IARIA

GREEN 2025

The Tenth International Conference on Green Communications, Computing and Technologies

ISBN: 978-1-68558-311-8

October 26th - 30th, 2025

Barcelona, Spain

GREEN 2025 Editors

W. Bernard Lee, Founder and Chief Executive, HedgeSPA, Private Limited, USA

GREEN 2025

Forward

The Tenth International Conference on Green Communications, Computing and Technologies (GREEN 2025), held between October 26th, 2025, and October 30th, 2025, in Barcelona, Spain, continued a series of events focusing on current solutions, stringent requirements for further development, and evaluations of potential directions related to green technologies.

Expected economic, environmental and society wellbeing impact of green computing and communications technologies led to important research and solutions achievements in recent years. Environmental sustainability, high-energy efficiency, diversity of energy sources, renewable energy resources contributed to new paradigms and technologies for green computing and communication.

Economic metrics and social acceptability are still under scrutiny, even though many solutions, technologies and products are available. Deployment on a large scale and a long-term evaluation of benefits are under way in different areas where dedicated solutions are applied.

We take the opportunity to warmly thank all the members of the GREEN 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 GREEN 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 GREEN 2025 organizing committee for their help in handling the logistics of this event.

We hope that GREEN 2025 was a successful international forum for the exchange of ideas and results between academia and industry for the promotion of progress in the field of green communications, computing, and technologies.

GREEN 2025 Chairs

GREEN 2025 Steering Committee
Mamadou Baïlo Camara, University of Le Havre Normandy, France
Sanjeev Sondur, Oracle / Temple University, USA
Bernard Lee, HedgeSPA, Singapore

GREEN 2025 Publicity Chairs
Lorena Parra Boronat, Universidad Politécnica de Madrid, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain

GREEN 2025
Committee

GREEN 2025 Steering Committee

Mamadou Baïlo Camara, University of Le Havre Normandy, France
Sanjeev Sondur, Oracle / Temple University, USA
Bernard Lee, HedgeSPA, Singapore

GREEN 2025 Publicity Chairs

Lorena Parra Boronat, Universidad Politécnica de Madrid, Spain
Laura Garcia, Universidad Politécnica de Cartagena, Spain

GREEN 2025 Technical Program Committee

Khelifa Abdelkrim, Unité de Recherche Appliquée en Energies Renouvelables - URAER | Centre de Développement des Energies Renouvelables - CDER, Ghardaïa, Algeria
Kouzou Abdellah, Ziane Achour University of Djelfa, Algeria
Ali Ahaitouf, USMBA, Morocco
Salem Al-Agtash, German Jordanian University, Jordan
Daniel Albiero, Universidade Estadual de Campinas (UNICAMP), Brazil
Angelo Algieri, University of Calabria, Italy
Mahmoud Amin, Manhattan College, USA
Zacharoula Andreopoulou, Aristotle University of Thessaloniki, Greece
Hala Alami Aroussi, Mohamed Premier University, Morocco
Abdellah Bah, ENSAM | Mohammed V University in Rabat, Morocco
Figen Balo, Firat University, Turkey
Hajji Bekkay, ENSA-Oujda | Mohammed First University, Morocco
Rachid Benchrifa, Université Mohammed V, Morocco
Lazhar Benmebrouk, Ouargla University, Algeria
Anne Blavette, SATIE (CNRS) | ENS Rennes | Univ. Rennes, France
Guillaume Bourgeois, La Rochelle Université, France
Khalid Bouziane, International University of Rabat, Morocco
Mamadou Baïlo Camara, University of Le Havre Normandy, France
M. Girish Chandra, TCS Research Whitefield, India
Dana Ciupageanu, National University of Science and Technology POLITEHNICA Bucharest, Romania
Daniele Codetta-Raiteri, Università del Piemonte Orientale, Italy
Luigi Costanzo, Università degli Studi della Campania Luigi Vanvitelli, Italy
Naouel Daouas, National Engineering School of Monastir, Tunisia
Rekioua Djamila, University of Bejaia, Algeria
Rachid El Bachtiri, USMBA University, Fez, Morocco
Hassan El Bari, Ibn Tofail University, Morocco
Hassan El Fadil, Ibn Tofail University, Kénitra, Morocco
Rafika El Idrissi, Mohammed V University in Rabat, Morocco
Abdellatif El Mouatamid, New Jersey Institute of Technology, USA

Copyright Information

Table of Contents

The Unaccounted Carbon Cost of AI-Assisted Software Engineering: A Hidden Debt and Sustainability Challenge

Nelly Nicaise Nyeck Mbialeu, Benjamin Leiding
Institute for Software and Systems Engineering
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
e-mail: {nelly.nicaise.nyeck.mbialeu | benjamin.leiding}@tu-clausthal.de

***Abstract*—Software engineering is experiencing the impact of AI on its productivity through rapid code generation, code fixes, and workflow automation. However, there is a hidden cost to this convenience, namely a growing double debt of carbon emissions and technical inefficiencies that jeopardise sustainability. Carbon debt is discussed in this paper, referring to the invisible and cumulative environmental damage resulting from the frequent use of AI-driven tools. AI sustainability discussions often overlook the impact of inference phase emissions in this field, where productivity tools lack built-in insights that measure hidden, accumulated environmental burdens. There is a lack of a conceptual and structured method to incentivise carbon debt. This paper conceptually illustrates the negative contribution of AI-assisted development tools, leading to pragmatic mitigation strategies and a preliminary formalization of a measurable sustainability framework for AI-driven development workflows.**

***Keywords*-carbon debt; sustainability; technical debt; AI DevTools; green software engineering**

I. Introduction

The swift adoption of Artificial Intelligence (AI) into software engineering has fundamentally changed how software is designed, developed, tested, and maintained. Tools such as GitHub Copilot, automated bug fixers, and intelligent testing frameworks now assist developers at nearly every stage of the software lifecycle, ensuring productivity through generating code, debugging, and streamlining workflows. However, their downside is an invisible cost due to the constant use of computing power to support large-scale AI models running in the background. When software engineers use AI-generated suggestions or machine learning tests, the associated processes are executed in high-energy-consuming data centres, which contribute to a significant environmental footprint. In areas where these centres use fossil fuels or operate with passive regulatory oversight, as revealed by Elon Musk's xAI data centre, which used unauthorized gas turbines [1], the carbon intensity of AI services continues to increase, exacerbating the hidden carbon debt. The infrastructure that powers AI-driven development tools is resource-intensive, and its environmental impact extends far beyond electricity consumption. Graphic Processing Units (GPUs), the computational core of these AI tools, are produced using rare earth minerals and resource-intensive manufacturing techniques with complex supply chains, which are frequently linked to labour issues and environmentally damaging mining activities [2], [3]. Once deployed, these GPUs work in data centres that need massive cooling systems whose operations require a lot of electricity and water to keep running at optimal efficiency [4]. Likewise, the short lifespans of AI devices add to the increasing amount of electronic waste, which worsen climate change and emit harmful substances if not adequately controlled [5]. These lifecycle impact, ranging from extraction to disposal, add up to what can be referred to as a form of carbon debt (a concealed but growing environmental cost associated with the creation and application of AI technologies). While the carbon cost of training huge AI models has received a lot of attention, the accumulated impact of using them during day-to-day software development is less known and rarely acknowledged.

Like financial debt, carbon debt is a hidden expense that eventually needs to be paid back to mitigate environmental impact, as discussed in [6], for the necessity of meeting climate targets. Carbon debt is an imperceptible environmental cost that accumulates over time as a result of adopting energy-intensive AI-driven software engineering methodologies [7]. Similar to the well-known concept of technical debt in software development [8], carbon debt builds subtly and is often overlooked in the short term. Indeed, these debts are interconnected as the same AI tools that accelerate development today also contribute to escalating sustainability risks, which can have significant long-term repercussions if left unaddressed [8]. As such, carbon debt highlights the trade-offs between immediate efficiency gains and future environmental liabilities. Although end users may not be aware of the carbon impact of AI technologies, acknowledging their influence is becoming increasingly essential as these tools are integrated into development workflows. This study provides a conceptual analysis for understanding and mitigating carbon debt in software engineering by proposing a shift in mindset from prioritizing efficiency to embracing environmental responsibility. Additionally, the goal is to educate with ideas or actionable solutions needed to make carbon-aware software development a core part of responsible AI practice and frame the urgency of AI-assisted Software engineering emissions as debt before regulators or climate consequences force our hand.

