported by appropriate models and system specification languages, the deployment of application-oriented systems raises critical problems. Specific requirements in terms of scalability, real-time, security, performance, accuracy, distribution, and user interaction drive the design decisions and implementations.

This leads to the need for gathering application-specific knowledge and develop particular design and implementation skills that can be reused in developing similar systems.

Validation and verification of safety requirements for complex systems containing hardware, software and human subsystems must be considered from early design phases. There is a need for rigorous analysis on the role of people and process causing hazards within safety-related systems; however, these claims are often made without a rigorous analysis of the human factors involved. Accurate identification and implementation of safety requirements for all elements of a system, including people and procedures become crucial in complex and critical systems, especially in safety-related projects from the civil aviation, defense health, and transport sectors.

Fundamentals on safety-related systems concern both positive (desired properties) and negative (undesired properties) aspects. Safety requirements are expressed at the individual equipment level and at the operational-environment level. However, ambiguity in safety requirements may lead to reliable unsafe systems. Additionally, the distribution of safety requirements between people and machines makes difficult automated proofs of system safety. This is somehow obscured by the difficulty of applying formal techniques (usually used for equipment-related safety requirements) to derivation and satisfaction of human-related safety requirements (usually, human factors techniques are used).

We welcomed academic, research and industry contributions. The conference had the following tracks:

- Complex and specialized systems
- Embedded systems and applications/services
- Computer vision and computer graphics
- Application-oriented systems

We take here the opportunity to warmly thank all the members of the ICONS 2025 technical program committee, as well as all the 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 effort to contribute to ICONS 2025. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the ICONS 2025 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that ICONS 2025 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of systems. We also hope that Nice provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

ICONS 2025 Chairs

ICONS Steering Committee

David Inkermann, Technische Universität Clausthal, Institute of Mechanical Engineering, Germany
Christoph Knieke, Technische Universität Clausthal, Institute for Software and Systems Engineering, Germany
Mo Mansouri, Stevens Institute of Technology, USA/ University of South-Eastern Norway, Norway
Mark Austin, University of Maryland at College Park, USA

ICONS Publicity Chairs

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

ICONS 2025

Committee

ICONS Steering Committee

David Inkermann, Technische Universität Clausthal, Institute of Mechanical Engineering, Germany
Christoph Knieke, Technische Universität Clausthal, Institute for Software and Systems Engineering, Germany
Mo Mansouri, Stevens Institute of Technology, USA/ University of South-Eastern Norway, Norway
Mark Austin, University of Maryland at College Park, USA

ICONS 2025 Publicity Chairs

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

ICONS 2025 Technical Program Committee

Qammer H. Abbasi, University of Glasgow, Scotland, UK
Mohamed A. Abd El Ghany, German University in Cairo, Egypt
Witold Abramowicz, Poznan University of Economics, Poland
Afaq Ahmad, Sultan Qaboos University, Oman
Abdelouhab Aitouche, YNCREA/HEI | University of Lille, France
Ali Al-Humairi, German University of Technology (GUtech), Oman
Walid Al-Hussaibi**,** Southern Technical University (STU), Iraq
Saloua Bel Hadj Ali, University of Tunis-El Manar / University of Gabès, Tunisia
Fady Alnajjar, United Arab Emirates University, UAE
Yasmine Arafa, University of Greenwich - School of Computing and Mathematical Sciences, UK
Marc Austin, University Of Maryland, USA
Muhammed Ali Aydin, Istanbul University-Cerrahpasa, Turkey
Lubomir Bakule, Institute of Information Theory and Automation, Czech Republic
Walter Balzano, Università degli Studi di Napoli Federico II, Italy
Kamel Barkaoui, Cedric - Cnam, Paris, France
Janibul Bashir, National Institute of Technology, Srinagar, India
Héla Belkhiria, Higher National Engineering School of Tunis - University of Tunis / Polytechnic School of Tunisia, Tunisia
Abir Ben Ali, National School of Computer Sciences (ENSI), Tunisia
Alejandro J. Bianchi, LIVEWARE S.A. / Universidad Catolica Argentina, Argentina
Razvan Bocu, Transilvania University of Brasov, Romania
Birthe Boehm, Siemens AG, Germany
Sander Bohte, Machine Learning group - CWI, Amsterdam, The Netherlands
Marilisa Botte, Federico II University of Naples, Italy

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

System Engineering Methods for Reliable Electrical Power Train Design for All Electric Aircraft

Jonas Franzki
Institute for Electrical Machines, Traction and Drives
Technische Universität Braunschweig
Braunschweig, Germany
e-mail: jonas.franzki@tu-braunschweig.de

Anna Nanzig
Institute :metabolon
Technische Hochschule Köln
Köln, Germany
e-mail: anna.nanzig@th-koeln.de

Markus Henke
Institute for Electrical Machines, Traction and Drives
Technische Universität Braunschweig
Braunschweig, Germany
e-mail: m.henke@tu-braunschweig.de

Anna-Lena Menn
Department of Engineering and Communication
Hochschule Bonn-Rhein-Sieg
Sankt Augustin, Germany
e-mail: anna-lena.menn@h-brs.de

***Abstract*—Reliability management, including hazard and risk analysis, is essential for the product development of All Electric Aircraft (AEA) systems to ensure the safety of people and the robustness of the system. In this study, a Model-Based System Engineering (MBSE) approach is proposed that integrates reliability and safety analysis into an accessible system model that improves collaboration among stakeholders, especially those with framed technical involvement. Using a bond graph-based method in Mathworks' System Composer, the interfaces and interactions of the components are modelled and the consequences of possible failures are shown. Established methods, such as Failure Mode and Effects Analysis (FMEA), Fault Tree Analysis (FTA), and Reliability Block Diagrams (RBD) are compared. All of these safety analyzes are compatible with the proposed MBSE approach. The aim is to outline an approach to analyze system reliability and safety rather than cataloguing every possible failure of an electric drive system. This method provides structured, visual means of analyzing complex failures that go beyond traditional, spreadsheet-based documentation, allowing for better alignment between safety and design.**

Keywords-MBSE, reliability, all electrical aircraft, system model, safety analysis, electrical powertrain

I. Introduction

Reliability management and Hazard And Risk Analysis (HARA) are one of the most important aspects in the product development process of all electrical aircraft systems. Therefore, the focus of research and development is on reliability improvement of electrification components as, e.g., presented in [1] and on the conscientiously conducted HARA to avoid injury and death of people. The aim of HARA is to avoid systematic errors in the product development process and to make the system robust against errors, requirements for the technical system are derived from the HARA. In addition, HARA is mandatory according to CS-23 and 25 [2][3].

This complex part of product development is to be simplified through the use of a complexity-reducing method and simple MBSE language, so that even stakeholders with little involvement have easy access to the technical system. Our method includes a physical approach and is based on bond graph theory [4]. SysML language is consciously not used, but System Composer, block-oriented language, because of the intuitive use and linkability to multiphysical 1D simulation (Simscape). Nevertheless, the building of the system model requires a deep understanding of the technical system.

The core of our method is the interface visibility to show the consequences of failures of the electric porpulsion system. At the beginning of this development, links between faults are shown in a systemic, model-based way, thereby promoting a better overview and collaboration between safety and design engineers. As a rule, HARA results are recorded in tables that do not provide any information about the possible relationships between faults. Our system model enables the direct derivation of HARA [5].

To discuss the method and the proposed procedure, an all electric aircraft propulsion system is chosen, in particular the drive unit consisting of: electric motor, gearbox, and propeller.

In electric aviation, performance, mass, and safety are the three most important development aspects, so it is particularly important to understand safety and technical requirements as a common construct from the outset. Finally, it is important to note that the focus is on showing and discussing a method; it is not the aim to show all possible failures of an electrical propulsion system as displayed in Figure 1.

The remaining content is structured as follows: In Section II the applicable standards as well as existing methods and their respective challenges are presented as a background for the system engineering method proposed in Section III. In Section IV a functional safety analysis is performed on an electric drive unit as a basis before applying the proposed system engineering method in Section V. Finally, conclusions are drawn in Section VI.

II. Standards, Methods and Challenges

In the context of aircraft, many standards define the requirements and procedures to follow. The most important ones for the design of electrical drives will be presented in this section

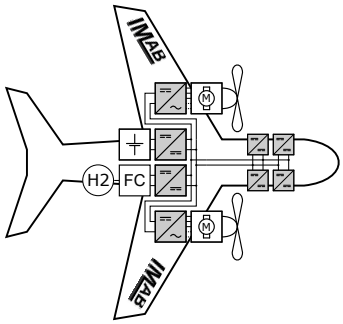


Figure 1. On-board power supply of an AEA concept

alongside methods and challenges in the evaluation of reliability of new electrical drives for aircrafts. Many procedures of realiability and safety analysis are already known: FMEA, FTA, HARA, RBD and PoF (Physics of Failure). Some of these methods are presented and discussed in this section. The method developed is intended to support existing methods and to improve and clarify their application and results.

A. Standards

The Certification Standards (CS) CS-23 [2] and CS-25 [3] define most important requirements for the certification of small and large aircrafts, respectively. The manufacturer must demonstrate compliance to these standards for the aircraft to be granted type certification. This involves requirements applicable to components such as electrical drives.

CS-23 [2] applies to small airplanes (e.g., commuter, private, and training aircrafts). Design and performance criteria are generally less strict than for large airplanes and apply to simpler systems with less redundancy since the operational environment is considered less demanding (fewer passengers, simpler flight profiles). Although safety is still a priority, the measures may be more straightforward and potential failure mode analysis and their mitigation less extensive. Thus, testing is less costly with fewer tests required compared to CS-25 and simpler documentation.

The design should ensure that there are means to give immediate warning to the flight crew in case of a failure of any generator or propulsor, and each must have an overvoltage protection system to prevent damage to the electrical system or equipment supplied by it in case of an overvoltage condition. Furthermore, each electrical system must be free from hazards in its operation and effects on other parts of the aircraft, ensuring safety and reliability. [2]

In contrast, CS-25 involves more complex systems with high redundancy requirements with great emphasis on redundancy, fault tolerance, and fail-safe design to ensure safety of more passengers and demanding operations. Hence, more rigorous testing, validation, and documentation processes including extensive Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) are required to minimize risk of failure. [3].

TABLE I. FUNCTIONAL RELIABILITY METHODS

	HAZOP	FMEA	FMEDA	FTA	RBD	Markov
in-/de-ductive	in	in/de	in	de	de	de
qualitative quantitative	qual	qual	quan	quan	quan	quan
depth of detail	rough	variable	detail	detail	rough	rough
IEC Standard	61882	60812	61508	61025	61078	61165

The complex nature of the compliance process highly motivates the development and use of guiding, supporting, structuring, and visualizing tools to facilitate the process.

B. Functional Safety Methods

There are many methods to investigate the functional safety of a system. They can be divided into inductive and deductive methods. Deductive methods work top-down, they start from known causes to find unknown effects, whereas inductive methods work bottom-up, starting with known effects to seek their unknown causes. Additionally, they can be split into qualitative and quantitative methods: qualitative methods look for the robustness and fault tolerance of architectures, while quantitative methods look into the failure rate, sum of parts and unavailability[6].

Common methods for the analysis of functional safety are: HAZard and OPerability study (HAZOP), Failure Modes and Effects Analysis (FMEA), Failure Modes, Effects and Diagnostic Analysis (FMEDA), Fault Tree Analysis (FTA), Reliability Block Diagram (RBD), Markov, and many more (see Tab. I).

As previously presented, FMEA and FTA are already integral parts in the certification process. They are well compatible with each other. While FMEA offers mainly a bottom-up approach (inductive), FTA can be used for top-down (deductive). Both require an initial Hazard and Risk Assessment (HARA). Thus, HARA can be used to perform an initial analysis and then either a FTA can be conducted or the failure modes can be assessed in their Severity (S), probability of Occurence (O) and Detection (D), the product of which results in a Risk Preference Number (RPN) for a FMEA. Thus, HARA and FMEA allow for a variable depth of detail in the analysis and are, hence, easy-access tools.

The combination of HARA, FMEA, and FTA is a compelling and often used tool chain in traction applications, also as manifested in the ISO 26262 automotive standard for functional safety ISO 26262 [7]. For this reason, the present study will conduct a combination of HARA and FMEA for an electric drive train to achieve a basis example on which a system reliability model will be created, which allows for a comprehensible and visually appealing depiction of system engineering approaches on reliability.

C. Challenges

HARA and FMEA are table-based tools with often extensive lists and little visual appeal, making it hard for less technically

adept stakeholders. Model-Based Systems Engineering gives the possibility to visualize the system topology from the beginning. System models using a simple modeling language could close this gap. Model-Based Systems Engineering (MBSE) is a methodology that focuses on using models as the primary means of information exchange and system design throughout the engineering lifecycle. Instead of relying solely on traditional documents, MBSE emphasizes graphical and digital representations to capture, analyze, and communicate system requirements, design, analysis, and validation. This approach improves consistency, traceability, and collaboration between stakeholders. Key advantages include reducing errors, enabling early detection of design issues, and facilitating integration across disciplines.

III. Proposed System Engineering, Method and Procedure

A. Structure MBSE method

The proposed method leads to an advanced system model that enables the mitigation of a hazard and risk analysis. To achieve the aim, the method is divided into four parts building on each other, the method is illustrated in Figure 2. The first part of "abstract modeling" is mandatory if a new product is developed, which did not exist before. The result of this part is the knowledge of the physical elementary functions and the possible solutions to convert energy from one form to another, like electrical to mechanical. If the system under investigation is already known, it can be skipped and initialized with system consideration with "basic modeling". In advance, it is mandatory to set up the framework that includes the nomenclature and the specification of the modeling language. This is important to create a common understanding of the description of the technical system. The next part is mainly concerned with the superordinate representation of the system to be analyzed, i.e., to clarify which main components make up the system and which components are connected to each other via which physical domain. The result of "basis modeling" is a basic system model that shows the energy and signal flow structure of all components. It represents only one level and shows which energy flow represents the input and output of the respective component. This one-level system model is the basis for the next part, which leads to the final advanced system model. This part is divided into three subparts: function analysis, risk analysis, and final risk mitigation.

Function analysis begins with a decomposition of the components, the components are disintegrated into subcomponents, and more levels are created. The motivation of this decomposition is to get to know the causes and hazards. Therefore, functions and possible malfunctions of the subcomponents are determined, and thus the system is analyzed by possible loss of function. The loss of function is declared for causes and hazards derived from this. Malfunctions are always a disturbance in the energy transmission or power transmission in the sense of the bond graph theory, divided into flow and effort variables. This theory also empowers the multiphysical system view, because each component is connected to several

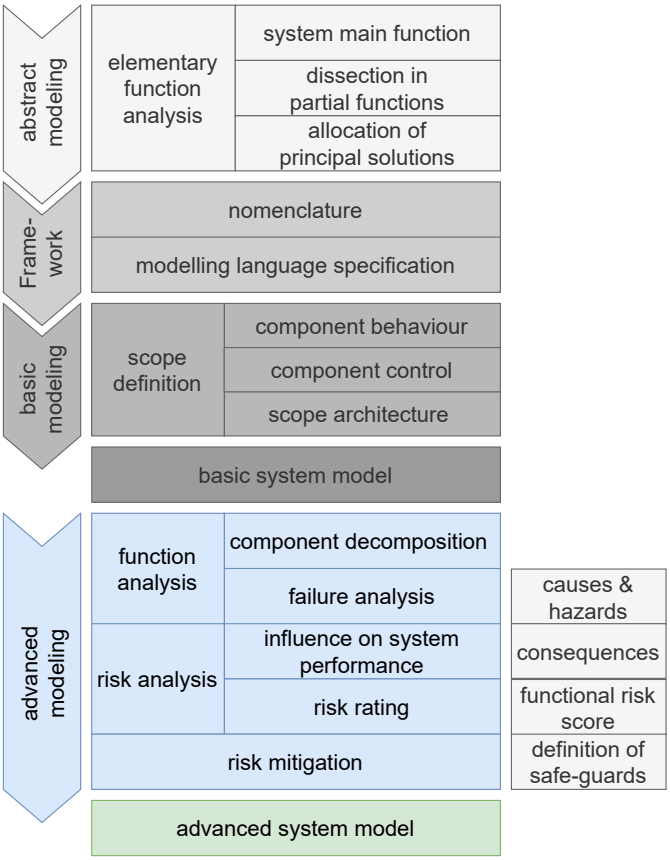


Figure 2. Schematic diagram of the method

physical domains. So, the result of the first subpart are causes and hazards, while hazards bundle several causes. To start risk analysis, the influence on the system performance is the next step. The result is called consequences and describes the impact on system behavior. Risk anlaysis ends with a risk rating according to severity, exposure, and controllabity, resulting in defining a functional risk score. This rating is still provided by human intelligence. The last step of the method is to mitigate the risks and define safe guards. The final advanced system model completely replaces any table. Technical requirements are derived from the safe guards and their effect can be proofed by 1D multiphysical simulation.

Thus, the proposed method offers a strategic and clearly visualized way to show compliance of newly developed systems with CS-23 / CS-25 or automotive standards. It maps failure scenarios, facilitates finding interacting failures, and includes mitigation strategies. A comparison to FMEA and FTA is shown in Table II.

Section V will show a detailed application of the method.

IV. Applied Functional Safety Analysis

As outlined in the previous section, the system model is based on the functional safety analysis of a defined subsystem. This is generically performed here on the electrical machine drive to demonstrate the applicability and merits of the proposed method. Usually HARA is performed on complete

TABLE II. COMPARISON OF PROPOSED SYSTEMS ENGINEERING METHOD (SEM) TO FMEA AND FTA

Criteria	FMEA	FTA	SEM
Visualisation	No	Yes	Yes
Link between interacting components and failures	No	No	Yes
System wide evaluation of failure consequences	Yes	No	Yes
Failure mitigation and safeguards	Yes	No	Yes

TABLE III. SCORING SYSTEM

Criteria	Score	Definition
Severity	0	No function reduction
	1	Moderate reduction in degree of performance
	2	Severe harm to drive unit
	3	Loss of full drive unit
Exposure	0	Incredible
	1	Very low probability
	2	Low probability
	3	Medium probability
	4	High probability
Control-lability	0	Controllable in general
	1	Simply controllable
	2	Normally controllable
	3	Difficult to control or uncontrollable

systems, however, it can also be utilized for subsystems with appropriate adaptation as later expanded.