This paper seeks to answer the following research questions:
RQ1: *How does AI-assisted software development contribute to carbon debt, a hidden form of environmental damage?*
RQ2: *What strategies can be applied to mitigate carbon debt without reinforcing the unethical adoption of AI in SE workflows?*

ISBN: 978-1-68558-311-8

In software engineering, AI Development Tools (DevTools) are frequently evaluated in terms of productivity and code quality, with their ecological impacts totally ignored and unmeasured. This study introduces carbon debt as a conceptual lens to understand better the long-term consequences and advocates for incorporating environmental criteria, such as the sustainability impact factor, into our assessment of the utility and responsibility of such tools. The value of this contribution is to spark discussion through the conceptual lens of carbon debt and guide future research to explore how this framing might be quantified, modeled, or embedded into development environments.

The paper is structured as follows: Section II illustrates the emissions from AI software engineering tools. Section III portrays the double debt trap and the invisible accumulation of carbon debt. Next, Section IV presents mitigation strategies across the perspectives of multiple stakeholders. The early contours of a sustainability impact assessment are proposed in Section V for future evaluation of the responsibilities of these AI tools. Critical reflections and limitations are discussed in Section VI. Finally, Section VII concludes this study and provides an outlook for further research.

II. Carbon Debt in AI-driven Software Engineering

The increasing integration of AI tools into software development workflows comes with an unclear sustainability concern that is not immediately visible but has long-term consequences for the planet, as it is underexplored. In contrast to financial costs that are obvious in cloud service bills, the AI coding tools increasingly being adopted in development build up carbon emissions [9] shrouded in opacity as invisible carbon debt. GitHub Copilot, Amazon CodeWhisperer, and ChatGPT are examples [10] of these tools that, with each query, assist developers with real-time code suggestions and blocks of logic generation, thus requiring computational resources [11]. Each interaction implies running inference on massive Large Language Models (LLMs) in energy-intensive data centres, resulting in non-trivial energy consumption [12] that scales rapidly with frequent usage. This leads to an unacknowledged environmental burden referred to as carbon debt [13].

A. *The Hidden Emissions of Everyday AI-assisted Tools*

These tools constantly feed user input to LLMs hosted in cloud infrastructures that consume much energy [12], [14], [15]. The training stage of large models, which is, in fact, energy-intensive, has been the focus of most research on the environmental impact of AI [13], [15]. But, inference (the real-time application of these models each time a developer inputs a line of code or requests code, then gets a suggestion) is instead the primary cause of carbon debt rather than training, as observed by studies quantifying carbon emissions from AI inference [16]–[18]. This continual inference workload [16], is multiplied across thousands of users, Integrated Development Environments (IDEs), and ongoing delivery pipelines, resulting in significant energy usage [19] that remains largely overlooked. As such, in the realm of AI-assisted software engineering, the impact of daily tool usage in terms of long-term carbon emissions is still quite open for investigation. The following are examples of energy-intensive cloud infrastructure that are ingrained in Software Engineering workflows and could be invoked repeatedly during coding:

- Github Copilot
  Every code suggestion GitHub Copilot generates requires inference from a LLM hosted on Microsoft Azure servers because it is frequently used as an AI coding assistance. The model powering Copilot (Codex) is fine-tuned from GPT-3 and probably used less energy during training; nonetheless, the estimated emissions of GPT-3 of about 500 tonnes of CO_2 [20] serve as a valuable benchmark for measuring the environmental impact of LLM-based tools. While Microsoft does not disclose precise figures, each suggestion is reported by community estimates to consume about 0.002kWh of energy, equivalent to roughly 1.2g CO_2 per inference [21]. This is in line with broader estimates of small-scale AI inference tasks energy use [13], [15], [17], [18]. Even though this reported value seems low in isolation, the total carbon emissions from ongoing, real-time inferences made during daily software engineering tasks could add up to a non-negligible environmental cost. This observation is consistent with broader research in the field, which shows how, when scaled to millions of operations, seemingly minor per-inference energy costs can substantially impact the overall carbon footprint [13].
- AI Testing Tools
  These tools are essential to CI/CD pipelines as they enhance developer productivity. Several automated test generators, such as EvoSuite [22] and DiffBlue [23], often generate redundant or inefficient test cases. The authors in [24] show that automated test generation using EvoSuite can produce up to 28% low cohesion and approximately 50% high coupling test methods even after test minimization. This observation suggests that many generated test cases are functionally redundant, which implies more execution and more energy consumption, as they do not improve code reliability but still burn energy. So, when such tools run frequently, the compounded energy from computing power used for testing quickly becomes significant [10], [12]. Additionally, prior studies have shown that continuous integration systems can amplify energy use by an order of magnitude (potentially 10x) when augmented with tools like test generation and fault localisation [25], [26]. The energy footprint of CI/CD workflows could rise rapidly if AI tools result in more test inefficiencies. Like the emission from Copilot, this waste is invisible to developers and silently adds to the carbon debt.

In contrast to broadly applicable AI technologies like ChatGPT, the aforementioned AI tools alleviate carbon debt more through regular integration within IDEs and pipelines as a result of their widespread adoption. These examples above highlight a critical blind spot: unlike performance measurements (such as latency and accuracy), which are clearly visible, developers do not have feedback mechanisms to identify the carbon cost of using AI tools in daily software engineering operations. Reliable carbon accounting for AI tools remains scarce, highlighting the need for transparent and standardized emission metrics for both training and inference phases.

B. Why Carbon Debt is a Debt

Debt is the commitment to pay back funds or resources, often with added interest, as defined by the Oxford English Dictionary [27]. Beyond finance, the term is often used in a metaphorical sense to refer to hidden accumulated costs that require future repayment, like environmental or technical debt. As such, the carbon debt of AI-assisted software engineering works similarly. Global threats are increasing exponentially with higher temperatures, according to the Intergovernmental Panel on Climate Change (IPCC) [28], which also notes that economic impacts from climate change are being tracked in energy, agriculture, and other vulnerable sectors [28]. As such, carbon emissions are increased by carbon debt from widespread computational operations, such as the real-time use of AI tools that make high energy consumption, which eventually contributes to the global emission budget, exacerbating the threats. The IPCC (2023) also emphasizes that postponing mitigation efforts is expected to increase future costs, including infrastructure damage and health-related impacts. Highlighting the importance of early intervention in mitigating carbon debt in AI-assisted software engineering. Similar to financial debt bearing compound interest, the seemingly negligible energy cost of each inference adds up to cumulative emissions through the numerous daily operations, which causes the carbon debt of AI-assisted tools to grow exponentially [9], [10]. For illustrative purposes, if 30% of the approximately 26 million developers [29] adopted AI coding tools at 50 suggestions per day, and assuming an estimated 0.002 kWh per suggestion [21]. The annual energy use could reach approximately 285 million kWh, which could lead to annual carbon emissions exceeding 100000 tonnes [30], that is comparable roughly to the emissions produced by 20000-25000 passenger vehicles each year [31]. These illustrative projections display the cumulative climate cost of real-time inference and are highly sensitive to adoption rates and infrastructure efficiency. The tech sector frequently presents AI as environmentally friendly (i.e., intrinsically "green") because of data centres that are powered by renewable energy. But this information is quite misleading. Firstly, cloud providers like Microsoft that are committed to achieving 100% renewable energy by 2025 through annual purchases and matching over 95% of its Scope 2 emissions using renewable energy instruments like Power Purchase Agreement (PPA) and Renewable Energy Certificate (REC) [32]. But Microsoft's 2022 report shows that, on an hourly basis, just 60% of its electricity use came from carbon-free sources, underscoring the difference between energy accounting and actual clean energy (real carbon-free) usage [32]. Secondly, training a large LLM, like GPT-3, has a carbon impact of about 552 metric tonnes of CO_2; nonetheless, this number does not include emissions from the construction of data centres or the manufacturing of GPUs [20]. This upfront carbon debt, which is paid before any inference is ever made, added to the overlooked embodied emissions, contributes highly to the overall environmental impact as increased productivity from faster code generation leads to higher energy use.

III. Technical Debt Analogy and Pipeline

Technical debt occurs when developers use shortcuts or suboptimal solutions to achieve short-term goals, which eventually increases complexity and maintenance expenses in the long run [8]. This section observes the role of AI in software engineering in relation to technical debt within the context of sustainability. It examines how the lifecycle of technical debt is well related to carbon debt.

A. AI's Hidden Tax on Code Quality

Developers identify inefficiencies in AI-generated code, as revealed in a study [33], where approximately 40% of code snippets suggested by Copilot contain security vulnerabilities. These flaws increase future maintenance workloads and reduce code quality, which may require a lot of energy use for the necessary later fixes. According to studies, code generated by AI-assisted tools often contains structural issues like poor modularity and tight coupling ("code smells") [33], [34], which can make it challenging to maintain and result in future updates that require high energy demands. These problems are similar to those of traditional technical debt but with a carbon twist.

B. Maintenance Burden of Hidden Cost

Due to challenges in code maintainability and security vulnerabilities brought by AI-suggested code, post-deployment fixes for AI-assisted software projects have been found to occur frequently [33], [35]. This goes to show the frequent need for fixes and remediation, thereby resulting in escalating energy demands indirectly linked to more carbon debt. Thus, AI-assisted development works on a carbon credit basis, whereby rising emissions from subsequent maintenance balance out the energy savings of quick initial coding. The industry lacks tools to account for this delayed sustainability responsibility, making the environmental benefits of AI's productivity questionable.

Both categories portray the double debt trap, in which AI-assisted tools create technical debt that silently inflates carbon debt. Whereby code quality tradeoffs lead to higher maintenance requirements that contribute to a loop of increasing energy consumption and carbon impact. In contrast, other technical debt dimensions, such as security and scalability debt, are typically resolved with localised solutions (such as patching a single library) instead of systemic energy loss. In essence, just as technical debt prioritises speed over quality, carbon debt similarly compromises sustainability for productivity.

The analysis of emissions from AI-SE tools (Section II) and their cumulative consequences through technical debt (Section III) yields insights into **RQ1**, illustrating that carbon debt builds up undetected throughout the lifecycle of AI-assisted software development.

IV. Strategies to Mitigate Carbon Debt

The first step in addressing the issue of carbon debt, which is a hidden cost of AI-assisted software engineering, is to identify targeted strategies and practical solutions inspired by the technical debt analogy. To avoid relegating sustainability to a secondary concern, proactive approaches are needed to ensure that AI development is in line with long-term environmental goals. Rather than providing concrete technical solutions, these approaches serve as foundational concepts for software design, educational initiatives, and future research endeavors.

A. Operationalizing Carbon Awareness

One key challenge to lowering carbon debt in software development is its invisibility. Usually, developers don't get feedback on the energy costs or carbon emissions associated with using AI tools. So, to improve awareness, carbon transparency features could be incorporated, relying on prior research on machine learning emissions tracking [18], [36]–[38].

- *Real-time emissions dashboard* to display approximate carbon emissions, for example, IDE plugins such as CodeCarbon [38], per AI suggestions, or after code completions per day.
- *Eco-modes* that limit the use of AI tools or give priority to suggestions that are energy-effective.
- *Contextual pop-ups* that alert developers when behaviours like repeated Copilot requests exceed sustainability thresholds.

These will help developers strike a balance between productivity and sustainability by highlighting the environmental impact of AI and recognizing carbon impact as an essential element of software quality.

B. Context-Aware AI Tool Usage

Selective invocation is a significant mitigating technique since the usage of AI tools varies depending on their impact. This means avoiding unnecessary applications like boilerplate code and deploying large models for complex tasks to limit useless inference overhead [12], [13]. Some sustainability-focused practices:

- Prompt engineering, which will reduce energy use by, for example, generating a low memory algorithm [34], [39].
- AI-assisted refactoring of carbon-heavy patterns, such as nested loops, to improve efficiency, and also flagging excessive energy use by setting CO_2 limits daily, for example.

Giving developers usage reports promotes introspection and effective adoption of tools.

C. Integrating Sustainability in Education

A cultural shift is necessary to mitigate carbon debt, as developers cannot efficiently handle what remains beyond awareness. Incorporating sustainability principles in software engineering education as AI becomes more ingrained in development practices [40], [41].

- Workshops on Green AI and techniques to audit AI tools for carbon efficiency.
- Hands-on exercises on energy consumption of both manual and AI-assisted tasks.
- Reflective workshops on environmental compromises in software design.

D. Policy Levers

Carbon accountability could be enforced, such as expanding the EU's Carbon Border Adjustment Mechanism to include cloud-based AI capabilities.

E. Advocating for Purpose-Limited Technology

Energy-intensive AI tools that offer only marginal benefits could be rejected or stop being used when their core justification remains weak. Thereby promoting simple, sustainable solutions like writing code manually to create maintainable software rather than using possibly redundant recommendations. Sometimes, the most moral and environmentally responsible course of action for a developer is to choose not to adopt technological advancement when the ecological costs are unjustifiable.

Without visibility into emissions, AI DevTools leave careful users in the dark (unaware of their ecological impact). Therefore, reforming the system is necessary for significant change, while individual efforts to reduce carbon debt are essential.

V. Towards Sustainability Impact Assessment

With the aim of understanding the environmental costs of AI-supported software engineering, this study presents the concept of carbon debt, which lays the groundwork for developing visible, measurable, and actionable assessment mechanisms. This concept can be considered as a focused instance of the broader Sustainability Impact Factor (SIF) framework suggested by Lawrenz et al. [42]. The authors propose measuring the fixed and variable environmental consequences of digital tools and services to promote the implementation of service-level sustainability reporting in circular ecosystems. It is worth acknowledging the significant cost associated with training large models and manufacturing the supporting hardware (i.e., *fixed sustainability impact*). However, in the present context, the focus is on the *variable sustainability impact* resulting from the ongoing, cumulative emissions from the regular use of AI tools, which accumulate invisibly over the daily interactions of millions of developers. As the systematic tracking of these emissions could form part of a measurable component within a tool-specific SIF that can guide responsible usage and development behaviours. The following Tab I suggests preliminary criteria intended for initial discussion regarding the SIF for AI DevTools:

The mitigation strategies proposed in Section IV collectively address **RQ2** and respond to the double debt issue of aiming to maintain productivity while minimizing both environmental and technical debt associated with AI tools. Section V extends the discussion toward future impact tracking models.

VI. Discussion and Limitations

Generally, in software engineering, emphasis is laid on fairness, bias, transparency, and privacy, whilst environmental

TABLE I. PRELIMINARY CONCEPTUAL DIMENSIONS PROPOSAL OF A SUSTAINABILITY IMPACT FACTOR FRAMED FROM THE CARBON DEBT CONCEPT.

Dimension	Illustrative Metric	Rationale
Operational Efficiency	CO_2e per 1000 completions or test executions	Measures the continuous cost of routine tool usage
Model Training Efficiency	Total carbon emissions during pretraining and fine-tuning	symbolizes the environmental impact of AI model development
Usage Intensity	Average daily invocations per developer	Shows the level of integration and influence the tool has in workflows
Energy Source	Ratio of renewable to fossil-powered inference	Distinguishes between greener and carbon-intensive AI operations
Hardware Lifecycle	Average GPU replacement cycle alongside e-waste per model	Highlight the physical resource impacts and disposal challenges
Transparency Practices	Availability of data on model size, energy use, and emissions	Encourages accountability in sustainability reporting

impact is often ignored [43]. To guarantee a really responsible approach to AI, software development must address carbon awareness and sustainability as equally essential components, as they indicate our broader interaction with digital infrastructure and planetary boundaries. Instead of being an afterthought, carbon impact must be a visible, measurable indicator. As AI DevTools continue to be integrated into daily workflows (e.g., Copilot, CodeWhisperer, testing suites), their substantial energy impact over time [13], [20] must be a shared responsibility between developers, hosting companies, regulators, and educational institutions. The actors mentioned above should broaden their view of accountability related to the impact of AI in software engineering on society and the environment. In an era of climate urgency, passive observation is unacceptable, as it is essential to ensure that advancement does not come at the price of sustainability.