Exemplary hazards to the electrical machine are overheating and winding failure. The former can be induced by a variety of causes like failure of coolant pump, coolant leaks, or other component failures (reservoir, filter). Winding failure could be caused by insulation aging or short circuit after overload operation. The hazard of power loss due to magnet demagnetization can be caused by a number of causes as well, like overheating, manufacturing error, or overcurrents.

All causes can be sorted according to the failing system (e.g., cooling or motor) and physical domain (e.g., hydraulic, thermal, mechanical, electrical), which can later be used for graphical highlighting. Furthermore, all cause-hazard combinations must be scored according to their severity, exposure, and controllability according to HARA. The scales according to ISO 26262 can be utilized while translating "harm to and loss of life" to "harm to and loss of the drive unit", as can be seen in Table III.

An examplary score for PM mechanical damage and demagnetization can be found in Table III.

Safeguards can then be defined based on these hazards, causes, etc. The SIL score gives an indication on the scope of measures to be taken. That is, the rating "QM" indicates quality management is sufficient, whereas A, B and C-levels require increasing consideration, respectively. These HARA results are used to augment the basic system model with reliability aspects as described in the following section.

V. SYSTEM MODELING APPLICATION

The concrete application of the developed method will be presented in this section. The Mathworks System Composer is used as a tool for creating the system model. The creation of an advanced system model will be carried out using the example of the motor of an AEA.

The first step is the creation of the basic system model, which is skipped at this point and started directly with the creation of the advanced system model. This is an important step in order to be able to perform a risk analysis based on a system model. Figure 3 therefore already shows the completed basic system model of the left wing of an AEA. The colors of the components are chosen to indicate their respective domains. Electrical components are shown in blue, while mechanical components are labeled in green.

This section describes the development of the advance system model of the motor. The term ‘system’ should be understood to mean that each subcomponent consists of further, more in-depth components that together form the overall model.

The first step is to decompose the motor into its main components: rotor and stator. These two components can in turn be decomposed into further subcomponents. Figure 4 shows a detailed decomposition of the rotor, which consists of the magnet system, the shaft bearing system, the rotor laminations and the rotor sleeve.

Analyzing possible faults, also known as causes, in the individual subcomponents inevitably identifies potential hazards that can result from these malfunctions. These hazards are shown as red blocks in the system model, as different faults can lead to the same hazard. For example, both bearing friction and loss of magnetization can lead to heating of the rotor. Risks can be derived from identified hazards that are assessed directly in the system using a risk score. This can be based on previously performed (or known) risk analyzes.

The identified risks leave the rotor component as the output. Logically, action must now be taken to manage these risks. Figure 5 shows how the various risks leave the PMSM. Basically, solutions are identified here to minimize the risks. For example, the risk of ‘dysfunction of magnets’ can be reduced through a more robust design, higher safety margins, or improved quality management.

TABLE IV. SCORING EXAMPLE

ID	Hazard	Causes	System	Category	Severity	Exposure	Controllability	SIL-Score
1	Demagnetization	overheating	motor	thermal	2	1	1	QM
2	Demagnetization	manufacturing	motor	mechanical	1	1	1	QM
3	Demagnetization	overcurrent	motor	electrical	2	2	2	QM
4	PM mechanical damage	incorrect sleeve	motor	mechanical	3	2	3	B
5	PM mechanical damage	incorrect operation	motor	mechanical	3	1	1	QM

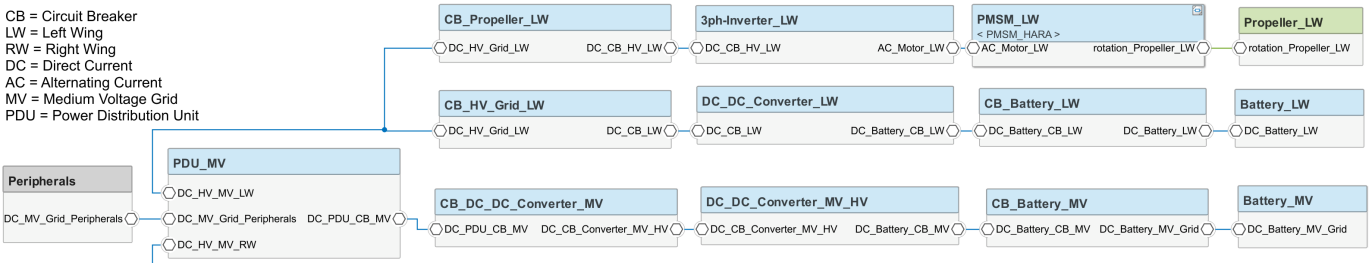


Figure 3. Basic system model of the left wing of AEA.

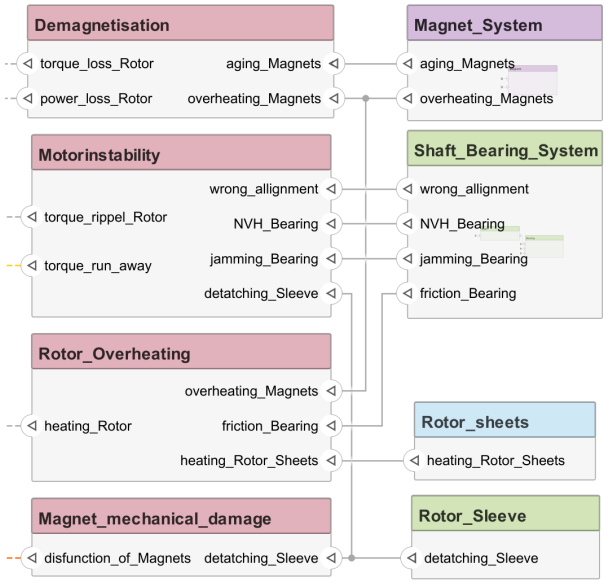


Figure 4. Decomposition of the rotor

The diagram also clearly shows that different risk scores are assigned to the various hazards. Particularly critical hazards are marked in red, less critical ones in yellow, and hardly critical ones in gray according to their SIL score.

VI. Conclusion

This paper presents a low-threshold MBSE method that integrates functional safety considerations into the system modeling process. Using this approach, the system model facilitates the identification and derivation of hazards and their causes in a structured and traceable manner. On the other hand, it must be said that setting up the system model for the first time requires expert knowledge in the respective technical field of application. The physical complexity is very high. Causes are always an impairment in energy transport according to bond graph theory. Moreover, the system model enables the mapping of functional safety aspects in a way that promotes better understanding, even among individuals with limited prior experience or involvement in safety-related topics. This aspect enhances cross-disciplinary collaboration and improves communication within teams. However, it is important to note that the creation of a comprehensive and advanced system model demands a deep understanding of the underlying technical system. This prerequisite highlights the need for skilled practitioners during the initial model development phase. The system model can be reused in a subsequent FMEA later in the product development process. Thus, the proposed MBSE method not only supports the integration of functional safety considerations but also contributes to the efficiency and effectiveness of safety engineering practices.The connection between the detection of errors and the bond graph theory will be addressed in greater depth in future research projects. It is planned to take a closer look at the mathematical underpinning with the help of the bond graph theory of the method presented here.

References

[1] R. Keilmann, L. Radomsky, D. Ferch, and R. Mallwitz, “Study of inverter topologies for electrified aircraft propulsion systems based on cyclic loading induced bond wire fatigue,” in *2024 Energy Conversion Congress and Expo Europe (ECCE Europe)*, 2024, pp. 1–8. DOI: 10.1109/ECCEEurope62508.2024.10752026.

[2] EASA, *Easy Access Rules for Normal-Category Aeroplanes (CS-23) - Amendment 6 (AMC/GM 4) | EASA*, en, https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-normal-category-0, retrieved: April 2025.

[3] EASA, *Easy Access Rules for Large Aeroplanes (CS-25) - Revision from January 2023 | EASA*, en, https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-large-aeroplanes-cs-25, retrieved: April 2025.

[4] A.-L. Menn and A. Nanzig, “Komplexitätsreduktion von Methoden im MBSE,” in *Tag des Systems Engineering*, Würzburg: GfSE Verlag, Nov. 2023, S.100–107, ISBN: 978-3-910649-00-2.

[5] Y. Jiang *et al.*, “MBSE-based functional hazard assessment of civil aircraft braking system,” in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec. 2020, pp. 460–464. DOI: 10.1109/ICMCCE51767.2020.00107.

[6] L. Fendrich and W. Fengler, Eds., *Handbuch Eisenbahninfrastruktur*, de. Berlin, Heidelberg: Springer, 2013, ISBN: 978-3-642-30020-2 978-3-642-30021-9. DOI: 10.1007/978-3-642-30021-9.

[7] N. Adler, *Modellbasierte Entwicklung funktional sicherer Hardware nach ISO 26262* (Steinbuch Series on Advances in Information Technology / Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung). KIT Scientific Publishing, 2015.

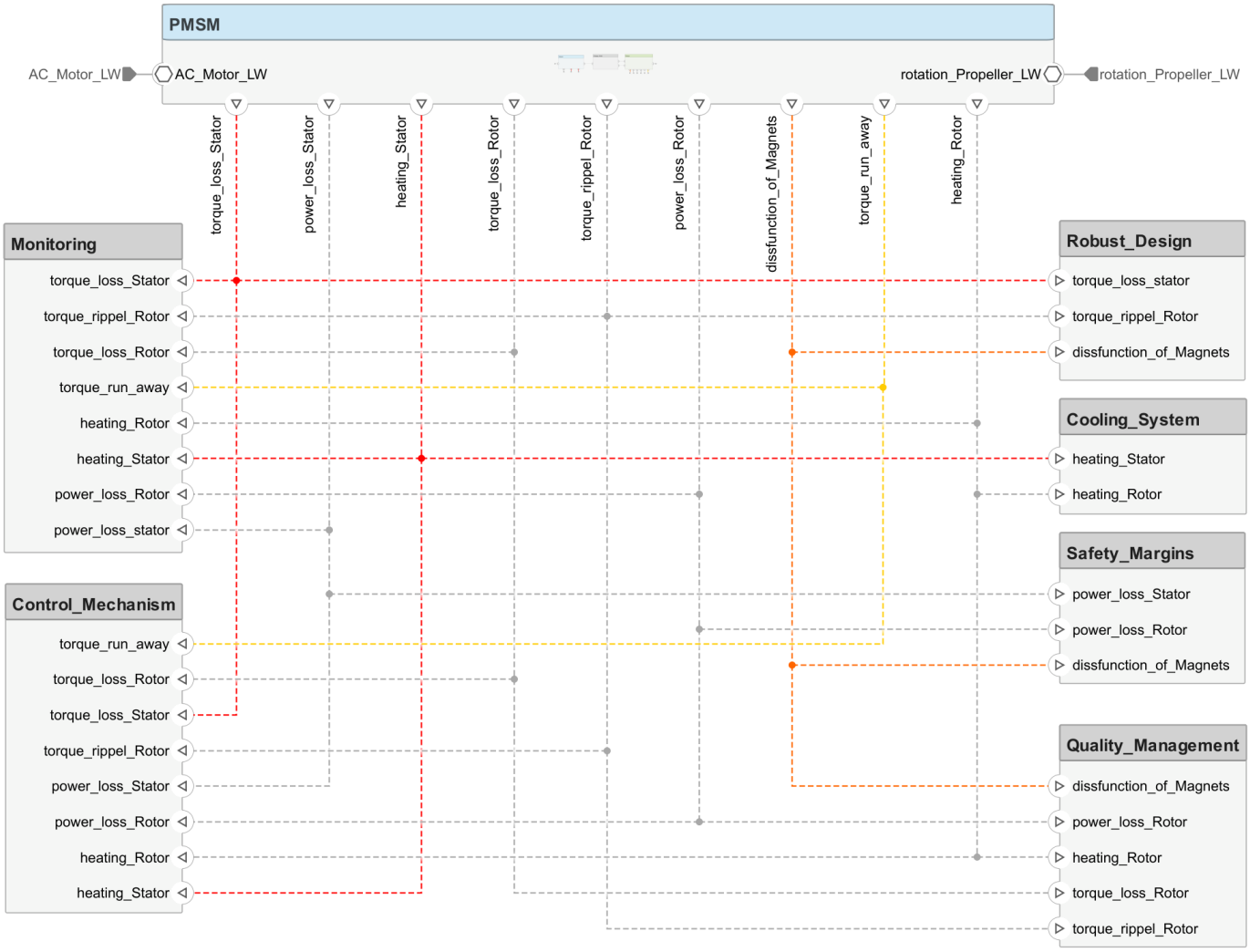


Figure 5. Definition of safe-guards for the PMSM

A Methodological Approach to Sustainable Product Development by Combining Life Cycle Assessment and Systems Engineering

Karolina Wirtz-Dürlich[a], Simon Adams[a], Anna-Lena Menn[a]*

[*a] *University of Applied Science Bonn-Rhein-Sieg*
Grantham Allee, Sankt Augustin 53757, Germany
**email: anna-lena.menn@h-brs.de*

***Abstract*— Product requirements and the reduction of its ecological footprint are often in conflict with each other. However, sustainable development is seen to be important as shown by international agreements. Environmental improvement of a product, e.g., with material substitution, might change a product-significant physical parameter. Therefore, a methodology is conducted which combines the result of Life Cycle Assessment with modelling and simulation concepts to design an environmentally friendly and physically functional product. By applying life cycle impact data to different system model configurations, their results can be compared to show a more sustainable product design, mitigating global warming for example. This is achieved by linking Life Cycle Assessment to the topology of a system in a five-step method. The conducted five-step method consists of Life Cycle Assessment, hotspots, data sorting, system topology and solutions. The developed method enables the identification of materials and components with high environmental impact already in early design stages, even before the physical product exists. This allows targeted decisions for sustainable design by evaluating environmental performance alongside functional requirements at a conceptual level.**

Keywords - Modelling and Simulation; System Enginnering; Life Cycle-Assessment ; Hotspots; Stereotypes

I. INTRODUCTION

The European Commission estimates that over 80 % of all product-related environmental impacts are determined during the Product Development (PD) phase of a product [1]. To ensure that the needs of future generations are met as well as the needs of the present, a shift in current product design towards sustainable development is essential. This aligns with the principles outlined in the Brundtland Report in 1987 [2]. The decreasing of the environmental impact of PD is also relevant to the EU goal of becoming carbon neutral by 2050 [3]. Sustainability has become a requirement for companies to make conscious decisions for the design and production phase regarding the environmental impact of their products [4], [5].

In order to achieve that goal, da Luz et al. developed a five-point assessment scale. Here, the numbers say how significant the improvement of each impact category of the Life Cycle Assessment (LCA) and the Product Development phase is [4].

Besides da Luz et al. other studies have also shown that the combination of LCA and PD is one of the most powerful tools for sustainable product design. The results of the LCA of a product show which parts of the product might need improvement even before it is on the market [6].

However, an environmental improvement of a product, for example with material substitution, might change a physical parameter which is significant for the product. A new methodology is needed which combines the results of the LCA with modelling and simulation concepts to get a combination of an environmentally friendly and physical functional product. This methodology could function as a guidance for product developers.

Model-based systems engineering (MBSE) provides a structured approach to managing complex systems by utilizing models instead of traditional document-based methods. It supports the entire product lifecycle, from conception to decommissioning, ensuring greater efficiency and consistency in the development process. By integrating MBSE into Product Development, complex interdependencies can be systematically analyzed, facilitating better communication and decision-making. This approach is particularly relevant for sustainable product design, as it enables the structured evaluation of different design alternatives, including those aimed at reducing a product's carbon footprint. To achieve this, clear frameworks are essential to provide transparency and to ensure that large amounts of data can be effectively analyzed and utilized [7].

In section two of this paper, Life Cycle Assessment and Sustainable Product Design are defined as well a state-of-the-art of the current development. In section three, the conducted method is presented in five consecutive steps. Afterwards, the method is applied to an exemplary design workflow of an electric motor. The benefits and limitations of this method are pointed out, as well as an outlook for future research on the topic.

II. LITERATURE

The state of the art of integrating resource efficiency into product development is described here.

A. Life Cycle Assessment

Life Cycle Assessment allows to analyze the environmental and human health impact of a product from production until recycling for example. Thereby, LCA shall help to provide opportunities to improve the environmental performance at different stages of the product. Moreover, it should help to select relevant indicators of environmental parameters and to find hotspots. All in all, an LCA of a product helps to give a whole overview over the environmental impacts of a product. Furthermore, it helps to find specific components which are affecting impact categories a lot, e.g., global warming potential (GWP).

The methodology of the LCA is set in the DIN EN ISO standards (14040 / 14044) and conducts four stages which can also be found in Figure 1; goal and scope, inventory analysis (LCI), impact assessment (LCIA) and interpretation.

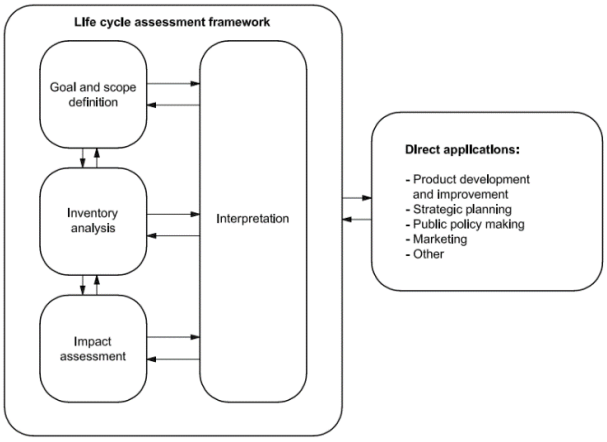


Fig. 1: Stages of an LCA [14], [15].