A more effective approach worth considering is to avoid utilising AI when the assumption of its benefits could be challenged. Techno-critical scholars like Schmachtenberger [44] argue that adopting technological advancements must start with a convincing, fact-based argument that proves that their advantages outweigh their harm while justifying that the harms are reasonable. From this perspective, harm mitigation is not enough compared to not adopting these energy-intensive AI tools, as non-adoption is the most ethical and carbon-conscious decision to begin with.

Throughout this study, increasing awareness about Carbon Debt in AI-assisted Software Engineering is the goal. But it is also important to recognise a number of limitations present here:

- Although studies show that AI technologies produce emissions when inferring, this research does not provide empirical energy values nor accurate measurements of carbon implications in development processes.
- The mitigation strategies proposed were all based on the assumption that AI will remain embedded in software workflows (currently reflecting industry trends). They can make a difference, but would not solve the carbon debt problem, and as a result, a "non-use" AI solution was included but not fully explored in its radicality.
- This study acknowledges technical aspects of carbon debt but does not explore issues such as organisational, economic, and political-economic considerations that also influence AI adoption and provider infrastructure.

VII. CONCLUSION AND FUTURE WORK

The increasing integration of AI-assisted software engineering tools has unveiled an invisibly unacknowledged environmental cost, namely carbon debt. Every output produced by AI is dependent on energy-intensive infrastructure, which results in cumulative emissions that are invisible but increase with usage. This debt, left unmanaged, would accumulate over time and threaten the viability of our ecosystem, even though it does not damage the code. This study defines the concept of carbon debt, inspired by technical debt as a prism through which environmental costs of AI can be viewed. Several strategies to identify and reduce carbon debt were mentioned, such as making the invisible visible, selective AI tool usage, prioritizing long-term sustainability, distributing responsibility, and providing developer education with an emphasis on sustainability. Bringing forth the fact that carbon awareness should be part of responsible AI components in software design. Notwithstanding, the strategy of technological minimalism is mentioned. A cultural and structural shift is necessary for the future. We need toolmakers to stop considering it as an afterthought, developers to follow responsible and reflective practices, educators to equip the next generation with, for example, green coding, and policymakers to create policies that make sustainability the profitable choice. Moreover, a preliminary proposal was discussed on how carbon debt could evolve into a traceable and measurable system (conceptually inspired by previous research in circular economy modelling), thereby promoting accountability in AI DevTool ecosystems. Sustainability must be a primary concern, and discussions about ethical innovation, software quality, and the future of digital systems should all include consideration of carbon debt. A collective action is then required for the transition to a carbon-aware digital economy.

To operationalise this concept, future research should focus on building measurement tools that can take into account both direct energy use (such as code completions) and indirect infrastructure emissions (such as CI/CD pipelines). By incorporating these measurements into developer environments through IDE plugins, the carbon debt of AI-assisted software development may become easier to understand and control. Additionally, carbon-aware DevTools that offer real-time feedback regarding carbon cost should be explored, and programmable "green modes" should be investigated that can restrict high-emission model invocation or promote lighter alternatives. In another view, research should be done on formalizing carbon debt as a measurable software quality attribute alongside performance and maintainability. Moreover, research should be done to develop actual metrics for the suggested SIF, as no such standardized rating currently exists, and to test their applicability in workflows. This paper outlines an initial conceptual structure and indicators that shape the language and questions that empirical work must eventually address. This entails accounting for contextual factors such as infrastructure quality, energy systems, and developer behavior. Finally, more studies should be done on the ethical analysis of AI in Software Engineering beyond harm reduction to critically examine its necessity and bring about discussion on the non-adoption justification and/or minimalism of AI tools. The goal is to put sustainability at the center of responsible innovation debates.

REFERENCES

[1] The Guardian, "Elon Musk's xAI accused of pollution over Memphis supercomputer", [Retrieved: May 2025], Apr. 24, 2025, [Online]. Available: https://www.theguardian.com/technology/2025/apr/24/elon-musk-xai-memphis.

[2] L. Hampton *et al.*, "From Mining to E-waste: The Environmental and Climate Justice Implications of the Electronics Hardware Life Cycle", MIT Schwarzman College of Computing, 2024, [Online]. Available: https://mit-serc.pubpub.org/pub/w9ht6hue/release/5.

[3] J. Hess, "Chip Production's Ecological Footprint: Mapping Climate and Environmental Impact", [Retrieved: March 2025], 2024, [Online]. Available: https://www.interface-eu.org/publications/chip-productions-ecological-footprint.

[4] A. Zewe, "Explained: Generative AI's Environmental Impact", Published by MIT News on January 17, 2025, 2025, [Online]. Available: https://news.mit.edu/2025/explained-generative-ai-environmental-impact-0117.

[5] A. A. Fawole, O. F. Orikpete, N. N. Ehiobu, and D. R. E. Ewim, "Climate change implications of electronic waste: strategies for sustainable management", *Bulletin of the National Research Centre*, vol. 47, no. 1, p. 147, 2023.

[6] International Institute for Applied Systems Analysis, "Tracking net-zero carbon debt: Who is responsible for overshoot of the 1.5°C climate limit?", [Retrieved: April 2025], Mar. 2025, [Online]. Available: http://iiasa.ac.at/news/mar-2025/tracking-net-zero-carbon-debt (visited on 04/05/2025).

[7] W. Buchanan, *The Carbon Footprint Of AI*, https://devblogs.microsoft.com/sustainable-software/the-carbon-footprint-of-ai/, [Retrieved: April 2025], 2020.

[8] P. Avgeriou, P. Kruchten, I. Ozkaya, and C. Seaman, "Managing Technical Debt in Software Engineering (Dagstuhl Seminar 16162)", *Dagstuhl Reports*, vol. 6, no. 4, pp. 110–138, 2016, [Retrieved: April 2025].

[9] Y. Yu *et al.*, "Revisit the environmental impact of artificial intelligence: the overlooked carbon emission source?", *Frontiers of Environmental Science & Engineering*, vol. 18, no. 12, pp. 1–5, 2024.

[10] J. Vaidya and H. Asif, "A critical look at AI-generate software: Coding with the new AI tools is both irresistible and dangerous", *Ieee Spectrum*, vol. 60, no. 7, pp. 34–39, 2023.

[11] OECD, "Measuring the Environmental Impacts of Artificial Intelligence Compute and Applications: The AI Footprint", OECD Digital Economy Papers, Report 341, Nov. 2022, [Retrieved: 7 April 2025].

[12] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, *Green AI*, http://arxiv.org/abs/1907.1059, [Retrieved: April 2025], 2019. arXiv: 1907.1059.

[13] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research", in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 13 693–13 696.

[14] A. De Vries, "The growing energy footprint of artificial intelligence", *Joule*, vol. 7, no. 10, pp. 2191–2194, 2023.

[15] R. Verdecchia, J. Sallou, and L. Cruz, "A Systematic Review of Green AI", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 4, e1507, 2023.

[16] A. Singh, "Assessing the Carbon Footprint of OpenAI Models and Developing Strategies to Reduce It", Retrieved: April 2025, 2025, [Online]. Available: https://papers.ssrn.com/abstract=5195550.

[17] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, *Quantifying the Carbon Emissions of Machine Learning*, http://arxiv.org/abs/1910.09700, [Retrieved: June 2025], 2019.