The first stage of an LCA is the definition of the goal and scope, as shown in Figure 1. A functional unit must be defined first. Also, the lifespan, assumptions, as well as the system boundary (e.g., "cradle to gate", "cradle to grave" ...) need to be documented. In the inventory analysis inputs and outputs of the material flows of the product are analyzed. During the impact assessment, different impact categories are evaluated by assigning equivalent emission factors to the inventory flows. Impact categories can be global warming potential or the freshwater ecotoxicity potential for example [8]. After the first iteration, the results are interpreted, and a sensitivity analysis is conducted. A sensitivity analysis helps to make a statement about the uncertainty of the assumptions supposed earlier.

B. Sustainable Product Design

According to Chang et al. Product Development can be divided into four different stages: concept design, part design, process design, and decision making.

Design stages of interest would be part design and process design because parts of these stages are material selection, waste minimization as well as identification of alternatives [9].

As mentioned by da Luz et al. [4] the PD process can be split into six different phases: planning, conceptual design, detailed design, testing / prototype, production and market launch as well as product review. Thereby, LCA is mostly integrated into the first three stages. The LCA results are analyzed with a SWOT analysis and hotspots can be elaborated. In a SWOT analysis strength, weaknesses, opportunities and threats are determined.

Hotspots are the most environmentally contributing elements inside of a product and can be identified with literature or by using the LCA-method described later. Furthermore, computer-aided design (CAD) can be additionally used not only for meeting physical or design requirements but for collecting data about material flows which are used for the LCA inputs during the LCI [10].

To get an overview of the information required for performing an LCA a table can be created which includes data about the component, the type of info required, characteristic parameters as well as indicators which are needed for the parametrization (e.g., spatial dimensions, energy balance) [11].

Baumann and Tillmann [12] offer a solid overview of the general life cycle assessment methodology and its application in product comparisons. However, their work is more about standalone life cycle assessments and does not show how life cycle assessment can be effectively connected with the actual product development process, especially not in a digital or model-based way.

Suh and Hwang [13] take a step further and present a design for environment (DfE) framework. They emphasize the integration of environmental considerations, including digital tools or simulations that could help to understand the physical impact of environmental changes on the product performance.

The method presented in this work aims to combine the results of the life cycle assessment directly with modeling and simulation techniques based on MBSE. By doing so, the impact of design changes can be evaluated not only in terms of sustainability, but physical performance as well. This helps to find a better balance between environmental improvements and technical functionality.

III. METHOD

To pursue a more sustainable development, ecological effects of a product need to be considered in its design phase already. However, combining a sustainable design with the physical requirements of a product can be challenging. Therefore, a new method was developed in this study to address this issue, which makes use of connecting Life Cycle Assessment and

system modelling. The priority of the conducted methodology is the combination of design and resource conservation on a physical level.

Fig. 2: Overview of the five-step method

The connection is needed to bridge the gap described and to ensure a more sustainable Product Development which includes the design phase and environmental data. The method (see Fig. 2) is divided into five distinct steps.

A. Initial LCA

In the preliminary stage of the process, a Life Cycle Assessment is conducted on a specific component of the product, for example, the stator or rotor of an electric vehicle. This may be accomplished in accordance with the DIN EN ISO 14040 and 14044. Firstly, the objective and purpose of the product are established, the so called goal and scope. It is important to emphasize that the functional unit (FU) of the product must be identical to the FU of the improved product in the third working step. It is also critical for the Life Cycle Assessment and step 4 of the methodology to set the system boundaries, which means it must be defined which system boundary, e.g., cradle to gate, is being observed. Once the goal and scope of the product have been established, the materials utilized in its construction must be identified through the analysis of primary or secondary data sources. Subsequently, the LCA can be conducted in an LCA program, such as LCA for Experts or OpenLCA [17]. This allows for the identification of components that exert a significant influence on the ecological footprint of the product. It is also important to note that the chosen impact assessment method, e.g., CML or ReCiPe, affects the results.In the conducted LCA, different impact categories are selected such as the global warming potential (GWP) or the freshwater ecotoxicity potential (FAETP), which are important according to the set goal and scope of the analysis.

B. Working out hotspots

In the subsequent step, the results of the Life Cycle Assessment are subject to rigorous examination and analysis. Subsequently, the results of the various impact categories must be tabulated in order to ascertain which flows and components of the selected product exert the greatest influence within the respective impact categories.

However, this stage often results in the identification of numerous specific areas of concern within each impact category. In this process, the most significant hotspots with the highest environmental impact of the product must be identified.

Should the relevant hotspots be selected, assumptions must then be made regarding material substitution or reduction. Such assumptions may be based on existing literature or on new ideas. It is crucial to ensure that all assumptions are adequately justified and documented.

For instance, material substitution or the alteration of a component's weight, which has a significant impact on the overall assessment, can serve as a novel data point for the enhanced Life Cycle Assessment in the third phase of the process.

C. Data sorting

In the third step of the shown method a second LCA is conducted based on the assumptions and materials substitution, depending on the goal which must be achieved.

For the second LCA, the material data or masses are changed for processes linked to relevant hotspots. Thereby, it is crucial that the improved LCA has the same FU and selection of impact categories as the original LCA.

After the improved LCA is conducted the results are tabulated in the same way and in the same order as in the original LCA. It is important for further steps that the results must be stored in a certain way, this can be done with separate tabulations, for example. These tabulations can be used to change relevant physical parameters in the chosen simulation software.

Furthermore, it can be quantified which components of a product have the highest impact. Either the chosen components can be analyzed more individually or the whole LCA, this depends on the determined goal and scope of the product. In the following steps, it will be identified how the changes affect the environmental impact of the product.

D. System topology transfer

The generated information, which are conducted in step three, are transferred to a system topology. The goal of the topology is that the impact categories can be used and assigned as stereotypes for technical components. The fourth step entails the utilization of modelling and simulation software. In the selected software, the topology of the product or the selected components must be implemented.

As part of the methodology, several codes are developed, which are key elements in the subsequent process. At the start, the topology is modelled, which resembles the old state of the product to be improved.

Second, a program is run which assigns the tabulated LCIA-data to the respective components in the topology. This can be done by allocating impact categories in the form of stereotypes. Stereotypes can be used to allocate special properties to components.

The stereotypes can be divided into the different life cycle phases of the product. For each phase the impact is given as LCIA data. Subsequently, the resulting sum of environmental impacts is stored, e.g., in a file format, to compare it to the results of other topologies.

This format must be uniform for comparison. The process is repeated for the improved system topology.

Once the new environmental impact has been summed up and stored, it can be contrasted.

The comparison is accomplished by a final code, which is parametrized by the two results of the previous actions.

The output of this function should contain numerical or graphical information about the relative improvement between the old and improved system.

E. Finding Solutions

In the last step of the conducted method the results are shown in a table, which can be seen in the example of section four in this paper. The table is relied on the Degree of Improvement (DI) of da Luz et al. [4].

The DI is based on the results of the first conducted LCA compared to the results of the improved LCA. The DI is calculated with the following equation (1):

$$DI = (\sum \text{Value obtained in the matrix}) / (\sum \text{maximum matrix sore}) \times 10 \quad (1)$$

The number 10 is the maximum number that the DI can achieve. The higher the DI, the better the improvement achieved. The results of each phase are colored, indicating in which impact category of the LCIA an improvement was achieved the most.

IV. Exemplary application

In the following section, an exemplary application of the method presented above is demonstrated. For this purpose, a simplified model of an electric motor is created.

A. Initial LCA of a PMSM

To perform an initial Life Cycle Assessment, a permanent magnetized synchronous motor (PMSM) is modelled inside the LCA software LCA for Experts ® [18]. The system components are determined and weighed according to an open source LCA-study [12], meaning that secondary data is being used instead of taking own measures for primary data. Since the secondary data and the associated LCA are performed with different LCA software, comparative flows and materials are used if the original are not included in the GaBi ® database [20]. Once the system is implemented in the LCA software and the masses are set based on the functional unit, the life cycle impact assessment (LCIA) can be calculated using the GaBi ® database [20] for further processing.

B. Identification of hotspots

Next up, the hotspots of the LCIA need to be identified. In the case of using LCA for Experts ® [18], this could be seen in resulting diagrams. Another way to reveal impactful components can be to research the topic and see if other LCAs have been conducted for similar and comperable technologies. It is important to note that these results must be detailed, because without good information about the impact of different components of the analyzed product, it is difficult to identify hotspots.

In this example, the rotor and stator are seen as impactful system parts. The rotor is resulting the highest impact category values and the stator is chosen as a way to demonstrate another interchangeable component, e.g., in the form of reducing copper wiring. For the comparison of the LCIA-results later, the material inputs of the components with the highest impact are changed to new input materials for the following comparative LCA. The aim is to see if the change made a difference to the LCIA results.

C. Sorting the LCIA data

Subsequently, the LCA needs to be re-done for the identified hotspots and the associated changes which are made, e.g., material substitution. In the conducted example shown in this study, this step is simply displaced by varying the LCIA data by a random factor for demonstration purposes. In reality this would of course be done by performing another environmental Life Cycle Assessment based on the identified hotspots of step 2 of the methodology to evaluate real recommendations for action.

This step is essential for the linkage between the LCA data and the system model. The LCIA results need to be filtered and sorted, consisting of the classification of processes inside of the flow diagram in LCA for Experts ® [18] into several groups. A group could be one component of the product, for example, the rotor production or the rotor operation, including detailed impact assessment results.

Each group of processes matches a phase inside of the life cycle of the product. The values for the components and corresponding phases are separated and stored individually in different tables to be easily read by a script.

The choice of phases depends on the set functional unit and system boundaries. In this case the system boundary is set to cradle to gate which means, in the case of this demonstration, that two phases were distinguished: Base material acquisition and production. Drawing boundaries between phases is not always easy, however, so discussing it is recommendable.

D. Adapting LCIA data to the system model

After having drawn the system boundaries and assessed the life cycle impact, the data is applied to the topology of the electrical machine. The modelling has been examined in MathWorks ® System Composer [16] and is shown in Fig. 3.

As the stator and rotor were set as hotspots, they have been assigned as variable components. In this case, two variables, namely A (e.g., the original copper winding) and B (e.g., an optimized aluminum winding), have been chosen as demonstrative alternatives.

The Profile Editor was used to declare stereotypes.

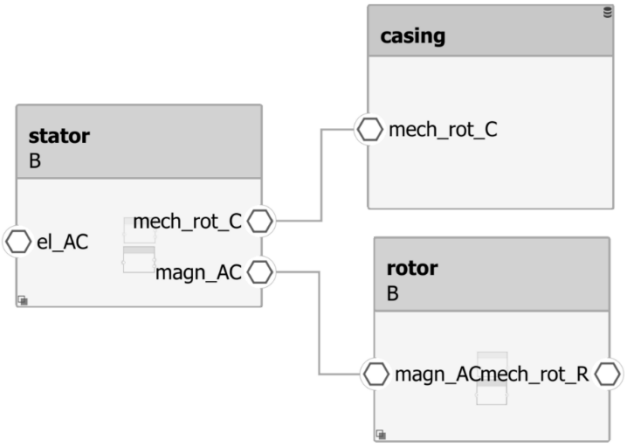


Fig. 3: PMSM model with applied stereotypes shown in System Composer [16]

Impact categories are assigned from the CML 2001-2016 method [19]. The connection between the system topology and LCIA is important to harmonize the technical requirements and the ecological footprint for example. The systemic linking of physical parameters to the LCA must be integrated into the general process for creating system models. The aim should be to optimize proven MBSE methods or to develop new methods [13], [14].

E. Evaluation of possible solutions

After calculating and simulating the conducted product in step 4 of the methodology the so-called DI is calculated with equation (1) in chapter III E.The equation is performed with a specialized MATLAB-code which calculates the DI and includes the results into a table which shows a color coded degree of improvement in each analyzed impact category of the LCIA. Figure 4 shows the table of improvement, the darker the impact category the better the improvement.

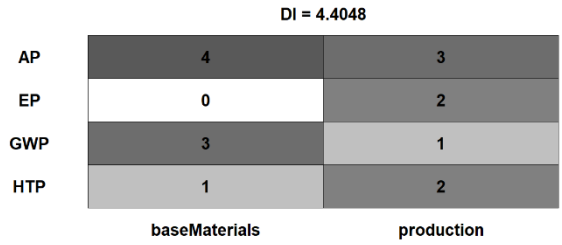


Fig. 4: Exemplary diagram of the degree of improvement (DI) based on [4]

For the case that data should not be available, it could be indicated by a separate color.

The table is divided into the two phases of the system boundary: base material acquisition and production. It is crucial to examine various phases to gain comprehensive understanding of the potential impact of improvements.

For instance, improvement in one impact category may be beneficial in the initial phase but may not yield the same results in the subsequent phase. The production phase for example could be enhanced to reduce the human toxicity potential (HTP) without affecting the GWP as much.

Ultimately, the table presents potential avenues for further enhancements to the LCA, which means the methodology can be repeated until the desired outcomes are attained adequately.

V. BENEFITS AND LIMITATIONS

The method conducted brings a lot of benefits but also some limitations which will be discussed in the following section. A significant advantage is that the methodology can be employed even in the absence of carrying out an LCA. A sufficient basis for the analysis can be provided by literature that includes detailed LCA results and allows the identification of hotspots.

Furthermore, while familiarity with the process of conducting an LCA is advantageous, it is not a prerequisite for success. This may be an advantage in the Product Development sector, where the methodology of an LCA is not widely disseminated. It is only necessary to possess knowledge of the LCA methodology if primary data pertaining to a given product is available and secondary data is not sufficient enough for the analysis. In such instances, the creation of a new LCA may be required and a comprehensive understanding of the subject matter would be indispensable. To facilitate a comparison between the enhanced LCA and the original LCA, one may utilize a calculation program, potentially relatively inexpensive in comparison to the cost of an LCA license.

The methodology enables the identification of environmental hotspots associated with a given product, facilitating a comparison with an improved product if the latter exhibits a reduced environmental impact relative to the former. The methodology employed is limited in that it requires both familiarity with modeling and simulation software and knowledge of how to write code in MATLAB ® [16] for the DI. Another limitation of MATLAB ® [16] is the necessity of a license, which is also a costly requirement. Furthermore, it has been necessary to enter all the analyzed stereotypes into the System Composer [16] tool, which has also required a significant investment of time. The primary reason for selecting System Composer [16] is that the program offers a user-friendly yet effective modeling environment for complex systems.

In conclusion, the conducted method presents a greater number of advantages than limitations, as it outlines the process of environmental development and improvement.

VI. CONCLUSION AND OUTLOOK

The methodology presented offers a way to integrate Life Cycle Assessment into the design phase of a product. This linkage can allow for uncovering weak points in terms of the

environmental impact of certain components and ideally indicates in which life cycle phases improvements are most helpful. The process of doing the LCA, identifying hotspots (and possible enhancements), sorting the data and applying it to an interchangeable system model could potentially be adapted into the workflow of common system engineering.

However, there are a few challenges. Currently, the integration process is not automated. The manual steps required to import, process, and export data add complexity and time, which limits usability in early design phases. Automating the workflow and linking LCA data more directly with system modelling tools (e.g., MBSE environments or digital twins) would enable more efficient ecological comparisons between system configurations.

In future work, the methodology will be extended by coupling it more closely with physical simulation models, allowing environmental impacts to be assessed in parallel with technical performance indicators. This would create a combined framework where engineers can directly evaluate the trade-offs between sustainability and stem functionality during early design and development decisions. For doing so, the variations made to the hotspot-components need to be linked to the physical parameters in the simulation software. This involves further research on how changes to the material or mass fractions affect the physical characteristics of the analyzed components. Applying an improved version of the method to real-world cases across different industries will help to validate its scalability. However, automation is crucial for integrating it into a product development process. Ultimately, this would allow for easier access to ecological comparisons between system topologies.

REFERENCES

[1] European Commission, “A new Circular Economy Action Plan For a cleaner and more competitive Europe.” [Online]. Available:https://eurlex.europa.eu/resource.html?uri=cellar:9903b325-6388-11ea-b735-01aa75ed71a1.0017.02/DOC_1&format=PDF

[2] Brundtland et al., “Report of the Word Commission on Environment and Development: Our Common Future,” 1987, [Online]. Available: https://sustainabledevelopment.un.org/content/documents/5987our-common-future.pdf

[3] European Commission, “A new Circular Economy Action Plan,” 2019.

[4] L. M. Da Luz, A. C. Francisco, C. M. Piekarski, and R. Salvador, “Integrating life cycle assessment in the product development process: A methodological approach,” *Journal of Cleaner Production*, vol. 193, pp. 28–42, Jan. 2018, doi: 10.1016/j.jclepro.2018.05.022.

[5] E. Lacasa, J. L. Santolaya, and A. Biedermann, “Obtaining sustainable production from the product design analysis,” *Journal of Cleaner Production*, vol. 139, pp. 706–716, Dec. 2016, doi: 10.1016/j.jclepro.2016.08.078.

[6] I. Bereketli Zafeirakopoulos and M. Erol Genevois, “An Analytic Network Process approach for the environmental aspect selection problem — A case study for a hand blender,” *Environmental Impact Assessment Review*, vol. 54, pp. 101–109, Sep. 2015, doi: 10.1016/j.eiar.2015.05.002.

[7] D. Inkermann, “Potentials of integrating MBSE and LCA to handle uncertainties and variants in early design stages,” in *DS 119: Proceedings of the 33rd Symposium Design for X (DFX2022)*, The Design Society, 2022, pp. 10–10. doi: 10.35199/dfx2022.19.

[8] European Commission, “Life Cycle Assessment & the EF methods.” [Online]. Available: https://green-business.ec.europa.eu/environmental-footprint-methods/life-cycle-assessment-ef-methods_en

[9] D. Chang, C. K. M. Lee, and C.-H. Chen, “Review of life cycle assessment towards sustainable product development,” *Journal of Cleaner Production*, vol. 83, pp. 48–60, Jan. 2014, doi: 10.1016/j.jclepro.2014.07.050.