[18] S. A. Budennyy *et al.*, "Eco2ai: Carbon emissions tracking of machine learning models as the first step towards sustainable AI", in *Doklady mathematics*, Springer, vol. 106, 2022, S118–S128.

[19] T. Zimmergren, *The Principles of Sustainable Software Engineering - Training*, https://learn.microsoft.com/en-us/training/modules/sustainable-software-engineering-overview/, [Retrieved: April 2025], 2025.

[20] D. Patterson *et al.*, "Carbon Emissions and Large Neural Network Training", *arXiv preprint arXiv:2104.10350*, 2021.

[21] GitHub Community, *How much CO_2/GHG does Copilot emit?*, https://github.com/orgs/community/discussions/38168, [Retrieved: April 2025], 2025.

[22] G. Fraser and A. Arcuri, "Evosuite: Automatic test suite generation for object-oriented software", in *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, 2011, pp. 416–419.

[23] Diffblue, *Discover Diffblue Cover | Diffblue Documentation*, https://docs.diffblue.com, [Retrieved: April 2025], 2024.

[24] F. Palomba, A. Panichella, A. Zaidman, R. Oliveto, and A. De Lucia, "Automatic test case generation: What if test code quality matters?", in *Proceedings of the 25th International Symposium on Software Testing and Analysis*, 2016, pp. 130–141.

[25] A. Hindle, "Green software engineering: The curse of methodology", in *2016 IEEE 23rd international conference on software analysis, evolution, and reengineering (SANER)*, IEEE, vol. 5, 2016, pp. 46–55.

[26] I. Manotas *et al.*, "An empirical study of practitioners' perspectives on green software engineering", in *Proceedings of the 38th international conference on software engineering*, 2016, pp. 237–248.

[27] Oxford University Press, *Debt Noun - Definition, pictures, pronunciation and usage notes*, https://www.oxfordlearnersdictionaries.com/definition/english/debt, [Retrieved: April 2025], 2025.

[28] K. Calvin *et al.*, *Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, First, H. Lee and J. Romero, Eds. Geneva, Switzerland: Intergovernmental Panel on Climate Change (IPCC), Jul. 2023, [Retrieved 9 April 2025].

[29] Evans Data Corporation, *Worldwide Developer Population Report 24.2*, https://evansdata.com/reports/viewRelease.php?reportID=9, [Retrieved: April 2025], 2025.

[30] US Environmental Protection Agency (EPA), *Greenhouse Gas Equivalencies Calculator*, https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator, [Retrieved: 10 April 2025], 2015.

[31] US EPA, *Greenhouse Gas Emissions from a Typical Passenger Vehicle*, https://www.epa.gov/greenvehicles/greenhouse-gas-

emissions-typical-passenger-vehicle, [Retrieved: April 2025], 2016.

[32] Microsoft Corporation, *2022 Environmental Sustainability Report*, https://news.microsoft.com/wp-content/uploads/prod/sites/42/2023/05/2022-Environmental-Sustainability-Report.pdf, [Retrieved: April 2025], 2023.

[33] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, “Asleep at the keyboard? Assessing the security of GitHub Copilot’s code contributions”, *Communications of the ACM*, vol. 68, no. 2, pp. 96–105, 2025.

[34] B. Yetiştiren, I. Özsoy, M. Ayerdem, and E. Tüzün, “Evaluating the code quality of AI-assisted code generation tools: An empirical study on GitHub Copilot, Amazon Codewhisperer, and ChatGPT”, *arXiv preprint arXiv:2304.10778*, 2023.

[35] P. Vaithilingam, T. Zhang, and E. L. Glassman, “Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models”, in *Chi conference on human factors in computing systems extended abstracts*, 2022, pp. 1–7.

[36] P. Henderson *et al.*, “Towards the systematic reporting of the energy and carbon footprints of machine learning”, *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[37] L. F. W. Anthony, B. Kanding, and R. Selvan, “Carbontracker: Tracking and predicting the carbon footprint of training deep learning models”, *arXiv preprint arXiv:2007.03051*, 2020.

[38] K. Lottick, S. Susai, S. A. Friedler, and J. P. Wilson, “Energy Usage Reports: Environmental awareness as part of algorithmic accountability”, *arXiv preprint arXiv:1911.08354*, 2019.

[39] N. Ding *et al.*, “Enhancing chat language models by scaling high-quality instructional conversations”, *arXiv preprint arXiv:2305.14233*, 2023.

[40] P. Becker, *Sustainability science: Managing risk and resilience for sustainable development*. Elsevier, 2023.

[41] B. Penzenstadler *et al.*, “Everything is INTERRELATED: Teaching software engineering for sustainability”, in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, 2018, pp. 153–162.

[42] S. Lawrenz *et al.*, “Implementing the circular economy by tracing the sustainable impact”, *International journal of environmental research and public health*, vol. 18, no. 21, p. 11 316, 2021.

[43] J. Fjeld, N. Achten, H. Hilligoss, A. Nagy, and M. Srikumar, “Principled artificial intelligence: Mapping consensus in ethical and rights-based approaches to principles for AI”, *Berkman Klein Center Research Publication*, no. 2020-1, 2020.

[44] Z. Stein, “Technology Is Not Values Neutral: Ending the Reign of Nihilistic Design”, *Consilience Project*, 2022.

Brains Without Brawn: Evaluating CPU Performance for Code Generation with Large Language Models

Miren Illarramendi[1], **Joseba Andoni Agirre**[1], **Aitor Picatoste**[2], **Juan Ignacio Igartua**[2]
[1]Software and Systems Engineering Research Group
Engineering Faculty of Mondragon University
Arrasate-Mondragon, Spain
e-mails: millarramendi@mondragon.edu, jaagirre@mondragon.edu,
[2]Circular economy and industrial sustainability
Engineering Faculty of Mondragon University
Arrasate-Mondragon, Spain
e-mails: apicatoste@mondragon.edu, jigartua@mondragon.edu

***Abstract*—This research presents a comparative analysis of the performance of various Large Language Models (LLMs) for code generation tasks executed on Central Processing Units (CPUs) without the use of dedicated Graphics Processing Units (GPUs). The study evaluates key metrics including inference time, code generation accuracy, CPU and memory usage, and energy consumption. By conducting repeated experiments, we assess the impact of model size and optimization on efficiency in environments lacking GPU resources. Energy consumption is measured using tools like CodeCarbon, focusing on the environmental impact of running these models on CPU-based systems. The findings offer insights into the trade-offs between model precision, resource usage, and energy efficiency, providing valuable guidance for developers and researchers aiming to balance performance and sustainability in low-resource computing environments.**

Keywords-LLMs; GenIA; GreenComputing; Code Generation; Energy Consumption; Sustainability.

I. Introduction

The rapid advancement of LLMs has revolutionized various fields, including Natural Language Processing (NLP), code generation, and even machine translation. Models like GPT-3, DeepSeek, and Llama have shown remarkable capabilities in tasks ranging from text generation to understanding and generating code. However, these models are computationally expensive and require significant resources, particularly during the training and inference stages [1] [2].

While GPUs are typically the hardware of choice for running large-scale machine learning models due to their high parallel processing capabilities, not all environments have access to dedicated GPUs. Many users, particularly those in resource-constrained settings or utilizing cloud computing, must rely on CPUs for model inference. CPUs, though less powerful than GPUs in terms of parallel processing, are widely available and more energy-efficient in certain use cases, especially for smaller models or lightweight tasks [3].

Despite the growing use of LLMs in production environments, there is a lack of comprehensive analysis comparing the performance and sustainability of these models when executed on CPUs versus GPUs. The existing literature focuses mainly on GPU-based performance, leaving a gap in understanding how LLMs perform in real-world scenarios where only CPU resources are available. Some research has pointed out that the energy consumption of LLMs is often underestimated in most studies, with the environmental impact becoming a significant concern when deploying models at scale.