[10] N. Ko, R. Graf, T. Buchert, M. Kim, and D. Wehner, “Resource Optimized Product Design – Assessment of a Product’s Life Cycle Resource Efficiency by Combining LCA and PLM in the Product Development,” *Procedia CIRP*, vol. 57, pp. 669–673, Jan. 2016, doi: 10.1016/j.procir.2016.11.116.

[11] R. Luglietti, P. Rosa, S. Terzi, and M. Taisch, “Life Cycle Assessment Tool in Product Development: Environmental Requirements in Decision Making Process,” *Procedia CIRP*, vol. 40, pp. 202–208, Jan. 2016, doi: 10.1016/j.procir.2016.01.103.

[12] Baumann & Tillman: "The Hitch Hiker's Guide to LCA: An Orientation in Life Cycle Assessment Methodology and Application" (2004). This comprehensive guide is widely used in the field of Life Cycle Assessment. ISBN 9144023642

[13] B. Suh and Y. Hwang, *Design for Environment (DfE): Strategies, Practices, and Guidelines*. Springer, 2005. [Online]. Available: https://dl.acm.org/doi/10.5555/1242339.1242343

[14] *DIN EN ISO 14040:2009-11*, Environmental management – Life cycle assessment – Principles and framework (ISO 14040:2006 + Cor. 1:2009); German and English version EN ISO 14040:2006. Berlin, Germany: Beuth Verlag, 2009.

[15] *DIN EN ISO 14044:2021-02*, Environmental management – Life cycle assessment – Requirements and guidelines (ISO 14044:2006 + Amd. 1:2017 + Amd. 2:2020); German and English version EN ISO 14044:2006 + A1:2018. Berlin, Germany: Beuth Verlag, 2021

[16] MathWorks, *MATLAB, Simscape, and System Composer R2023a*. Natick, MA, USA. [Online]. Available: https://www.mathworks.com

[17] GreenDelta, *openLCA Version 1.11*, Berlin, Germany. [Online]. Available: https://www.openlca.org

[18] thinkstep AG (now Sphera), *LCA for Experts*, Leinfelden-Echterdingen, Germany.

[19] Institute of Environmental Sciences (CML), Leiden University, *CML-IA Baseline Method*, 2001–2016.

[20] Sphera Solutions GmbH, *GaBi Database*, Leinfelden-Echterdingen, Germany. [Online]. Available: https://gabi.sphera.com

A PyTorch-Compatible Spike Encoding Framework for Energy-Efficient Neuromorphic Applications

Alexandru Vasilache[1,2], Jona Scholz[1], Vincent Schilling[2], Sven Nitzsche[1,2],
Florian Kaelber[3], Johannes Korsch[3], Juergen Becker[2]
[1] *FZI Research Center for Information Technology, Karlsruhe, Germany*
[2] *Karlsruhe Institute of Technology, Karlsruhe, Germany*
[3] *NXP Semiconductors Germany GmbH, Munich, Germany*
email:{vasilache,jona.scholz,nitzsche}@fzi.de
utmlb@student.kit.edu Juergen.Becker@kit.edu
{florian.kaelber,johannes.korsch}@nxp.com

***Abstract*—Spiking Neural Networks (SNNs) offer promising energy efficiency advantages, particularly when processing sparse spike trains. However, their incompatibility with traditional datasets, which consist of batches of input vectors rather than spike trains, necessitates the development of efficient encoding methods. This paper introduces a novel, open-source PyTorch-compatible Python framework for spike encoding, designed for neuromorphic applications in machine learning and reinforcement learning. The framework supports a range of encoding algorithms, including Leaky Integrate-and-Fire (LIF), Step Forward (SF), Pulse Width Modulation (PWM), and Ben's Spiker Algorithm (BSA), as well as specialized encoding strategies covering population coding and reinforcement learning scenarios. Furthermore, we investigate the performance trade-offs of each method on embedded hardware using C/C++ implementations, considering energy consumption, computation time, spike sparsity, and reconstruction accuracy. Our findings indicate that SF typically achieves the lowest reconstruction error and offers the highest energy efficiency and fastest encoding speed, achieving the second-best spike sparsity. At the same time, other methods demonstrate particular strengths depending on the signal characteristics. This framework and the accompanying empirical analysis provide valuable resources for selecting optimal encoding strategies for energy-efficient SNN applications.**

***Keywords*-Spiking Neural Networks (SNNs); spike encoding; neuromorphic; energy-efficiency; Reinforcement Learning (RL).**

I. INTRODUCTION

An increasing number of tasks rely on Machine Learning (ML) to enhance accuracy, streamline development processes, or facilitate automation in the first instance. Neural Networks (NNs) are often employed for this purpose due to their flexibility and capacity to solve even the most complex tasks. As neural networks become more widespread, they are also being employed with increasing frequency in embedded systems. However, their deployment in such embedded contexts is frequently constrained by the requisite computing power and the associated high energy demands. Given these considerations, SNNs may offer a promising avenue for integrating high-performance machine learning into embedded systems, provided that suitable neuromorphic hardware is available.

In order to utilize SNNs, it is necessary to have access to adequate spiking data, also referred to as event-based data. Moreover, the event-based data must be sparse to leverage the advantages of spiking neural networks fully. Ideally, an event-based sensor would be employed to provide this data. However, there is a notable scarcity of such sensors on the market. Currently, the only commercially available event-based sensors target the vision modality through event-based cameras [1][2].

As an alternative approach, non-spiking data can be converted using spike encoding algorithms. Such algorithms are designed to transform real-valued data into spike trains, which can then be utilized as input for an SNN. The existing literature classifies spike encoding algorithms into two principal categories: rate coding and temporal coding [3]. In rate coding, the signal amplitudes are directly mapped to spike frequencies, resulting in high spiking activity. In contrast, temporal coding results in the firing of spikes only when specific events occur, thereby making the timing of spikes crucial and reducing the overall number of spikes. This sparsity renders temporal coding more power-efficient and suitable for low-power applications. This paper focuses on evaluating four temporal encoding methods: Leaky Integrate-and-Fire (LIF), Step-Forward (SF), Pulse Width Modulation (PWM), and Ben's Spiker Algorithm (BSA). These methods were chosen as they are frequently discussed and compared in spike encoding literature [4][5][6][7], serving as representative techniques for this study.

Even though many spike encoding methods are publicly available, no open-source framework or library currently groups various such algorithms and allows for straightforward integration into popular machine-learning workflows. Accordingly, this work seeks to provide a framework that offers out-of-the-box PyTorch support and automatic parameter optimization of the encoding algorithms for a given dataset. Additionally, we evaluate the performance of various spike encoding algorithms for specific signal types and implement them on an embedded platform to assess runtime and power demand. The library is publicly accessible at its GitHub Repository [8]. This work aims to reduce the overhead associated with integrating SNNs into real-world applications.

The remainder of this paper is organized as follows: Section II compares our work with existing state-of-the-art approaches. Section III then summarizes the investigated methods and details the experimental setup, followed by Section IV

 ISBN: 978-1-68558-278-4

which displays the results regarding reconstruction error, spike sparsity, runtime, and power consumption. Subsequently, Section V discusses these results focusing on energy efficiency and accuracy for different signal types. Finally, Section VI outlines the structure of the provided spike encoding library.

II. Related Work

A. Evaluation of Spike Encoding Algorithms

Wang et al. [4] implemented four spike encoding algorithms on a Field Programmable Gate Array (FPGA) and compared them by their speed, power consumption, accuracy, and robustness to noise. They evaluated Sliding Window (SW) encoding [9], BSA [10], SF encoding [11] and the PWM algorithm [12]. Furthermore, they verified their evaluation results on two real-world applications: tactile signal encoding reconstruction and a robotic arm control task. They found that overall BSA had the highest power consumption and was therefore not recommended for most applications, with the notable exception of encoding square signals, where its energy efficiency was high. PWM performed well in most signal reconstruction tests, and its simple implementation made it favorable for real-world use. However, its accuracy may be decreased if an unsuitable curve-fitting algorithm is chosen. Also, if nonlinear operations are used, the power consumption may be elevated. This added variability is reduced in SF encoding, which only has one adjustable parameter. While SF encoding also performed well in most signal reconstruction tests, its accuracy could degrade when high changes in the signal amplitude occurred. Finally, SW encoding had no notable advantages over the other algorithms.

Chen et al. [5] compared (among others) SF to PWM and provided further evidence that both algorithms achieve high signal reconstruction accuracy. They observed that SF is more accurate at lower thresholds, but the increased spike count may also increase power consumption.

In [6], the authors suggest a workflow for selecting, optimizing, and validating spike encoding methods, which they evaluate on SF, BSA, and others. Their experiments provide further evidence for SFs versatility and robustness, consistent with similar studies' findings. However, their evaluation of BSA contradicts those of [4] regarding step signals since they observed high signal reconstruction errors for this signal type. An implementation choice may partly explain this since they use a multiplicative threshold rather than a subtractive one.

Yarga et al. [7] applied BSA, LIF encoding, and three additional methods to encode voice recordings. Their results are not fully applicable to our comparison, as they focused on the accuracy of a classification task rather than evaluating reconstruction error. Initially, features were extracted into a spectrogram or cochleagram and then converted to spike trains using the evaluated encoding methods. These spike trains were subsequently processed by a Convolutional Neural Network (CNN) for classification. Their evaluation metrics included spike density and classification accuracy, which are indirectly related to the energy efficiency of the encoding and its impact on information loss or gain. However, these metrics may not fully represent the performance of the encoding methods. Most methods allowed spike density to be adjusted by modifying their parameters, such as the membrane threshold in LIF encoding. On spectrogram features, their findings showed that LIF encoding produced the highest accuracy across most spike densities among the evaluated methods. Notably, when using spectrogram features, BSA also performed well at very low spike densities. However, on cochleagram features, BSA performed poorly, and LIF was outperformed by other methods at low spike densities. At higher spike densities, LIF once again achieved the highest overall classification accuracy. In summary, their findings suggest that LIF encoding can achieve both high classification accuracy and low spike densities, contributing to better energy efficiency. In contrast, BSA is more restricted to specific scenarios where it performs well.

B. Spike Encoding Repositories

The currently available spike encoding repositories tend to prioritize applications outside the domain of machine learning. SpikeCoding [13] emphasizes the real-time control of robots and has been developed to integrate with ROS rather than focusing on machine learning workflows. Similarly, Spikes [14] offers foundational tools for spike generation but is not designed for integration with neural network models. In contrast, the proposed framework addresses this limitation by providing a PyTorch-compatible, open-source library focused on machine learning and neural network applications, including support for reinforcement learning environments, facilitating broader use in machine learning research and practice.

III. Methods

This section presents the investigated encoding methods and their evaluation on an embedded hardware platform and outlines the experimental setup. This includes an explanation of the underlying hardware processes, the types of signals used, and the parameter optimization of the encoding algorithms.

A. Encoding Algorithms

SF encoding is a straightforward and efficient method for signal processing. A spike is generated whenever the signal surpasses a defined baseline, which is subsequently incremented by a constant value. Conversely, if the signal did not exceed the baseline, the baseline is lowered by a constant value. A pseudocode implementation is shown in Figure 1 (based on [15]). Encoding spikes in this way is similar to the LIF encoding method, although the latter is inspired by a spiking neuron model of the same name.

In LIF encoding, the signal serves as an input current that increases the membrane potential. When the potential exceeds a predefined threshold, a spike is emitted. The membrane potential decays proportionately to a decay variable at each time step. It is important to note that the original signal must be normalized, as neither the decay rate nor the threshold adapts to the varying range of possible signal values. A pseudocode implementation of the LIF encoding method is provided in

```
Input: signal, threshold
Output: spike_train
base ← 0, up_spikes ← 0, down_spikes ← 0
for t = 1 to time_steps do
    if signal(t) > base + threshold then
        up_spikes(t) ← 1
        base ← base + threshold
    else if signal(t) < base − threshold then
        down_spikes(t) ← −1
        base ← base − threshold
    else
        up_spikes(t) ← 0
        down_spikes(t) ← 0
    end if
end for
spikes ← up_spikes + down_spikes
```

Figure 1. SF Encoding Method

```
Input: signal, threshold, membrane_constant
Output: spike_train
signal ← min_max_normalize(signal)
signal ← signal × 2 − 1
voltage ← 0, up_spikes ← 0, down_spikes ← 0
for t = 1 to time_steps do
    voltage ← voltage + signal(t)
    if voltage > threshold then
        up_spikes(t) ← 1
        voltage ← 0
    else if voltage < −threshold then
        down_spikes(t) ← −1
        voltage ← 0
    else
        up_spikes(t) ← 0
        down_spikes(t) ← 0
    end if
    voltage ← voltage × membrane_constant
end for
spikes ← up_spikes + down_spikes
```

Figure 2. LIF Encoding Method

Figure 2 (based on [7]), where min_max_normalize refers to a rescaling of the range of features to $[0, 1]$.

Similar to LIF encoding, BSA requires data normalization but employs a fundamentally different approach. Instead of encoding a continuous signal into spikes, BSA assumes the signal has already been transformed into a spike train. The objective is to decode this spike train with minimal reconstruction error. The encoding process is assumed to involve convolution with a Finite Impulse Response (FIR) filter, and decoding requires reversing this operation through deconvolution. At each step, an error term is computed to measure the sum of the differences between the filter and the signal, while a second error term quantifies the sum of the signal itself. Spikes are emitted when the first error term is smaller than the second minus a threshold value. Upon spike emission, the filter is subtracted from the signal. A pseudocode implementation of BSA encoding is provided in Figure 3 (based on [16]).

Finally, PWM is based on comparing the input signal to a carrier (or "reference") signal and emitting spikes whenever the carrier signal exceeds the input signal. Once this condition is met, the input signal must first fall below the carrier signal before exceeding it again for a new spike to be emitted. The carrier signal is a sawtooth wave with an adjustable frequency

```
Input: signal, filter_order, filter_cutoff, threshold
Output: spike_train
signal ← normalize(signal)
fir_coeff ← fir_filter(filter_size = filter_order + 1, filter_cutoff, sampling_frequency = 1)
spikes ← 0
for t = 1 to time_steps do
    err1 ← 0
    err2 ← 0
    for j = 1 to filter_size do
        if t + j − 1 ≤ time_steps then
            err1 ← err1 + |signal(t + j − 1) − fir_coeff(j)|
            err2 ← err2 + |signal(t + j − 1)|
        end if
    end for
    if error1 ≤ err2 − threshold then
        spikes(t) ← 1
        for j = 1 to filter_size do
            if t + j − 1 ≤ time_steps then
                signal(t + j − 1) ← signal(t + j − 1) − fir_coeff(j)
            end if
        end for
    else
        spikes(t) ← 0
    end if
end for
```

Figure 3. BSA Encoding Method

parameter. As with the other encoding methods, the input signal must be normalized to ensure it overlaps with the carrier signal. A pseudocode implementation of the PWM encoding method is shown in Figure 4 (based on [17]).

```
Input: signal, frequency, downspike (boolean)
Output: spike_train
signal ← normalize(signal)
carrier ← sawtooth(frequency)
neg_carrier ← 1 − carrier
pwm ← 0, up_spikes ← 0, down_spikes ← 0
for t = 1 to time_steps do
    if signal(t) < carrier(t) then
        pwm(t) ← 1
    else if signal(t) > neg_carrier(t) and downspike = True then
        pwm(t) ← −1
    else
        pwm(t) ← 0
    end if
end for
for t = 2 to time_steps do
    if pwm(t) = 1 and pwm(t − 1) ≠ 1 then
        up_spikes(t) ← 1
    else if pwm(t) = −1 and pwm(t − 1) ≠ −1 then
        down_spikes(t) ← −1
    end if
end for
spikes ← up_spikes + down_spikes
```

Figure 4. PWM Encoding Method

B. Experimental setup

To evaluate the performance characteristics, such as runtime and power consumption, we utilized specific embedded hardware. While the core encoding algorithms are implemented in C++, the subsequent performance results presented are specific

to the chosen platform and configuration. We use an MCX-N947-EVK development board from NXP Semiconductors as an embedded evaluation platform. The MCX N is a multi-core System on Chip comprising two Arm Cortex®-M33 cores, 2MB flash, 512kB SRAM, a digital signal co-processor, and a neural processing unit for accelerating inference of conventional neural networks. One Cortex-M33 has a maximum clock speed of 150MHz and offers a favorable trade-off between energy efficiency, real-time determinism, and system security. The power consumption comes down to 57 µA/MHz in active mode. For our evaluation, we configured the system's clock frequency to 12 MHz and set the power mode to overdrive with a DCDC core voltage of 1.2V, balancing power consumption and performance.

The interaction with the development board is facilitated by eRPC calls, wherein the board is treated as a server and the PC as a client. First, the client transmits its signal data to the board via UART. It then requests the execution of the selected encoding function on the transmitted data. Finally, the client requests the results once encoding is completed, and the board transfers them back. All measurements are performed between the start and end of the encoding method to exclude data transfer and communication overhead. Additionally, the normalization operations required by some methods have been excluded.

The power consumption of each method was measured over 10 hours at a sample rate of 62500 samples per second, with the results averaged over the entire duration. It should be noted that even when no operation is performed, the board continues to consume power. Therefore, a baseline power consumption measurement was first taken and subsequently subtracted from each method's measurements, allowing for a direct comparison between the encoding methods. The time measurements were taken using C functions, which initiated and terminated a timer at the beginning and end of method execution, respectively.

In order to quantify the extent of information loss incurred during the encoding process, we calculate the reconstruction error for each encoding method by decoding the encoded signal and measuring the Mean Squared Error (MSE) between the reconstruction and the original signal.

C. Evaluated Signals

This study selected four artificially generated signals with 16384 time steps, each normalized to unit mean and variance, to assess the encoding methods. These specific signals were chosen to represent a diverse set of common temporal dynamics, including irregularity, drift, abrupt transitions, and smooth periodicity.