This study aims to address this gap by conducting a comparative analysis of the performance of various LLMs for code generation tasks when executed in GPU-based environments remotely. Specifically, we will focus on several key metrics, including inference time, code generation accuracy, energy consumption, and computational cost. The initial phase will involve measuring the cost of running inference from our local CPU to understand the energy and computational efficiency of remote execution. The next step will be to extend this analysis by deploying the LLMs directly on our CPU for inference, allowing us to compare performance and resource usage when running these models in resource-constrained environments. Models to be evaluated include "gpt-4o", "gpt-4-turbo", "gpt-3.5-turbo", "gpt-4o-mini", "mistralai/Mistral-7B-Instruct-v0.3", "meta-llama/Meta-Llama-3-8B-Instruct", "alpindale/WizardLM-2-8x22B", and "Qwen/Qwen3-235B-A22B-Instruct-25072". These models have been selected due to their variety in size and diverse platforms (OpenAI, HuggingFace), providing a comprehensive comparison of different model architectures and inference performance across various levels of complexity and computational demands.

The contribution of this research is to remotely measure and monitor code generation tasks in inference across different LLMs, in terms of accuracy and CO2 footprint. By analyzing the efficiency of these models in resource-constrained environments, we provide insights into optimizing the use of LLMs and balancing cost with environmental impact.

In the following sections, we will explore the methodology used to measure these performance metrics and present the results of the experiments to offer a comprehensive comparison of model efficiency across different hardware configurations. By doing so, we aim to provide practical guidance for researchers, developers, and organizations seeking to optimize the use of LLMs in resource-constrained environments while considering cost-effective and sustainable deployment strategies.

ISBN: 978-1-68558-311-8

It is important to note that this study is limited to inference-time evaluation, does not include model training or fine-tuning, and relies partially on remote execution data, which may be affected by variables such as network latency, backend optimizations, and limited visibility into the energy consumption of proprietary systems.

The remainder of the paper is organized as follows: Section II and Section III cover the background and related work, respectively, providing the foundational context for this study. In Section IV, we describe the experimental setup and methodologies employed. Section V presents the experimental results, focusing on performance, throughput, code generation accuracy, and energy efficiency. Section VI offers an in-depth discussion of the evaluation, interpreting the results and their implications. Finally, Section VII concludes the paper and proposes directions for future work.

II. Background

The field of LLMs has seen significant advancements in recent years, driven by the rapid development of deep learning techniques and the availability of large-scale datasets. These models, such as GPT-3 [4] and BERT [5], have achieved impressive results across a wide range of NLP tasks, including text generation, translation, and question answering. More recently, specialized models, such as Codex [6] have been developed for tasks related to code generation and software development.

While LLMs have demonstrated remarkable capabilities, they come with substantial computational requirements. Training these models involves large-scale distributed computing on specialized hardware, often utilizing GPUs to speed up the process. However, inference—the process of using pre-trained models to generate outputs—can also be computationally demanding, particularly when deployed in real-time applications. Typically, GPUs are used for inference due to their ability to handle parallel processing, but not all environments have access to GPUs, especially in resource-constrained settings such as edge devices, mobile platforms, or smaller cloud infrastructures.

The challenge of resource efficiency has become increasingly important as the size of LLMs continues to grow. Models like GPT-3, with over 175 billion parameters [4], consume significant amounts of energy during inference. Studies have highlighted the environmental impact of training and running large-scale models, particularly with respect to their carbon footprint and energy consumption [1]. Energy-efficient models and the optimization of inference processes on CPUs have therefore become crucial areas of research, especially when considering the global push toward sustainable AI [2].

In parallel, the demand for code generation has increased, driven by the need to automate repetitive programming tasks, assist with code completion, and enhance software development processes. Models, such as Codex [6] have shown that LLMs can generate syntactically correct and semantically meaningful code from natural language descriptions. These models have the potential to reduce development time and improve software quality by generating boilerplate code, automating refactoring, and even suggesting optimizations.

However, the deployment of LLMs for code generation in environments with limited hardware resources, such as those relying on CPUs instead of GPUs, raises concerns about the trade-offs between performance and energy efficiency. There is limited research comparing the inference performance and energy consumption of different LLMs in CPU-based environments, which is critical for determining their practical use in everyday software development tasks. Moreover, little to no studies address the cost of inference when utilizing remote models, such as ChatGPT provided by OpenAI, which runs on cloud-based infrastructures. Understanding the energy consumption and computational costs when querying remote models from local CPU environments is crucial for optimizing resources, especially when these models are not deployed locally. This gap in the literature underscores the need for comprehensive analyses that consider both local and remote execution scenarios for LLMs.

III. Related work

The growing reliance on LLMs for tasks, such as code generation, text generation, and question answering has significantly advanced the field of artificial intelligence. However, these models, especially large-scale ones like GPT-3 [4], Codex [6], and BERT [5], have raised concerns regarding their environmental impact due to their substantial energy consumption and carbon footprint. As these models become larger, the need for energy-efficient deployment methods becomes critical, particularly when leveraging resources such as CPUs instead of GPUs, which are commonly used in research environments.

A. Energy Consumption and Sustainability in AI

The environmental impact of LLMs has been a topic of growing concern in recent research. Strubell et al. [1] highlighted the significant energy consumption required to train and run models like GPT-3, estimating that the carbon emissions of training such models can rival those of several cars over their lifetimes. This study emphasizes the need for developing models that are not only accurate but also energy-efficient, promoting the idea of Green AI. However, this research focuses primarily on the training phase and the larger-scale infrastructures typically used for training these models, rather than on their inference phase or CPU-based execution.

Schwartz et al. [2] further expanded on the concept of sustainable AI, advocating for a shift toward models that prioritize resource efficiency. They call for reducing the carbon footprint of deep learning models and propose that energy-efficient algorithms should be a focus in model design. However, their work lacks a focus on real-world inference scenarios, particularly in environments where GPUs are unavailable or impractical for deployment.

Xu et al. [7] provided a comprehensive survey on strategies to improve energy efficiency in deep learning models, addressing the growing need to reduce the environmental impact of AI systems. They reviewed a variety of techniques, including model compression, pruning, quantization, and efficient data usage, all aimed at optimizing the energy consumption of machine learning models. These methods can significantly reduce the computational load during inference, particularly for large-scale models. While this work offers valuable insights into improving the energy efficiency of deep learning systems, it does not specifically focus on the trade-offs involved in running LLMs, such as GPT-3 or Codex, on CPUs for code generation tasks—an area central to our research.

B. Inference Performance and Resource Allocation

Recent studies have explored optimizing inference efficiency by balancing the load between CPU and GPU. Patterson et al. [8] analyzed the energy consumption and carbon footprint of large-scale deep learning models and discussed strategies for improving energy efficiency during model training and inference. They highlighted the cost and energy-efficiency trade-offs between using GPUs and CPUs, with a focus on reducing the environmental impact of large models like GPT-3. However, their work primarily focuses on general model training and does not specifically address LLMs or code generation tasks. Furthermore, it does not consider the role of automated tools like MLFlow and CodeCarbon, which adjust resources dynamically based on real-time performance and energy consumption metrics.

Furthermore, a more recent study by Patterson et al. [3] provided insight into carbon-efficient machine learning practices, offering actionable strategies to reduce energy consumption in inference tasks, especially when models are run on CPUs in resource-constrained environments. This study is highly relevant to our research, as it provides an essential framework for making inference more sustainable, though it still lacks specific analysis on LLMs for code generation and their optimization on CPUs.

Incorporating energy monitoring into DevOps pipelines has recently been explored by some researchers. For example, CodeCarbon [9] provides a simple framework to measure the carbon footprint of machine learning models during training and inference. By integrating CodeCarbon into the DevOps workflow, practitioners can track the energy consumption and CO2 emissions of models in real time, making it easier to evaluate the environmental impact of model deployment. This approach has been integrated into workflows for smaller, less resource-intensive models but is rarely used for large-scale models like GPT-3 or Codex, especially in CPU-based environments.