The "Vibration" signal (Figure 5 (a)) simulates oscillations with a specific standard deviation and frequent fluctuations, challenging the encoders due to its irregularity. The "Trended" signal (Figure 5 (b)) introduces drift, testing the stability of the reconstruction over time. The "Rectangular" signal (Figure 5 (c)) resembles a square wave, providing insights into the methods' performance on constant and abruptly transitioning

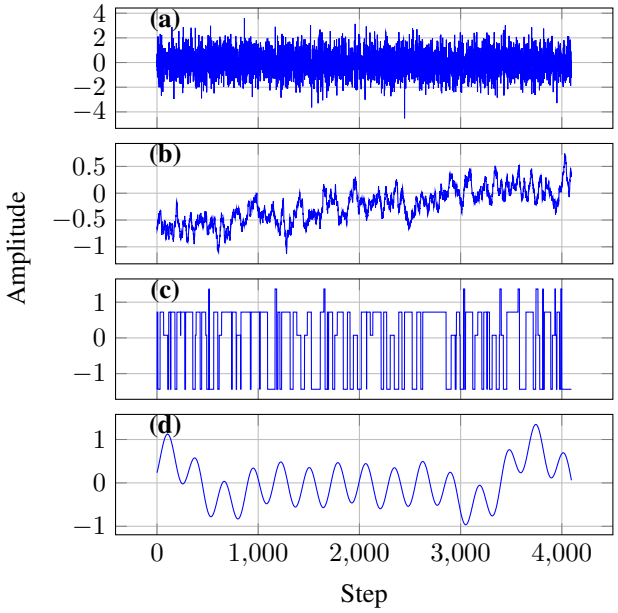


Figure 5. Four different signal types: (a) Vibration Signal, (b) Trended Signal, (c) Rectangular Signal, (d) Sinusoidal Signal.

values. Lastly, the "Sinusoidal" signal (Figure 5 (d)), smooth and noise-free, assesses the preservation of periodic patterns.

D. Paramter Optimization

Each encoding method was optimized through 500 trials. In each trial, the chosen set of parameters was used to encode the signal into spikes. The resulting spike train was then decoded to reconstruct the original signal, and the reconstruction error, quantified as the MSE between the original and reconstructed signals, was computed. The optimization aimed to minimize this reconstruction error by adjusting the encoding parameters.

To perform this optimization, we employed Optuna [18], a widely used hyperparameter optimization framework implemented in Python. For most encoding methods, random sampling was used to explore the parameter space randomly. However, for the BSA method, which involves tuning three interdependent parameters and is computationally demanding, the TPESampler [19] was utilized due to its efficiency in complex, multi-variable optimizations.

IV. Results

This section presents the results for each encoding method applied to different signal types, evaluated using four main criteria: reconstruction error (TABLE I), encoding time, power consumption (TABLE III), and spike sparsity (TABLE II). Spike sparsity is expressed as the ratio of spikes to the total signal length (16384).

The absolute power values reported in TABLE III reflect the average power consumption measured over ten hours for the entire board. Additionally, the absolute power was measured for the no-operation (NOP), resulting in a value of 99.74 nW. The NOP reference power is subtracted from this value to determine the dynamic power consumed by each encoding

TABLE I. RECONSTRUCTION ERROR (MSE)

Encoding Method	LIF	SFC	PWM	BSA
Vibration	**0.370641**	0.487395	1.099153	0.845590
Trended	0.102157	**0.000970**	0.015004	0.006510
Rectangular	0.081864	0.157202	0.172042	**0.063650**
Sinusoidal	0.126749	**0.000139**	0.004631	0.013472
Mean Error	0.170353	**0.161427**	0.322707	0.232305

TABLE II. SPIKE SPARSITY

Encoding Method	LIF (%)	SFC (%)	PWM (%)	BSA (%)
Vibration	**36.83**	41.26	50.00	53.81
Trended	65.59	35.82	**3.66**	61.34
Rectangular	62.47	**10.20**	14.33	48.48
Sinusoidal	42.72	35.14	**3.02**	48.70
Mean Sparsity	51.90	30.60	**17.75**	53.08

TABLE III. TIME, POWER AND ENERGY MEASUREMENTS

Encoding Method	LIF	SF	PWM	BSA
time (ms)	127.15	**123.52**	691.32	2238.58
absolute power (nW)	109.45	**106.61**	111.15	119.74
dynamic power (nW)	9.71	**6.87**	11.41	20.00
dynamic energy (pW h)	0.34	**0.24**	2.19	12.44
time wrt. SF (%)	2.94	**0.00**	459.64	1712.28
dynamic power wrt. SF (%)	41.34	**0.00**	66.08	191.12
dynamic energy wrt. SF (%)	41.67	**0.00**	812.50	5083.33

method. The last three rows of the table indicate the encoding time, dynamic power, and dynamic energy consumption for each method, expressed as a percentage increase relative to the SF converter, which shows the lowest values across all methods.

V. Discussion

This study evaluated four encoding methods—LIF, SF, PWM, and BSA—based on their performance in reconstructing signals, spike sparsity, and energy efficiency.

LIF demonstrated high accuracy with vibration data due to its suitability for signals fluctuating around a baseline. However, its mean bias hampered performance with trended signals and capturing extreme values in sinusoidal signals. Despite a high spike rate, its simple implementation resulted in good energy efficiency, aligning with findings from Yarga et al. [7] on its suitability for fluctuating signals.

The SF Converter generally achieved the lowest reconstruction error, especially with sinusoidal signals, and offered superior speed and energy efficiency. However, it exhibited instability with rectangular waveforms. Our results confirm prior evidence from Chen et al. [5] on SF's accuracy, showing it had the lowest mean reconstruction error (0.1614 - TABLE I) and minimal power consumption (TABLE III), underscoring its efficiency and versatility.

PWM, despite low spike sparsity, underperformed in reconstruction accuracy, struggling with fluctuating and rectangular waveforms. Its complex operations resulted in higher encoding time and power consumption. This contrasts with Wang et al. [4], who reported favorable reconstruction and power consumption for PWM in simpler implementations. Our results indicate PWM's higher reconstruction error (mean MSE of 0.3227) and elevated power consumption (812.5% relative to SF's energy usage), likely due to its sensitivity to parameter tuning.

BSA effectively filtered high frequencies and achieved low reconstruction error for rectangular signals (MSE of 0.0636) but suffered from high computational costs and sensitivity to initial signal values, consistent with Wang et al. [4]. This limits its practicality for real-time embedded systems, as its encoding duration and power consumption are significantly higher than SF.

Overall, SF emerged as the most energy-efficient and reliable method, while LIF and BSA provide niche strengths in specific applications. Future work may benefit from investigating hybrid approaches.

```python
import torch
from encoding.step_forward_converter import
    StepForwardConverter

signal = torch.tensor([0.1, 0.3, 0.2, 0.4, 0.8])

converter = StepForwardConverter()
spikes = converter.encode(signal)
```

Figure 6. Using the StepForwardConverter to encode a signal.

```python
# ... same as above ...
optimalthreshold = converter.optimize(signal)
optimized_converter = StepForwardConverter(
    optimalthreshold)

spikes = optimized_converter.encode(signal)
reconstructed_signal = optimized_converter.
    decode(spikes)
```

Figure 7. Minimal example of optimizing a converter and decoding signals in order to reconstruct them.

VI. Spike Encoding Framework

Our framework is based on the idea of a Converter. In our approach, a Converter is an object that can encode and decode signals. Optionally, Converters may also include a method of optimization that allows them to finetune their hyperparameters to a specific signal. In this way, each encoding method can achieve its highest reconstruction accuracy without the need to finetune its parameters manually.

For example, consider using our SF implementation depicted in Figure 6. LIF, PWM and BSA encoding all function analogously.

Decoding spike trains is achieved similarly, except that *encode* is replaced with *decode*. In Figure 7, we see how converters can be optimized and used to decode signals.

Additionally, we introduce a custom encoder designed explicitly for Gymnasium [20] environments, the GymnasiumEncoder. It extends rate encoding methods by adding utility methods tailored for reinforcement learning tasks. We also provide the BinEncoder, which utilizes Gaussian Receptive Field (GRF) encoding to transform single values into multiple bin responses. However, like the GymnasiumEncoder, it is restricted to encoding operations only.

VII. Conclusion and Future Work

This paper introduces a novel, open-source PyTorch-compatible Python framework for spike encoding, explicitly designed for machine learning and reinforcement learning applications. The framework offers support for a diverse range of encoding methods, encompassing conventional algorithms, such as LIF, SF, PWM, and BSA, as well as specialized components like a reinforcement learning-optimized encoder and Gaussian Receptive Field-based population coding. The framework is accompanied by documentation and testing, ensuring seamless integration into machine learning workflows and fostering accessibility and ease of use.

Furthermore, a comprehensive evaluation of the performance trade-offs of each encoding method was conducted by implementing them in C++ and testing them on embedded hardware. Our findings indicate that while SF exhibited the highest energy efficiency and fastest encoding time, it tended to falter in abrupt signal transitions. LIF demonstrated efficacy in handling fluctuating signals but exhibited limitations in the presence of trends or extreme values. PWM demonstrated lower accuracy and higher energy consumption than the other methods. In contrast, BSA demonstrated high accuracy for certain signal types and filtering capabilities but at the cost of increased computational demands. These comparative insights provide valuable guidance for selecting the most suitable encoding method, supporting the broader adoption of Spiking Neural Networks in machine learning applications.

Future work will focus on extending the framework's capabilities by implementing additional encoding algorithms, including both temporal and rate coding approaches, to allow for broader comparisons.

ACKNOWLEDGMENTS

This research is funded by the German Federal Ministry of Education and Research as part of the project "ThinKIsense", funding no. 16ME0564.

Supplementary Materials

We provide Python implementations for all the investigated algorithms. Our repository includes LIF encoding, SF, PWM, and BSA, as well as two additional algorithms. The first is a custom encoder particularly suited to reinforcement learning, especially to Gymnasium [20] environments. The second one implements a form of population coding that is based on Gaussian Receptive Fields. Our code can be accessed at [8].

References

[1] C. Aerne, *iniVation – Neuromorphic vision systems*, en-US.

[2] *PROPHESEE | Metavision Technologies*, en-US.

[3] D. Auge, J. Hille, E. Mueller, and A. Knoll, "A survey of encoding techniques for signal processing in spiking neural networks", *Neural Processing Letters*, vol. 53, no. 6, pp. 4693–4710, 2021.

[4] K. Wang, X. Hao, J. Wang, and B. Deng, "Comparison and selection of spike encoding algorithms for snn on fpga", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 17, no. 1, pp. 129–141, 2023.

[5] Q. Chen, D. Li, T. Tao, H. Ma, and E. Li, "Temporal neural encoding methods for spiking neural networks", in *2022 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC)*, IEEE, 2022, pp. 88–90.

[6] B. Petro, N. Kasabov, and R. M. Kiss, "Selection and optimization of temporal spike encoding methods for spiking neural networks", *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 358–370, 2019.

[7] S. Y. A. Yarga, J. Rouat, and S. Wood, "Efficient spike encoding algorithms for neuromorphic speech recognition", in *Proceedings of the International Conference on Neuromorphic Systems 2022*, 2022, pp. 1–8.

[8] A. Vasilache, *Alex-Vasilache/Spike-Encoding*, [Online]. Available: https://github.com/Alex-Vasilache/Spike-Encoding, Feb. 2025.

[9] A. Webb, S. Davies, and D. Lester, "Spiking neural pid controllers", in *Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III 18*, Springer, 2011, pp. 259–267.

[10] B. Schrauwen and J. Van Campenhout, "Bsa, a fast and accurate spike train encoding scheme", in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, IEEE, vol. 4, 2003, pp. 2825–2830.

[11] N. Kasabov *et al.*, "Evolving spatio-temporal data machines based on the neucube neuromorphic framework: Design methodology and selected applications", *Neural Networks*, vol. 78, pp. 1–14, 2016.

[12] A. Arriandiaga, E. Portillo, J. I. Espinosa-Ramos, and N. K. Kasabov, "Pulsewidth modulation-based algorithm for spike phase encoding and decoding of time-dependent analog data", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 3920–3931, 2019.

[13] J. Dupeyroux, S. Stroobants, and G. C. De Croon, "A toolbox for neuromorphic perception in robotics", in *2022 8th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*, IEEE, 2022, pp. 1–7.

[14] A. R. Gollahalli, *Github.com/akshaybabloo/Spikes*, original-date: 2016-10-05, Sep. 2024.

[15] N. Kasabov *et al.*, "Evolving spatio-temporal data machines based on the neucube neuromorphic framework: Design methodology and selected applications", *Neural Networks*, vol. 78, pp. 1–14, 2016.

[16] B. Schrauwen and J. Van Campenhout, "Bsa, a fast and accurate spike train encoding scheme", in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, IEEE, vol. 4, 2003, pp. 2825–2830.

[17] A. Arriandiaga, E. Portillo, J. I. Espinosa-Ramos, and N. K. Kasabov, "Pulsewidth modulation-based algorithm for spike phase encoding and decoding of time-dependent analog data", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 3920–3931, 2019.

[18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework", in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[19] Y. Ozaki, Y. Tanigaki, S. Watanabe, and M. Onishi, "Multiobjective tree-structured parzen estimator for computationally expensive optimization problems", in *Proceedings of the 2020 genetic and evolutionary computation conference*, 2020, pp. 533–541.

[20] M. Towers *et al.*, "Gymnasium: A standard interface for reinforcement learning environments", *arXiv preprint arXiv:2407.17032*, 2024.

Security Risk Assessment System Based on the Similarity to Victims

Masahito Kumazaki
Graduate School of Informatics
Kyoto University
Kyoto, Japan
e-mail: kumazaki@inet.\
media.kyoto-u.ac.jp

Hirokazu Hasegawa
Center for Strategic
Cyber Resilience R & D
National Institute of Informatics
Tokyo, Japan
e-mail: hasegawa@nii.ac.jp

Hiroki Takakura
Center for Strategic
Cyber Resilience R & D
National Institute of Informatics
Tokyo, Japan
e-mail: takakura@nii.ac.jp

***Abstract*—In the current situation where the damage caused by targeted attacks is becoming more serious, it is important to minimize their damage through early detection and response. However, there are problems with speed and coverage in the investigation methods based on suspicious IP addresses and Indicators of Compromises(IoC), which are common responses. Therefore, we propose a security risk assessment system based on the similarity to victims. This system uses elements other than IP addresses and IoCs, such as used software and logged-in users, and assesses the possibility of intrusion based on similarity to the victim. In this paper, we show the effectiveness and potential of this system by evaluating it using a prototype.**

Keywords-cyber security, risk assessment, incident response

I. INTRODUCTION

In the current situation where the damage caused by targeted attacks is becoming more serious, it is important to minimize their damage by improving detection and response measures. In particular, many researchers have been discussing various proposals for detection methods aimed at minimizing damage.

In general, responders collect attack traces such as suspicious IP addresses from the victim and devices on the attack route, and also collects Indicator of compromises (IoC) from the provider on the internet. Based on the information collected in these ways, they estimate the scope of the intrusion and the attack technique, and attempts to minimize the damage. However, the following problems exist with this method.

- Coverage of suspicious IP addresses
- Reliability of IoC providers
- Time required for information collection

Therefore, we propose a system for assessing the security risk of neighboring terminals using the similarity to the victim. In this paper, "security risk" refers to the possibility that other terminals will be attacked using the same techniques as the victim. The system uses the user's input to collect information on victim and neighboring terminals from various sources and assess the security risk of neighboring terminals. The system can assess the scope of the intrusion earlier than existing IP address/IoC-based methods, making it possible to minimize the damage. In addition, even in situations where little information is available about an attack, such as a zero-day attack, it is possible to predict the occurrence of damage by using a device with a configuration similar to the device that was first affected.

The outline of this paper is as follows. In Section 2, we introduce related works from the perspectives of security risk assessment and attack detection, and point out the issues that exist in them. Section 3 describes the system we propose, and Section 4 performs a simple evaluation using a prototype. We discuss the improvements to the system in Section 5, and we present our conclusions and future works in Section 6.

II. RELATED WORK

In existing studies on security risk assessment, researchers have assessed risk from a variety of perspectives. The first perspective is risk assessment focusing on important assets within an organization. Kumar et al. proposed a framework called integrated Cybersecurity Risk Management (i-CSRM) that identifies important infrastructures, assesses the risk of vulnerabilities in those infrastructures, and evaluates the safety of current operations[1]. From another perspective, there is also study on risk assessment that focuses on the costs required to implement security measures. Lee proposed a framework that realizes the optimal security improvement procedure by estimating the existing threats and economic losses used these, as well as the necessary costs for countermeasures, based on the situation inside and outside the organization[2]. There are also studies that focus on physical and human factors. Ganin et al. pointed out that existing risk assessment methods based on threats, vulnerabilities and consequences[3][4] do not cover physical or social vulnerabilities, and proposed a framework using multi criteria decision analysis (MCDA) to cover them[5]. There is also study that focuses on the possibility of a intrusion to the terminal, which is the same perspective as ours, such as Sugimoto et al.[6]. They assess the security risk of each device from three points of view: accessibility to the device, the number of routes, and the scope of the intrusion after the attack, in order to determine the priority of dealing with vulnerabilities. These studies function as a pre-emptive measure against attacks, and do not discuss the security risk in response to detection. In terms of post-detection response, it is important to minimize the scope of the intrusion and the damage it causes, so it is necessary to evaluate the risk from the perspective of being able to achieve this and in a short period of time.

Other related study of proposed is attack detection. Since the proposed system is used repeatedly from the initial stage of attack response, it is assumed that there will be a conflict with the timing of use with existing attack detection technology. Some of these studies includes improving detection accuracy using intrusion detection systems (IDS)[7][8] and security

 ISBN: 978-1-68558-278-4

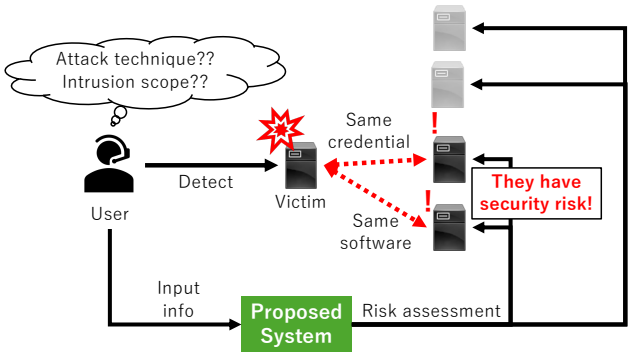


Figure 1. Proposed system's concept

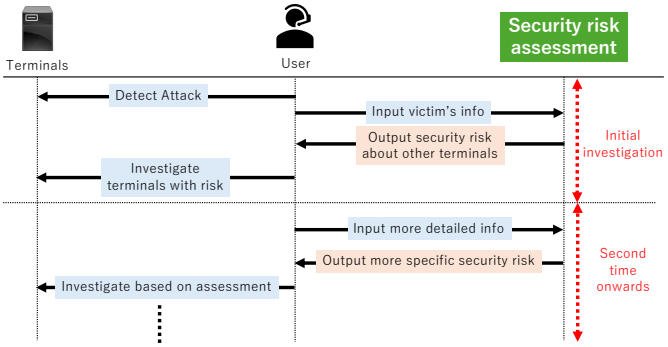


Figure 2. Proposed system's usage flow

information and event management (SIEM)[9][10]. Studies using these systems may show effectiveness in detection, but this does not necessarily mean they are effective in response. Mohsenabad et al. showed that it is possible to improve the detection accuracy of IDS by selecting feature values used in machine learning based on various algorithms[7]. However, the detection results of IDS at this time are limited to information such as victim and attack techniques, so additional investigation is required to learn the details of the attack. In such cases, users can learn about the security risks of the neighboring terminals by entering the IDS's result into the proposed system, and they can narrow down the scope of their investigation based on this information. In this way, we consider that our system does not conflict with these attack detection technologies, but rather coexists with them.

III. Proposed System

A. Outline

We propose a risk assessment system for neighboring terminals based on the similarity of victims. The Figure 1 shows the concept of proposed system. We assume that users of this system are people who are familiar with networks and security, such as those who respond to security incidents. If the user detects an attack, they will want to know the details of the attack and the scope of the intrusion, but this takes time and effort. In such cases, the user inputs the victim's information into the proposed system, and the system assesses the security risk of the neighboring terminals based on similarities to the victim (e.g. same users, same software, etc.). In this way, the system supports the user's response by narrowing down the scope of the investigation and providing information about vulnerabilities that may exist in the terminals.

The Figure 2 shows system usage. This system is designed to support users in their repeated use of the system from the initial stage of an investigation. In the initial investigation, it supports narrowing down the terminals that have security risk and used attack techniques, based on the limited information. In the second time onwards, the users can input more detailed info, so the system also outputs more accurate information on the scope of intrusion, attack techniques and vulnerability information to them.

B. Assumption

The proposed system collects various information about the terminal in order to evaluate similarity. So we assume that devices providing services such as asset management, firewalls, and file servers exist with in the range accessible from this system.

In addition, this system is designed to be used in an attack response. Of course, it is best to be aware of the security risk of all terminals in advance. However, given the large number of terminals and the easy introduction of terminals such as mobile devices, this is unrealistic. Therefore, this system is designed to be used in an attack response and to be useful for investigating terminals and vulnerabilities that have not been identified at that time.

In this proposal, the user can specify the range of assessments by defining a neighboring terminal as a terminal with a number of network hops from the victim that is less than or equal to a threshold. If the user use the system for a specific segment (e.g. server segment), they can set the threshold to 1. If the user expect an intrusion into another segment, they can adjust the threshold accordingly, and apply the system to any range them want.

When the user confirms a security incident, the system assesses the security risk of neighboring terminals through user input. The system focuses on the following attack stages and outputs the attack techniques and vulnerability risks associated with them.

- External intrusion that has occurred
- Lateral movement from the victim

However, as this paper is initial prposal, we will only discuss external intrusion. Therefore, this paper does not discuss lateral movement such as intrusion into other services from the victim.

C. System Architecture

The risk assessment system consists of four modules as shown in Figure 3. It performs input and output with users in the dialogue module, and the other modules collect information and assesses security risks in response to the input in the dialogue module.

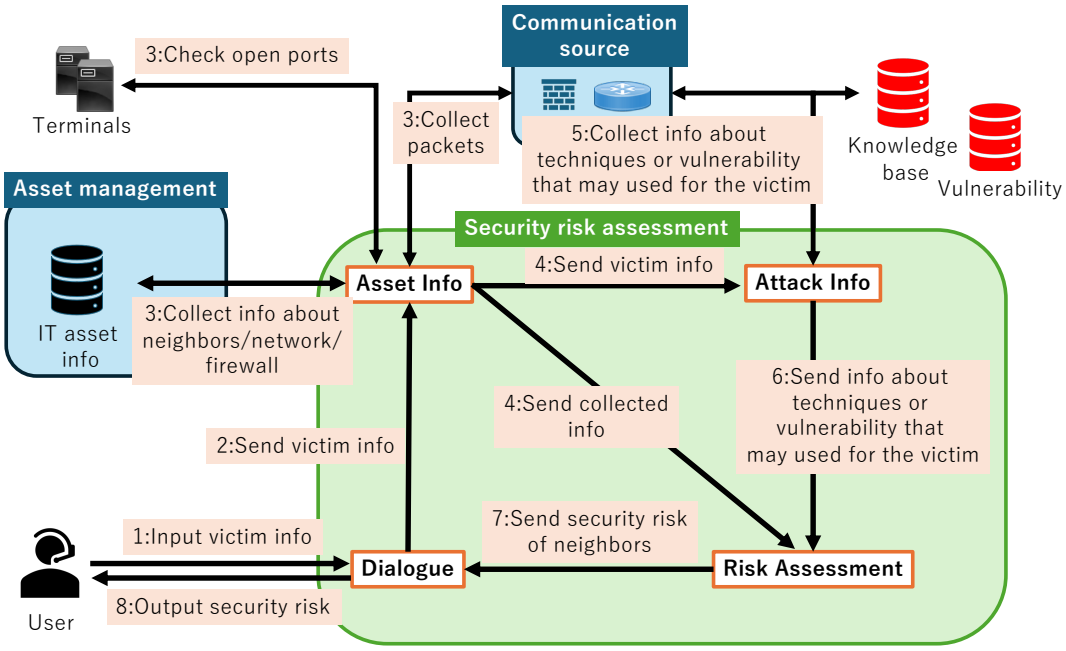


Figure 3. Proposed system's architecture

TABLE I. INPUT BY USER

Contents		Required or Optional
Threshold for neighbors		Required
Occurrence time		Optional
Victim info	IP address	Required
	Role	Optional
	Admin account	Optional
Attacked software/hardware		Optional
Used technique/vulnerability		Optional

1) Dialogue Module: The Dialogue Module receives input about the victim information from users and outputs the results of the assessment of security risks to users. Table I shows the victim information that the user enters. With regard to the input, the threshold for neighbors and the victim IP address are required, and the rest are optional. If a user inputs optional information such as attacked software or used techniques, the module attaches a "Used" tag. Regarding the "Used" tag, the Asset/Attack Info Module collects information only for the software/techniques with that tag.

The module sends these information to the Asset Info module.

2) Asset Info Module: The Asset Info module collects asset information with in an organization. The Figure 4 shows functions and flow of operation. The module works in the following way.

- Receive the victim's information from the victim
- Collect information about the victim and network around it from the asset management system
- Based on the information about the network around the victim, determines which devices are neighboring devices
- Collect following information

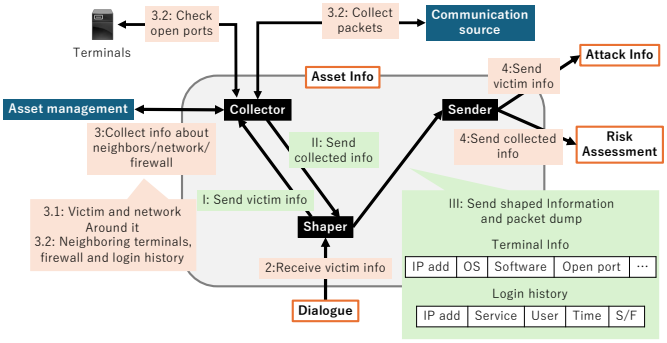


Figure 4. Functions and operation flow of Asset Info module

  - From asset management system: Neighboring terminals, firewall rule and login history of each service
  - From Terminals: Open ports
  - From communication source: Mirroring packets
- Send shaped information to each module

3) Attack Info Module: The Attack Info module collects information about the techniques and vulnerabilities that could be used for the victim. The module receives information about the victim from the Asset Info module. This information includes the victim's operating system and software, service login history on the victim, etc.. Based on the information, the module collects the security holes that exist in the victim by the following ways.

- Based on the open ports, services and login history, the module collects the related attack techniques from vulnerability

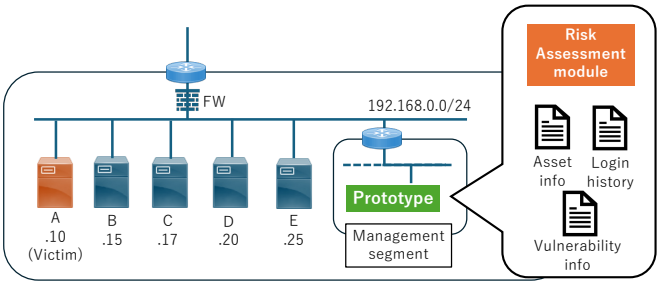


Figure 5. Assumed network in evaluation

knowledge base such as MITRE ATT&CK[1]

- Based on the OS and software, the module collects vulnerabilities from vulnerability databases such as the National Vulnerability Database (NVD)[2]

The module shapes the collected information in a form that includes IDs and their requirements, and sends it to Risk Assessment module. As we pointed out in Section 1, there is very little publicly available information on zero-day attacks, so information based on the NVD is not mandatory.

4) Risk Assessment Module: This module assesses the security risk of neighboring terminals based on the information received from each module. The information sent from Asset Info module contains data about neighboring terminals and victims, and the information sent from Attack Info module contains information about each attack that could occur on the victim. The module assesses the risk of each attack for each neighboring terminal. If the user enters the attack technique or vulnerability used in the Dialogue module, the module also assesses its risk.

In this paper, security risks are classified into three levels: *high*, *medium*, and *low*. For each technique or vulnerability, the module assesses the security risk for each terminal as follows.

- The terminal satisfies the requirements for the method or vulnerability: *high*
- There are similarities with the victim in multiple contexts (e.g. software used, login history, etc.): *medium*
- Otherwise: *low*.

Finally, the module sends the results of the assessments to the Dialogue Module.

IV. Evalution

In order to evaluate the proposed system, we implemented and tested a prototype of the Risk Assessment module. Since Asset Info and Attack Info modules have not been implemented yet, their outputs were prepared in advance and provided as inputs for the Risk Assessment module.

A. Evaluation Method

The figure 5 shows the assumed network in the evaluation. There were 5 terminals in the assumed network. Terminal A was the first victim, and Terminals B, C, D, and E were on the same network segment, i.e., a hop count of 0.

[1]https://attack.mitre.org/
[2]https://nvd.nist.gov/

TABLE II. Asset Info in evaluation

Terminal	IP address	OS	Software	Open ports
A	192.168.0.10	Windows	WordPress, 5.8	22,80
			OpenSSH, 9.7	
B	192.168.0.15	Windows	WordPress, 6.0	22,80
			OpenSSH, 9.9	
C	192.168.0.17	Windows	WordPress, 5.9	80
D	192.168.0.20	Windows	OpenSSH, 9.7	22
E	192.168.0.25	Windows		22,80

TABLE III. Login history (Only terminals A and B)

Terminal	Service	User	Time	S/F
192.168.0.10	OpenSSH	Alice	2025/2/3 10:23:51	S
		Hack	2025/2/4 00:31:20	F
		Hack	2025/2/4 00:31:21	F
		Hack	2025/2/4 00:31:21	F
	WordPress	Alice	2025/2/4 10:25:14	S
		Bob	2025/2/4 13:08:05	S
192.168.0.15	OpenSSH	Bob	2025/2/3 13:41:35	S
		Black	2025/2/4 01:20:54	F
		Black	2025/2/4 01:20:54	F
		Black	2025/2/4 01:20:55	F
	WordPress	Bob	2025/2/4 13:08:05	S

The prototype of Risk Assessment module was on the terminal in the management segment. As we explained, the information sent from the Asset Info and Attack Info modules is prepared as JSON files in advance. These json files also existed on the terminal in the management segment. The Asset Info file contained information about A, B, C, D, and E, the Login History file contained login histories for the services provided on each terminal, and the Vulnerability Info file contained information about the vulnerabilities that may exist on A. The contents of each file are shown in Tables II, III and IV respectively. Based on the input from these files, the prototype performed a risk assessment using the method shown in Section III-C4, and output the result as a json file.

B. Evaluation Result and Findings

The output result is shown in Figure 6 and Table V. Regarding the vulnerability CVE-2024-6387, the prototype of Risk Assessment module is thought to assess the risk of each terminal for the following reasons.

- B: This terminal used the same software as A and the login history was similar, so the risk was *medium*.
- C: This terminal didn't use the OpenSSH that was a requirement, so the risk was *low*.
- D: This terminal satisfied requirements (used the OpenSSH and this version was less than 9.8), so the risk was *high*.
- E: It was not known what software was used on this terminal. However, from the information about open ports and softwares of A, B and D, the prototype estimated that E

TABLE IV. Attack Info in evaluation

ID	Requirement	
CVE-2024-6387	Software: OpenSSH	Version: <9.8
CVE-2022-21661	Software: WordPress	Version: <5.8.3

TABLE V. SECURITY RISK ASSESSMENT RESULTS AND REASON ASSUMPTIONS

Vulnerability	CVE-2024-6387			
Terminal	B	C	D	E
Risk	*medium*	*low*	*high*	*high*
Reason	Similarity in software and login status	No similarity	Satisfy requirement	Satisfy requirement (possibly)
Vulnerability	CVE-2022-21661			
Terminal	B	C	D	E
Risk	*low*	*low*	*low*	*high*
Reason	No similarity	No similarity	No similarity	Satisfy requirement (possibly)

```
{
    "CVE-2024-6387":[
        {"192.168.0.15":"medium"},
        {"192.168.0.17":"low"},
        {"192.168.0.20":"high"},
        {"192.168.0.25":"high"}
    ],
    "CVE-2022-21661":[
        {"192.168.0.15":"low"},
        {"192.168.0.17":"low"},
        {"192.168.0.20":"low"},
        {"192.168.0.25":"high"}
    ]
}
```


Figure 6. Output file after evaluation

used OpenSSH. Because the version was unknown, there was a possibility that E satisfied the requirements for vulnerability. So the risk was *high*.

Regarding the vulnerability CVE-2022-21661, the prototype is thought to assess the risk of each terminal for the following reasons.

- D: This terminal didn't use the WordPress that was a requirement, so the risk was *low*.
- E: From the information about open ports and softwares of A, B and C, the prototype estimated that E used WordPress. Because the version was unknown, there was a possibility that E satisfied the requirements for vulnerability. So the risk was *high*.
- Other terminals: They used WordPress but these version were more than 5.8.3. B and D didn't satisfied requirements and the similarities only existed at the OS and software, so the risk was *low*.

As a result, it was confirmed that proposed the system can assess the risk of attacks with clear requirements, such as vulnerabilities that depend on specific softwares and versions. Even when the software used on the terminal is unknown, the system was able to assess the risk by estimating from the service operating status of other terminals. It's thought that this is because there were many terminals providing similar services on the same port in this evaluation. Even when there are few terminals opening the same port, it is necessary to estimate the provided services and the software of the terminal. As a future work, we are planning to make use of well-known ports and etc. to enable such estimation.

In addition, the current prototype only assesses security risks based on user login history, service operation status, etc., for vulnerabilities and attack techniques that do not have clear requirements. In this case, the prototype can only assess risks as *medium* or *low*. As a future work, we will consider improving the prototype so that it can assess the risk of such attacks by adding up the similarity of each element.

V. DISCUSSION

This system evaluation showed that the proposed system is effective in assessing risk from external attacks. However, we have not conducted a quantitative evaluation and have not been able to show how effective the proposed system compared to previous studies. There are two problems in comparing the proposed system with previous studies.

- The previous studies focused on risk assessment before attacks. The proposed system cannot be simply evaluated because it is assumed to be used while responding to attacks.
- The previous studies on attack detection do not cover the steps after the detection of an attack. This system is expected to be effective in response after detection.

Based on these issues, we consider the following comparisons.

- Risk assessment: Evaluation of how close to the accuracy of previous studies in the short time available during response to an attack.
- Attack detection: Comparison of the time required to respond to an attack using the proposed system with that of the previous studies.

In addition, we define a neighboring terminal as a terminal with a number of network hops from the victim that is less than or equal to a threshold, in order to specify the range of assessments by the user. However, in actual network configurations, there are many cases where users don't know how many hops there are from the victim in the range they want to check. On the other hand, the more huge the network becomes, the more difficult it becomes to investigate everything within the network. In future work, we will improve the definition of neighboring terminals so that users can more easily set the range they want to search.

Furthermore, the proposed system omits the aspect of ease of access to the equipment in assessing the security risk. Since this paper only focuses on external attacks, we assumed that all targets are accessible from the outside (i.e., easily accessible). When this system assesses the risk of lateral movement in

internal network, ease of access will be an important metric. (e.g.: The system evaluates only the risk of lateral movement for terminals that can only be accessed from the internal network. This evaluates both external attack and lateral movement risks for terminals that can be accessed externally.)