A recent study by Rangineeni et al. [10] explored the integration of MLFlow within DevOps pipelines for continuous monitoring and optimization of machine learning models in production environments. Their work highlighted how MLFlow can be utilized to track performance metrics, log experiments, and manage model versions, enabling efficient deployment and resource allocation during inference. They also emphasized the importance of adaptive resource management, which can ensure cost efficiency and sustainability in cloud-based environments. While their research provides valuable insights into optimizing resource allocation using MLFlow, it does not specifically address the application of these practices to LLMs, such as GPT-3 or Codex, for tasks like code generation, nor does it consider the role of energy consumption metrics in the optimization process.

C. Code Generation with LLMs

The application of LLMs for code generation has gained significant attention, particularly with models such as Codex [6], which are specifically designed to generate programming code from natural language prompts. Codex has shown great potential in automating code completion, bug fixing, and refactoring tasks, but there is a lack of research on how these models perform when executed on CPU-based systems as opposed to GPUs.

A recent study by Arora et al. [11] introduced SetupBench, a benchmark designed to evaluate the ability of LLM agents to bootstrap development environments autonomously. The benchmark comprises 93 tasks spanning various programming languages, database engines, and multi-service orchestration scenarios. The evaluation of OpenHands, a state-of-the-art coding agent, revealed low success rates across task categories, particularly in repository setup and local database configuration. The study identified substantial inefficiencies in agent exploration strategies, with a significant percentage of actions being unnecessary compared to optimal human behavior. These findings highlight gaps in current agents' practical environment-bootstrap capabilities.

However, the research does not investigate the inference efficiency of these models when deployed in resource-constrained environments or on CPUs, which is a critical gap in the current body of literature. Furthermore, their focus was mainly on cloud-based models and did not consider the potential environmental impact of running these models in cloud infrastructures, where energy consumption and carbon footprint can vary significantly depending on the hardware used.

D. Gap in Literature

While the literature provides a strong foundation for understanding the energy consumption and performance of deep learning models, particularly in large-scale environments using GPUs, there is limited research specifically addressing the trade-offs and performance of LLMs for code generation when executed in CPU-only environments. Most of the existing studies focus on training and GPU-based inference, overlooking the operational efficiency and sustainability of running LLMs in low-resource environments where only CPUs are available.

Energy-efficient deployment strategies using tools like MLFlow and CodeCarbon remain underexplored for LLMs, particularly in real-time inference tasks like code generation. This paper addresses this gap by comparing the CPU-based performance and energy efficiency of various LLMs, focusing on code generation and incorporating energy monitoring through MLFlow and CodeCarbon to evaluate inference performance and environmental impact in resource-constrained environments.

IV. Experimental Setup

This section describes the setup for evaluating the performance, energy efficiency, and cost of various LLMs for code generation tasks. The models selected for the experiments are from OpenAI and Hugging Face repositories, with performance monitoring conducted using MLFlow and energy consumption tracking via CodeCarbon.

A. LLMs Selected for the Experiments

The models shown in Table I will be evaluated for code generation in C programming tasks from OpenAI and HuggingFace (via Novita as inference provider):

TABLE I: SELECTED LLMs FOR THE EXPERIMENTS.

From OpenAI	From Hugging Face
gpt-4o	mistralai/Mistral-7B-Instruct-v0.3
gpt-4-turbo	meta-llama/Meta-Llama-3-8B-Instruct
gpt-3.5-turbo	alpindale/WizardLM-2-8x22B
gpt-4o-mini	Qwen/Qwen3-235B-A22B-Instruct-2507

These models represent a range of architectures, including large-scale models like gpt-4o and optimized models like gpt-4o-mini.

B. Hardware and Operating System

- CPU: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (2.42 GHz)
- Operating System: Windows 11 Pro

The experiments will be conducted on this CPU-based system, a typical environment for users without access to high-performance hardware like GPUs for inference tasks.

C. Performance Monitoring and Energy Consumption Measurement

To evaluate the energy consumption and carbon emissions, the following tools will be employed:

- MLFlow will be integrated to monitor and track the inference time, accuracy, and computational cost of each model.
- CodeCarbon will be used to track CO emissions and energy consumption for each inference task, helping assess the environmental impact of running LLMs.

D. Inference Tasks for the LLMs

The models will generate C source code for a set of problems, which are as follows:

- Prime Number Check: *Prompt: Write a C program that checks if a number is prime. The program validation returns 1 if it is prime and 0 if not. The number to check is 11.*
- Finding the Greatest of Three Integers: *Prompt: Write a C program that defines three integer variables and prints the greatest of them. The numbers to check are 11, 22, and 33.*
- Even/Odd Check: *Prompt: Write a C program that returns 1 if the input number is even, and 0 if it is odd. The input number for testing will be 122.*
- Absolute Difference Calculation: *Prompt: Write a C program that calculates the absolute difference between two numbers. The input numbers are -122 and 11.*
- Sum of Digits: *Prompt: Write a C program that calculates the sum of the digits of the input number. The input number is 123.*

Each task was repeated 10 times for each model to ensure that the results are statistically significant and to account for potential variations in model performance across multiple runs. The generated code will be validated using gcc to ensure correctness, and the inference time, accuracy, energy consumption, and computational cost will be tracked.

E. Evaluation Criteria

- Inference Time: Time taken by the model to generate the required C code.
- Code Generation Accuracy: Correctness of the generated code and whether it can be compiled and executed without errors.
- Energy Consumption: Measured using CodeCarbon to assess the energy used during inference.
- Computational Cost: The cost of running inference on remote models (via APIs for models like gpt-4o).

F. Experimental Phases

- Phase 1: Remote Inference via APIs: The first phase will focus on querying the models remotely using API calls (for models like GPT-4 and others from Hugging Face) and measuring inference time, accuracy, and energy consumption.
- Future Work - Phase 2: Local Inference on CPU: The second phase, which will be explored in future work, will involve deploying the LLMs locally on the CPU to assess their performance and energy efficiency in resource-constrained environments. This phase will compare the local CPU performance against the remote inference to evaluate trade-offs in energy efficiency and computational cost when running on CPUs.

This study will provide insights into the cost and environmental impact of deploying LLMs for code generation, particularly in scenarios where access to GPU resources is limited. The analysis will focus on the trade-offs between performance, energy efficiency, and cost, with the remote inference phase being the first step toward a more comprehensive study that will include local deployment on CPUs as future work.

V. Results

The results of our experiments are presented across three key categories: Performance and Throughput, Code Generation Accuracy, and Energy Efficiency. The models considered for this study are from both OpenAI and Hugging Face, with varying parameter sizes ranging from 3.8 billion to 235 billion parameters. Each model was evaluated based on its execution time, accuracy, energy consumption, and CO2 emissions, which are discussed in detail below.

A. Performance Throughput

The execution time of each model varied significantly, primarily due to the differences in model size and complexity. GPT-4o and GPT-4-turbo, the largest models in the study, had execution times of 2.3 minutes and 3.5 minutes, respectively. Despite optimizations in GPT-4-turbo, it required more time to complete the task, suggesting trade-offs between speed and accuracy.

In contrast, GPT-3.5-turbo was the fastest, taking only 1.9 minutes to generate code. However, this speed came at the cost of accuracy, as shown in the next section. The smaller model, GPT-4o-mini, took 2.8 minutes, slightly slower than GPT-4o, but still significant for its reduced size.

The smaller models from Hugging Face, including Mistral-7B and Meta-Llama-3-8B, took between 3.9 and 4.7 minutes for the task, which is relatively long compared to their smaller size. Finally, the largest models like WizardLM-2-8x22B and Qwen/Qwen3-235B took 7.8 minutes and 1.2 hours, respectively, showing the strong correlation between model size and execution time.

B. Code Generation Accuracy

The accuracy of the models in generating correct C code varied significantly, with the larger models performing better in generating valid code. GPT-4o achieved the highest accuracy of 54%, demonstrating its effectiveness in generating correct code for the given tasks. On the other hand, GPT-4-turbo showed a slight decrease in performance, with 36% accuracy, indicating the speed optimizations may have sacrificed some code generation quality.