VI. Conclusion and Future Work

In this paper, we propose a system for assessing the security risk of neighboring terminals using the similarity to the victim. This system can assess the scope of the intrusion from the beginning of the investigation, and it is possible to minimize the damage compared to existing IP address and IoC-based methods. In addition, we evaluated the prototype and demonstrated its potential and effectiveness. As a result, when there were many terminals providing similar services on the same port, it was confirmed that proposed system can assess the risk of attacks with clear requirements. However, there are some issues on the following.

- When there are few terminals opening the same port, the system can't accurately assess the risk for terminals that lack information.
- The system cannot assume that the risk is high for vulnerabilities and attack techniques that do not have clear conditions.

In order to solve these issues, we will work on the following for the Risk Assessment module.

- Use well-known ports, etc. to estimate the services provided by the terminal.
- Improve the prototype so that it can accurately assess the risk of vulnerabilities and techniques that don't have clear requirements.

In addition, this paper only focuses on external attacks and does not consider lateral movement. There are two possible routes of compromise to the terminal in a real attack: external attack and lateral movement. This system should be able to handle both of these attack tactics. In addition to the topics in the Discussion section, there are the following issues.

- Comparison with previous studies in terms of accuracy and time required
- Improve the definition of neighboring terminals so that users can more easily set the range they want to check
- Implement other modules and conduct evaluations in a form that is more suited to the system architecture
- Development of a risk assessment that includes lateral movement

In future work, we will solve the above issues.

Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Number JP24K14959.

References

[1] N. Kumar, V. Poonia, B.B. Gupta and M.K. Goyal, "A novel framework for risk assessment and resilience of critical infrastructure towards climate change," Technological Forecasting and Social Change, Vol. 165, 2021.

[2] I. Lee, "Cybersecurity: Risk management framework and investment cost analysis," Business Horizons, Vol. 64, No. 5, pp. 659-671, 2021.

[3] R. A. Caralli, J. F. Stevens, L. R. Young, and W.R. Wilson, "Introducing octave allegro: Improving the information security risk assessment process," Hansom AFB, MA, 2007.

[4] A. Ashok, M. Govindarasu, "Cyber-physical risk modeling and mitigation for the smart grid using a game-theoretic approach," 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), pp. 1-5,2015

[5] A. A. Ganin, P. Quach, M. Panwar, Z.A. Collier, J.M. Keisler, D. Marchese, and I. Linkov, "Multicriteria decision framework for cybersecurity risk assessment and management," Risk Analysis, Vol. 40, No. 1, pp. 183-199, 2020

[6] A. Sugimoto, Y. Isobe and H. Nakakoji, "Risk Assessment Based on Intrusion Routes of Cyber Attacks," Journal of Information Processing (JIP), Vol. 57, No. 9, pp. 2077-2087, 2016. (In Japanese)

[7] H. N. Mohsenabad and M. A. Tut, "Optimizing cybersecurity attack detection in computer networks: A comparative analysis of bio-inspired optimization algorithms using the CSE-CIC-IDS 2018 dataset," Applied Sciences, Vol. 14, No. 3, 2024.

[8] A. Raza, K. Munir, M. S. Almutairi, and R. Sehar, "Novel class probability features for optimizing network attack detection with machine learning," IEEE Access, 2023.

[9] R. Marri, S. Varanasi, and S. V. K. Chaitanya, "Integrating Next-Generation SIEM with Data Lakes and AI: Advancing Threat Detection and Response," Journal of Artificial Intelligence General science (JAIGS), Vol. 3, No. 1, pp. 446-465, 2024.

[10] A. R. Muhammad, P. Sukarno and A. A. Wardana, "Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning," Procedia Computer Science, Vol. 217, pp. 1406-1415, 2023

GLACI: Arbitrary Code Instrumentation Tool for OpenGL

Shotaro Tsuboi
Graduate School of Informatics,
Nagoya University
Nagoya, Japan
e-mail: s_tsuboi@ertl.jp

Yixiao Li
Graduate School of Informatics,
Nagoya University
Nagoya, Japan
e-mail: liyixiao@ertl.jp

Yutaka Matsubara
Graduate School of Informatics,
Nagoya University
Nagoya, Japan
e-mail: yutaka@ertl.jp

Hiroaki Takada
Graduate School of Informatics,
Nagoya University
Nagoya, Japan
e-mail: hiro@ertl.jp

***Abstract*—Modern embedded systems usually run multiple graphics applications concurrently, making efficient Graphics Processing Unit (GPU) resource management a critical challenge. To address this need, we present Arbitrary Code Instrumentation tool for OpenGL (GLACI), a flexible tool that enables transparent interception of Open Graphics Library (OpenGL) Application Programming Interface (API) calls to instrument arbitrary code without modifying the application or the graphics stack. GLACI-based module can cooperate with the GPU resource manager to support advanced features such as real-time Frames Per Second (FPS) monitoring, Quality of Service (QoS) based resource limiting and on-demand tracing. A prototype is created and evaluated on Intel and NVIDIA platforms to show the portability and usefulness of GLACI. By offering a unified, hardware-independent and lightweight solution, GLACI broadens the scope of GPU resource control and provides a practical foundation for both development and production environments.**

Keywords-embedded systems; OpenGL; code instrumentation, GPU resource management.

I. INTRODUCTION

In modern embedded systems, multiple graphics applications with varying reliability and requirements can share a single Graphics Processing Unit (GPU). For example, in automotive systems, the GPU is used for displaying the speedometer, navigation In-Vehicle Infotainment (IVI) displays and other third-party applications concurrently. Currently, Open Graphics Library (OpenGL) [1] is the most commonly supported and widely used graphics Application Programming Interface (API) for such applications.

However, due to the lack of tracing and resource management techniques for production environment, it is difficult to debug and develop such systems with necessary Quality of Service (QoS) satisfied. For instance, if a third-party application installed by the user consumes too much GPU resource, it can cause interference with the critical services (e.g., speedometer). A reliable system should be able to detect such kind of performance issues, record useful traces for analysis, and adjust the GPU resource allocation according to the QoS settings while it is running.

Most of the previous studies about scheduling GPU-sharing tasks focus on the scope of General-Purpose computing on Graphics Processing Unit (GPGPU) applications rather than the graphics applications [2]. These studies typically assume that the source code of the GPGPU tasks is available and can be modified to assist the GPU resource management. Meanwhile, many graphics applications, especially the third-party ones, are only available in binary executable files. The programming models of GPGPU tasks (computation-intensive functions offloading) and graphics tasks (complex rendering pipeline) also have a significant difference. Therefore, it is difficult to reuse these techniques for GPGPU tasks on graphics applications.

Some previous studies on improving the QoS of graphics applications have been proposed, but these methods face many challenges in terms of practicality. A common approach is to override the default scheduler with a QoS-aware one by modifying the kernel-space GPU driver [3][4], which has poor portability and maintainability since it depends on a specific GPU model and kernel version. It is also possible to manage the GPU resource by extending the implementation of OpenGL library [5][6] but many popular GPU vendors, including NVIDIA, only provide unmodifiable proprietary OpenGL libraries. Therefore, the usefulness of these studies is highly restricted in real-world systems.

In this paper, we propose Arbitrary Code Instrumentation tool for OpenGL (GLACI), an open-source tool which can assist the system developer to overcome the above limitations. With GLACI, hardware-independent modules for OpenGL API instrumentation can be effortlessly implemented. It allows us to dynamically trace and change the behavior of graphics applications, by adding custom code around OpenGL API calls, without acquiring and modifying any source code of the application and OpenGL library. If an application is executed with GLACI-based module loaded, the GPU resource manager can attach to it for monitoring and controlling.

The main contributions are listed as follows.

- GLACI, a generic tool for implementing hardware-independent OpenGL API instrumentation modules, which can change the behavior of application and library without modifying any source code, is proposed.

- A prototype including examples of GLACI-based module and GPU resource manager is created to show the usefulness of our method.
- Two representative platforms, based on Intel and NVIDIA, are used to evaluate the functionality and overhead.
- The source code of GLACI and the prototype is publicly available for reproducing and extension [7].

The rest of the paper is organized as follows. Section II discusses and compares previous studies with similar goals and methods. The details of GLACI are explained in Section III. Section IV uses a prototype of GLACI-based module and GPU resource manager to show the usefulness. Section V assesses our method by evaluating the prototype on Intel and NVIDIA platforms. Finally, the research is concluded in Section VI.

II. Related Work

A. GPU Resource Management

Previous studies have shown that it is possible to limit GPU bandwidth or guarantee Frames Per Second (FPS) for each application by inserting some processing into the graphics stack.

In the FPS control methods using execution time prediction [5][6], OpenGL API calls are monitored to obtain parameters such as the number of vertices and fragments to predict execution time which is used for GPU task scheduling. This approach modifies the source code of OpenGL library to acquire parameters, and the low-level GPU driver to apply scheduling policies.

In the QoS-based controlling methods [3][4], graphics APIs are modified to acquire QoS metrics. These methods also modify GPU drivers to apply scheduling policies. Some studies replace the low-level GPU task scheduler with a custom one by modifying the GPU driver [2][8].

These existing control methods lack generality for different platforms and GPU drivers, and require re-implementations for various environments. The modifications to the target applications are also needed in some methods, which makes them not feasible for third-party applications without source code. Meanwhile, GLACI focuses on supporting the resource management on the high-level hardware-independent OpenGL API layer as possible, rather than modifying the source code of existing graphics stack. If necessary, GLACI-based module can also cooperate with the GPU driver for fine-grained control.

B. OpenGL Tracing Tools

Several tracing tools for OpenGL API have been proposed to support the analysis of various metrics of the rendering commands and procedures. There are mainly two types of such tools: vendor-independent tools, and vendor-specific tools.

RenderDoc [9] and Apitrace [10] are two representative vendor-independent tools for debugging, tracing, and performance analysis of multiple graphics APIs, including OpenGL. These tools always hook every single OpenGL API call when the application is running, in order to produce a detailed trace file with all inputs, outputs and states recorded. Users can use the trace file to replay the rendering process of a frame for detailed behavior and performance analysis.

GPU vendors also provide tools to visualize rendering processes on their GPUs, such as Intel GPA [11] and NVIDIA Nsight Graphics [12]. These tools offer detailed views of processing at the GPU core level and can be used for low-level optimization. However, these vendor-specific tools only work on specific platforms, and most of them are proprietary software without source code provided, which makes it difficult to extend their functionality.

These existing tracing tools have fixed tracing scope and are designed for the test environment. For example, the tracer cannot be dynamically switched on and off when the application is running. It is also not possible to specify what information should be obtained to meet different requirements. Therefore, using these tools for tracing all applications in the production environment will cost a huge amount of resource. Further, since the tracing results can not be accessed from the GPU resource manager in real time, they are only useful for the postmortem analysis.

GLACI is not only capable of implementing the fixed-purpose tracing feature equivalent to the vendor-independent tools, but can also expose interfaces to communicate with the GPU resource manager to support advanced features like live performance monitoring, on-demand tracing and resource limiting. Therefore, unlike other tools, GLACI can be used in both testing and production environments.

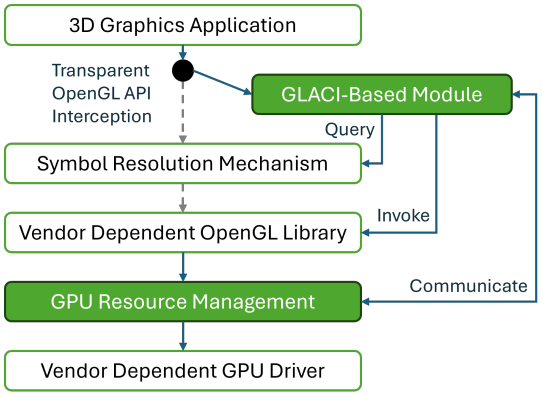


Figure 1. The overview of common functions in a GLACI-based module.

III. Proposed Method

A. Overview

GLACI is a hardware-independent tool that allows developers to effortlessly create modules capable of extending the functionality of existing graphics stack by instrumenting OpenGL API calls. Figure 1 shows an overview of how a GLACI-based module typically works in the graphics stack. The module can transparently intercept the OpenGL API calls and communicate with GPU resource manager, without modifying the source code of graphics application, OpenGL library and GPU driver.

A graphics application must be able to run on different versions of graphics stack without rebuilding the software, because the GPU vendor frequently updates the OpenGL library and GPU driver for optimization and bug fixing. Some systems even require the same application to run on GPUs from different vendors (e.g., the diversified GPU solutions for Android devices). To meet this portability requirement, OpenGL has introduced an advanced symbol resolution mechanism, instead of naively linking the application to some specific libraries.

When a GLACI-based module is loaded, it will use the symbol resolution mechanism to query the symbol addresses of all OpenGL APIs at first, and then mimic and override that mechanism to redirect the API calls to automatically generated wrapper functions with custom code instrumented.

If a system runs multiple applications with different QoS levels or priorities, there is usually a GPU resource manager located between the OpenGL library and the GPU driver to monitor and properly schedule the GPU usage of each application. However, a GPU resource manager can only attach to the applications with necessary interfaces for communication included, which means most of the third-party and proprietary applications are out of scope. A main benefit GLACI offers is that it can insert such interfaces to any OpenGL application to make it controllable from the GPU resource manager.

Figure 2 shows the flow of how GLACI will process a user-defined module project to build a loadable module binary. OpenGL is a very complex API specification with many different versions (e.g., GL 1.0 to 4.6, ES 1.0 to 3.2, SC 1.0 to 2.0) and additional extensions (e.g., ARB, GLX). Further, although OpenGL is a platform-independent specification in general, vendors also have added some special features in their proprietary library implementation. In the field of embedded systems, target boards usually support some specific versions of OpenGL (e.g., the popular Raspberry Pi 4 only runs GL 2.1 and ES 3.1). Therefore, it is impractical for us to assume the system uses and only uses the latest OpenGL version and vendor-independent features. Khronos Group has released the official Extensible Markup Language (XML) definition files of OpenGL API specification, including all versions and optional features. To address the challenge above, GLACI can load these XML files to build modules for a specific target system.

The user-defined module project consists of an instrumentation script in Python and some extra source files in C++. The instrumentation script defines the rules to instrument OpenGL API calls. The extra source files include the code with no need to be dynamically generated (e.g., data structure definitions and algorithm implementations). The GLACI core will follow the instrumentation script to generate a single source file for the module with OpenGL API wrapper functions and extra source code included. Finally, a shared library binary of the module will be built from the generated source file. We can use the `LD_PRELOAD` environment variable to start graphics application with the module loaded.

It should be noted that while the GLACI module is written in C++, it does not impose restrictions related to the programming languages of target graphics applications. Since the method operates at the binary interface level, applications developed in any language capable of invoking OpenGL APIs from the instrumented library can seamlessly benefit from GLACI without additional adaptation efforts.

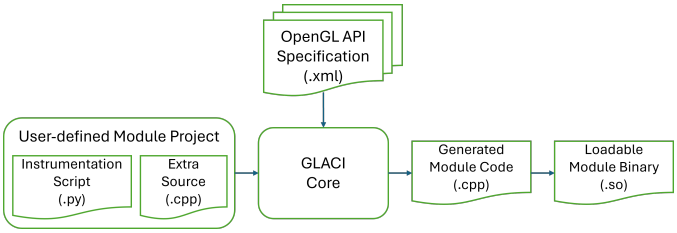


Figure 2. The process flow of how GLACI builds a module project.

B. Transparent OpenGL API Interception

GLACI-based module is loaded with the `LD_PRELOAD` feature, which allows us to override existing functions in the standard shared libraries of the graphics stack. However, to achieve portability and compatibility, OpenGL applications are not directly linked with a specific graphics stack. Instead, an advanced symbol resolution mechanism including the following three methods is provided for the applications to find the symbols at runtime.

- **Runtime linker:** In some systems, especially those using Mesa 3D graphics stack, a part of OpenGL API symbols (e.g., GLX extension for X11 window system) may be implemented in a shared library with stable Application Binary Interface (ABI). Therefore those shared libraries are directly linked to the application, and their symbols are resolved by the standard runtime linker.
- **dlopen/dlsym dynamic loader:** The graphics stack calls `dlopen` function to load OpenGL libraries by explicitly specifying the file names according to the actual running platform. It will then call `dlsym` function to dynamically search the symbol addresses of OpenGL API functions in these loaded libraries.
- **OpenGL *GetProc* functions:** OpenGL API specification also defines functions (e.g., `glXGetProcAddress`) to obtain symbol address by API name. Unlike the above two methods are provided by and dependent on the OS, this method is platform-independent.

To intercept OpenGL API transparently on various platforms and graphics stacks, GLACI must support all these methods to completely override the original symbol resolution mechanism.

To support the runtime linker method, GLACI will generate wrapper functions with the same prototypes for all OpenGL API functions, so the linker will always return the symbol addresses in our module instead of the original ones. Figure 3 shows an example of the glXSwapBuffers API.

To properly invoke the original API implementation from the generated wrapper function, GLACI shall initialize the

```
static typeof(glXSwapBuffers) *
    original_glXSwapBuffers = NULL;

void glXSwapBuffers(Display *dpy, GLXDrawable
    drawable) {
  ... /* instrumented code before glXSwapBuffers */
  original_glXSwapBuffers(dpy, drawable);
  ... /* instrumented code after glXSwapBuffers */
}
```

Figure 3. Example of generated glXSwapBuffers wrapper function.

```
__attribute__((constructor))
void load_original_functions(){
    ...
    original_glXSwapBuffers =
        dlsym(RTLD_NEXT, "glXSwapBuffers");
    ...
}
```

Figure 4. Example of initializing original_glXSwapBuffers.

function pointers to correct symbol addresses before the application starts to call any OpenGL API function as shown in Figure 4.

To support the dlopen/dlsym dynamic loader method, GLACI must solve the following issues.

- All the generated wrapper functions for OpenGL API are ignored because the `dlsym` function will only search symbols in the library file dynamically loaded by the `dlopen` function.
- The method of initializing original function pointers at startup does not work since the symbol addresses are unknown until the graphics stack calls and obtain the return value from the `dlsym` function.

GLACI addresses these issues by overriding the `dlsym` function with a modified version as shown in Figure 5. It will call the original `dlsym` function at first to get the symbol address. If the symbol is not an OpenGL API function, it will just return the address obtained. For OpenGL API symbols, the obtained symbol address will be stored in the original function pointer, and the address of corresponding wrapper function will be returned.

It must be noted that we cannot use the name `dlsym` to call

```
void *dlsym(void *handle, const char *symbol) {
    auto ptr = original_dlsym(handle, symbol);
    ... /* other OpenGL API functions */
    if (strcmp("glDrawArrays", symbol)==0) {
        /* initialize original function pointer */
        original_glDrawArrays = ptr;
        /* return GLACI wrapper function */
        return glDrawArrays;
    }
    ... /* other OpenGL API functions */
    return ptr;
}
```

Figure 5. Example of resolving glDrawArrays with modified dlsym.

```
void *original_dlsym(
  void *handle, const char *symbol)
{
  static dlsym_func_t original_dlsym_ptr
    = nullptr;
  if (original_dlsym_ptr == nullptr)
  {
    auto lib_handle =
      dlopen("libc.so.6", RTLD_LAZY);
    original_dlsym_ptr
      = dlvsym(lib_handle, "dlsym",
          GLIBC_VERSION_STR);
  }
  return original_dlsym_ptr(handle, symbol);
}
```

Figure 6. The core logic of original_dlsym.

```
void *glXGetProcAddress(const char *procName)
{
    auto procPtr = (*original_glXGetProcAddress)(
        procName);
    ... /* other OpenGL API functions */
    if (strcmp("glHint", procName) == 0)
    {
        /* initialize original function pointer */
        original_glHint = procPtr;
        /* return GLACI wrapper function */
        return glHint;
    }
    ... /* other OpenGL API functions */
    return procPtr;
}
```

Figure 7. Example of resolving glHint with modified glXGetProcAddress.

the original version of `dlsym` function, since it has already be overridden by our module. To avoid this circular reference, GLACI implements `original_dlsym` function as shown in Figure 6, which can search and call the original `dlsym` using `dlvsym` (dlsym with versioning) function.

Similarly, to support the method using the OpenGL *GetProc* functions, GLACI also implements modified versions to override them. An example of `glXGetProcAddress` is shown in Figure 7. Because the original function pointers of *GetProc* functions can be obtained from the other two methods, they are more easier to implement than the modified `dlsym` function.

With these symbol resolution methods supported, the GLACI-based module can fully intercept all OpenGL API calls to execute the instrumented code.

C. Code Instrumentation Example

GLACI instruments the OpenGL API functions by following the hooks defined in the instrumentation script of the module project. Figure 8 is an example of a hook for printing debug messages. A filter function is set to the `is_target` parameter so GLACI core will only apply this hook to OpenGL API of draw commands. The `before_run` and `after_run` parameters specify the code should be added before and after calling the hooked function.

```
def _print_enter(f: func.Func) -> str:
    return f'std::cerr << "{f.name} Enter" << std::
        endl;'

def _print_leave(f: func.Func) -> str:
    return f'std::cerr << "{f.name} Leave" << std::
        endl;'

debug_hooks = func.Hooks(
    header="#include <iostream>",
    hook_funcs=[
        lambda f: func.Hook(
            is_target=lambda f: "glDraw" in f.name,
            before_run=_print_enter(f),
            after_run=_print_leave(f),
        ),
    ],
)
```

Figure 8. Example of hook in the instrumentation script.

```
extern "C" PUBLIC
void glDrawBuffer(GLenum buf) {
    std::cerr << "glDrawBuffer Enter" << std::endl;
    (*original_glDrawBuffer)(buf);
    std::cerr << "glDrawBuffer Leave" << std::endl;
}

extern "C" PUBLIC
void glDrawBuffers(GLsizei n, const GLenum *bufs) {
    std::cerr << "glDrawBuffers Enter" << std::endl;
    (*original_glDrawBuffers)(n, bufs);
    std::cerr << "glDrawBuffers Leave" << std::endl;
}

... // other instrumented *glDraw* functions
```

Figure 9. Example of generated wrapper functions.

After processing this hook, GLACI will generate the wrapper functions for OpenGL *glDraw* API as shown in Figure 9.

IV. GPU Resource Manager Prototype

Although the GLACI-based module is also able to work as a standalone tool, the key characteristic distinguishing our method from existing tools is that it can communicate and cooperate with the GPU resource manager to dynamically monitor and control the running OpenGL applications. This feature makes GLACI a useful tool in both the development environment and the production environment. As a proof-of-concept, we have created a prototype that includes a GLACI-based module and a GPU resource manager. In this section, we will use it to explain how GLACI can help in implementing several real-world use cases.

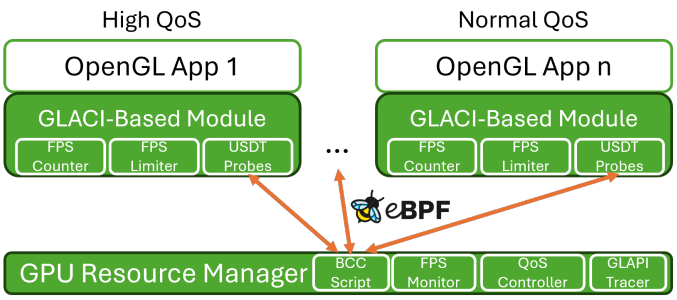


Figure 10. The overview of GPU resource manager prototype.

Figure 10 shows the overview of our prototype. The OpenGL applications are launched by the GPU resource manager with QoS priority assigned and GLACI-based module loaded. Userspace Static Defined Tracing (USDT) probes [13], generated by the GLACI-based module, are used as the communication channel between the application and the GPU resource manager to achieve dynamic control. By default, these probes are just No Operation (NOP) instructions with ignorable performance cost. The GPU resource manager includes a BPF Compiler Collection (BCC) [14] script which can dynamically generate and attach extended Berkeley Packet Filter (eBPF) programs to obtain information from and send control parameters to the running OpenGL applications. This lightweight yet extendable communication mechanism allows us to support the services of GPU resource manager with very low overhead. We have implemented the several services to demonstrate that GLACI can help to address real-world use cases as follows.

FPS monitor and alarm. OpenGL applications, especially those prebuilt ones, are usually designed to work at a specific target FPS to achieve a predictable GPU resource usage. If the actual FPS of an application differs significantly from the target FPS, there is a high probability that some issue has occurred during the execution. GLACI will insert the code and USDT probe of a frame counter around the OpenGL frame-swapping API to gather the FPS data. The GPU resource manager will attach to the related probe of each application to achieve a system-wide FPS monitoring in real time. If any unexpected FPS value has been detected, it can further generate an alarm to trigger necessary actions (e.g., start tracing the related OpenGL application).

QoS-based resource control. To deliver a sufficient quality of service, it is typically necessary to adjust the GPU resource usage limit per application at runtime. For example, if an application with normal QoS priority is the only running application, it can be allocated with full GPU resource. However, if applications with normal and high QoS priority are running at the same time, we should limit the GPU usage of the normal one to guarantee the FPS of application with high priority. The GLACI-based module has implemented a simple FPS limiter which is disabled by default. The limiter uses the control parameter to calculate the minimum render time of a frame. If the frame time of application is rendered faster than the limit's value, necessary delay duration will be inserted. When the application of high QoS priority is executed or terminated, the GPU resource manager will adjust and enable the FPS limiter of the normal QoS ones by sending control parameter to their related probes.

On-demand OpenGL API tracing. Although tracing the API calls is very helpful to analyze the performance and behavior of OpenGL applications, the usefulness of exist-

TABLE I. EVALUATION PLATFORMS

	Intel NUC	NVIDIA Jetson
CPU	Core i5-1240P	6-core Arm Cortex-A78AE v8.2
GPU	Intel Iris Xe	1024-core NVIDIA Tegra Orin
RAM	32GB	8GB
OS	Ubuntu 22.04	Ubuntu 22.04

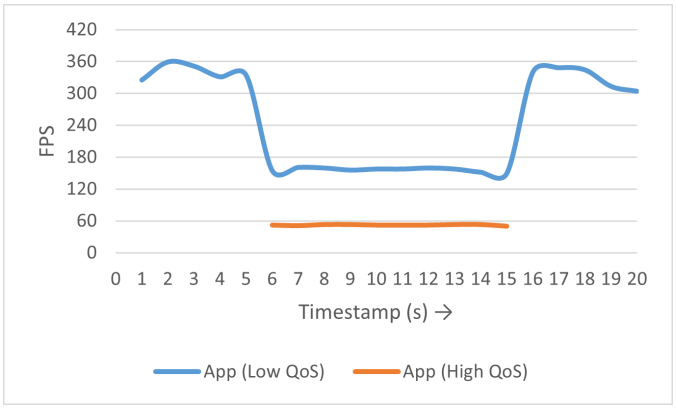
ing tools is severely restricted due to the lack of support in the production environment. The scope of these tracing tools cannot be dynamically changed while applications are running, which leads to unavoidable high overhead of system-wide continuous tracing and frequent restarts of applications. GLACI can overcome this drawback by supporting the on-demand tracing feature. It will insert USDT probes at the entry and exit points of each OpenGL API function. The tracing code of these probes can be attached or detached as needed by the GPU resource manager without restarting the running applications. This feature allows us to effortlessly create useful tracing policies. For example, we can disable all tracers by default to deliver the best overall system performance, and automatically enable tracing for a specific application when a performance issue is detected from that application.

Our experience in developing this prototype confirms that the learning curve for implementors is modest in practice. The GLACI-based module employs a simple instrumentation script written in Python, which is accessible to developers with basic scripting experience. Moreover, integrating GLACI modules with the GPU resource manager via USDT probes and eBPF programs does not require extensive prior knowledge, as these technologies are widely adopted and have substantial community support. Therefore, implementors can efficiently leverage our proposed method with minimal initial effort.

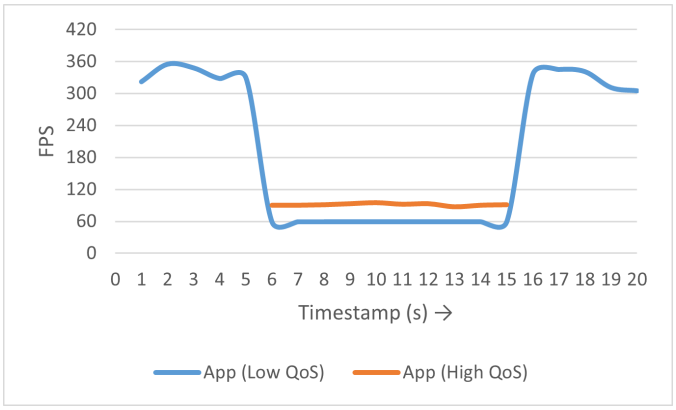
V. EVALUATION

In this section, we evaluate the GPU resource manager prototype on two mainstream platforms, Intel NUC and NVIDIA Jetson, to examine the functionality and overhead of GLACI. Intel platform provides an open source OpenGL library while the NVIDIA one is proprietary. Although the OpenGL implementations are vastly different, the high portability of GLACI allows us to use the same source code to build the prototype project without reimplementing for each platform. The specifications of the evaluation platforms are shown in Table I.

The benchmark programs from `glmark2` [15] (version 2021.02) are chosen as the graphics applications to test our prototype. `glmark2` is a lightweight OpenGL benchmark suite widely available on many platforms, with 17 representative scenes included to measure many aspects of the OpenGL specification. Since our method does not require any source code modification to the application and graphics stack, all related software components are installed using the official binary packages from Ubuntu. In this section, we always set the rendering resolution of `glmark2` to 1920x1080 for evaluation.



(a) Default settings without GLACI



(b) Prototype of GPU resource manager and GLACI-based module

Figure 11. Example to demonstrate how our prototype can improve QoS.

To demonstrate the effectiveness of QoS-based resource control, we run and measure an example using two `glmark2` program: `refract` and `terrain`. `refract` can run at around 333 FPS with full GPU resource allocated on the NVIDIA platform while `terrain` can run at around 119 FPS under the same condition. We use `refract` as the application with low QoS level and `terrain` with high QoS level. In the experiment, `refract` starts at first and keeps running, and `terrain` will start 5 seconds later and run for 10 seconds, to simulate the scenario the user launches an application with high QoS level while an application with low QoS level is running. Figure 11 shows the FPS data measured on the NVIDIA platform, and the Intel platform also has a similar trend. Without GLACI, the application with high QoS level failed to meet 60 FPS requirement (only 52 FPS on average). With the GPU resource manager and GLACI-based module, when application with high QoS level is running, the application with low QoS level will be locked to 60 FPS to deliver a desired performance for both applications.

We have measured the average FPS of `terrain` under the following conditions on the two platforms to evaluate the runtime overhead of our prototype.

- **No GLACI Loaded**: The application runs without GLACI-based module loaded.

TABLE II. OVERHEAD OF THE PROTOTYPE

	`glmark2 (terrain)` Average FPS	
	Intel NUC	NVIDIA Jetson
No GLACI Loaded	222	119
FPS Monitor & Alarm	221	118
QoS-based FPS Limiter	220	118
API Tracing (NOP)	218	117
API Tracing (Logging)	212	116

- **FPS Monitor & Alarm**: GLACI-based module is loaded and the GPU resource manager enables the function of FPS monitoring and alarming.
- **QoS-based FPS Limiter**: Besides the FPS monitoring and alarming, the QoS-based FPS limiter is also enabled.
- **API Tracing (NOP)**: GLACI-based module is loaded and the GPU resource manager enables API tracing. However, all API probes are attached with an empty function.
- **API Tracing (Logging)**: All API probes are attached with a logging function sending related messages to the trace buffer.

The measured results are shown in Table II. Compared to the average FPS without GLACI loaded, the overhead is barely perceptible to human eyes. It indicates that, the performance cost of GLACI should be small enough to be used in the production environment.

VI. CONCLUSION AND FUTURE WORK

This paper introduced GLACI as a generic, portable and low-overhead framework for dynamically instrumenting OpenGL API calls. By completely overriding the symbol resolution mechanism of OpenGL, GLACI can overcome the common and major limitation of existing methods that requires source code modifications to the application, OpenGL library or GPU driver.

We created a prototype to showcase how GLACI-based module and GPU resource manager can communicate and cooperate to support real-world use cases including performance monitoring, dynamic resource allocation adjustments and on-demand tracing. Two representative Intel and NVIDIA systems are used to evaluate the portability and usefulness of the prototype. The experimental results of overhead measurement confirm that the proposed approach remains lightweight enough for production scenarios.

Future work will focus on applying GLACI to further improve the GPU resource management, particularly in areas such as adaptive QoS management on metrics like utilization, frame time and render latency. Since some vendors have started to release open-source kernel-space GPU drivers in recent years, expanding GLACI's capabilities to support hooks at the GPU driver level and enabling closer integration with GPU driver control mechanisms are also key directions. These vendors may also embrace our approach to create open source debugging, tracing and management tools, since the lightweight and non-intrusive design of GLACI can offer enhanced instrumentation features with minimal efforts. Leveraging GPU driver abstraction layers, such as Gallium3D [16], could facilitate these advancements and further enhance the tool's applicability. Taken together, these advances can mark an important step toward a comprehensive, vendor-agnostic framework for managing the complex GPU requirements of modern embedded systems.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Suzuki Motor Corporation for their valuable collaboration and support throughout this research. Their expertise and contributions have been essential to the success of this work.

REFERENCES

[1] Khronos, "Opengl - the industry standard for high performance graphics," 2021, [Online]. Available: https://www.opengl.org/ (visited on 04/13/2025).

[2] Y. Wang, C. Liu, D. Wong, and H. Kim, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks," in *36th Euromicro Conference on Real-Time Systems (ECRTS 2024)*, 2024.

[3] Q. Lu, J. Yao, H. Guan, and P. Gao, "Gqos: A qos-oriented gpu virtualization with adaptive capacity sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, pp. 843–855, 2020.

[4] M. Xue *et al.*, "gScale: Scaling up GPU virtualization with dynamic sharing of graphics memory space," in *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, 2016, pp. 579–590.

[5] S. Schnitzer, S. Gansel, F. Dürr, and K. Rothermel, "Concepts for execution time prediction of 3d gpu rendering," *Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014)*, pp. 160–169, 2014.

[6] S. Schnitzer, S. Gansel, F. Dürr, and K. Rothermel, "Real-time scheduling for 3d gpu rendering," in *11th IEEE Symposium on Industrial Embedded Systems (SIES 2016)*, 2016, pp. 1–10.

[7] GLACI, "Source code," 2025, [Online]. Available: https://github.com/ertlnagoya/glaci-icons-2025 (visited on 04/13/2025).

[8] S. Kato, K. Lakshmanan, R. R. Rajkumar, and Y. Ishikawa, "Timegraph: Gpu scheduling for real-time multi-tasking environments," in *USENIX Annual Technical Conference*, 2011.

[9] B. Karlsson, "Renderdoc," 2025, [Online]. Available: https://renderdoc.org/ (visited on 04/13/2025).

[10] apitrace, "Source code," 2025, [Online]. Available: https://apitrace.github.io/ (visited on 04/13/2025).

[11] Intel, "Graphics performance analyzers," 2025, [Online]. Available: https://intel.github.io/gpasdk-doc/ (visited on 04/13/2025).

[12] NVIDIA, "Nsight graphics," 2025, [Online]. Available: https://developer.nvidia.com/nsight-graphics (visited on 04/13/2025).

[13] B. Cantrill, M. W. Shapiro, and A. H. Leventhal, "Dynamic instrumentation of production systems," in *USENIX Annual Technical Conference, General Track*, 2004.

[14] iovisor, "Bpf compiler collection (bcc)," 2025, [Online]. Available: https://github.com/iovisor/bcc (visited on 04/13/2025).

[15] glmark2, "Source code," 2025, [Online]. Available: https://github.com/glmark2/glmark2 (visited on 04/13/2025).

[16] Mesa 3D, "Gallium documentation," 2025, [Online]. Available: https://docs.mesa3d.org/gallium/index.html (visited on 04/13/2025).