GPT-3.5-turbo, with its smaller size, performed poorly, with an accuracy of 12%, reflecting the limitations of smaller models for such complex tasks. The smaller models, such as Mistral-7B and Meta-Llama-3-8B had accuracy rates of 6% and 18%, respectively, indicating that smaller parameter models struggle with generating accurate code. Larger models like WizardLM-2-8x22B and Qwen/Qwen3-235B both showed 17% accuracy, suggesting that despite their massive size, they also faced challenges in code generation.

C. Energy Efficiency

The energy consumption per inference varied based on model size, with larger models generally consuming more energy. GPT-4o and GPT-4-turbo consumed between 0.1–0.5 kWh, which is typical for models of their size and complexity. Interestingly, GPT-4o-mini, despite being smaller, consumed 0.003 kWh, slightly more than GPT-4o, likely due to specific optimizations and the inherent inefficiency of smaller models for complex tasks.

In contrast, smaller models like Mistral-7B and Meta-Llama-3-8B consumed 0.0015 kWh and 0.0028 kWh, respectively, indicating their efficiency relative to their size. However, larger models like WizardLM-2-8x22B and Qwen/Qwen3-235B consumed significantly more energy, with values of 0.0047 kWh and 0.0071 kWh, respectively, consistent with their massive size and computational demands.

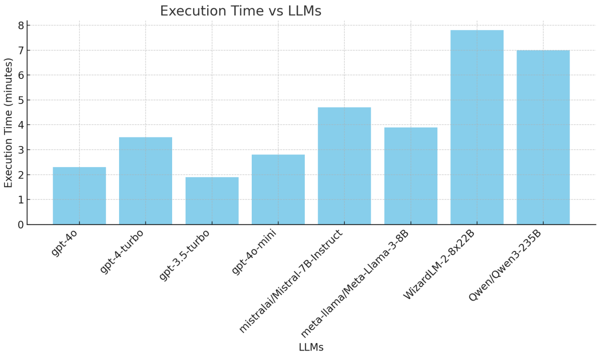


Figure 1: Execution Time vs LLMs.

CO2 emissions follow the same trend as energy consumption. GPT-4o generated 0.000132657 kg of CO2 per inference, while GPT-4-turbo emitted 0.000337222 kg. GPT-3.5-turbo produced 0.00044873 kg, further showing the inefficiency of smaller models in terms of their carbon footprint. Mistral-7B and Meta-Llama-3-8B had lower CO2 emissions of 0.000270392 kg and 0.000498069 kg, respectively, reflecting their lower energy consumption.

The largest models had the highest emissions: WizardLM-2-8x22B generated 0.001331077 kg and Qwen/Qwen3-235B generated 0.005493181 kg.

VI. Discussion And Evaluation

The results of our experiments provide valuable insights into the trade-offs between performance, accuracy, and energy efficiency when evaluating different LLMs for code generation tasks. Based on the execution time, accuracy, energy consumption, and CO2 emissions, we analyze the performance of the selected models and evaluate their practical application for real-world code generation tasks. This discussion will draw comparisons between the models and explore the implications of these results for both developers and environmental concerns.

A. Performance and Throughput

As shown in Figure 1, there is a clear correlation between model size and execution time. Larger models, such as GPT-4o and GPT-4-turbo require more time to generate code. Specifically, GPT-4o took 2.3 minutes per task, while GPT-4-turbo took 3.5 minutes. Although GPT-4-turbo is optimized for faster inference, its performance trade-offs manifest in a longer execution time compared to the base model. On the other hand, GPT-3.5-turbo is the fastest model at 1.9 minutes, but this speed comes at the cost of lower accuracy, as shown in the next section.

The smaller models like Mistral-7B and Meta-Llama-3-8B have execution times of 4.7 minutes and 3.9 minutes, respectively. Despite their smaller parameter sizes, they do not achieve significant speed advantages. In contrast, Qwen/Qwen3-235B and WizardLM-2-8x22B take the longest to execute, with 1.2 hours and 7.8 minutes, respectively, reflecting the computational burden of their massive parameter sizes.

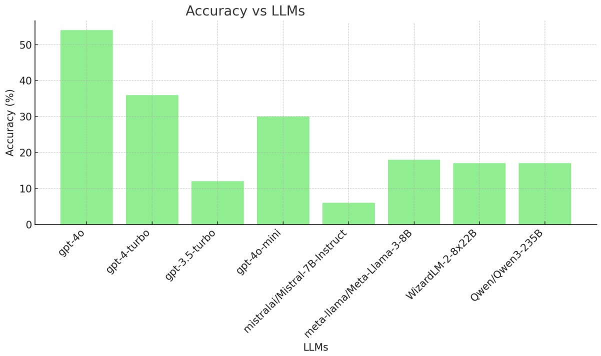


Figure 2: Accuracy vs LLMs.

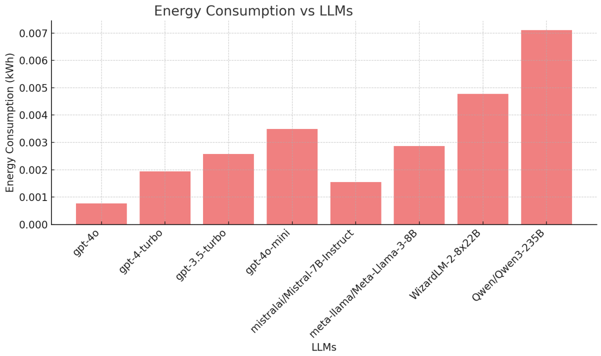


Figure 3: Energy Consumption VS LLMs.

B. Code Generation Accuracy

The accuracy of code generation, as shown in Figure 2, is heavily influenced by model size. GPT-4o outperforms all other models with a 54% accuracy, highlighting its ability to understand and generate correct code. In contrast, GPT-4-turbo achieved a lower accuracy of 36%, which suggests that speed optimizations negatively impacted the model's ability to generate correct code.

The smaller models, such as GPT-3.5-turbo (12% accuracy), Mistral-7B (6% accuracy), and Meta-Llama-3-8B (18% accuracy) perform poorly in generating correct C code, which is expected due to their limited number of parameters and training data. Despite their reduced size, larger models like Qwen/Qwen3-235B and WizardLM-2-8x22B also showed relatively low accuracy (17%), indicating that even large models do not always excel in specialized tasks like code generation, which requires deep understanding of syntax and logic.

C. Energy Efficiency

When evaluating energy consumption (Figure 3) and CO2 emissions (Figure 4), we observe a direct correlation with the model's size and computational requirements. The larger models, such as GPT-4o and GPT-4-turbo consume between 0.1 and 0.5 kWh per inference, with GPT-4o using 0.000762179 kWh and GPT-4-turbo using 0.001937503 kWh. These higher consumption rates reflect the larger energy footprint of running complex models, particularly when deployed in cloud environments that require significant computing resources.

Smaller models, such as Mistral-7B and Meta-Llama-3-8B, use far less energy, consuming 0.001553531 kWh and 0.002861641 kWh, respectively. This demonstrates that smaller models are more energy-efficient, although their reduced size results in lower accuracy for code generation tasks. While the smaller models are more energy-efficient, their performance is not optimal for generating high-quality code, making them less suitable for complex software development tasks.

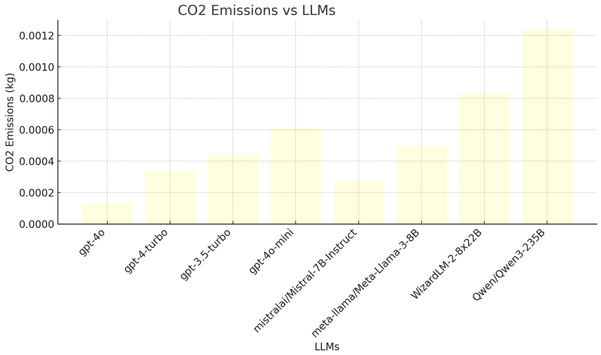


Figure 4: CO2 Emissions vs LLMs.

The larger models like Qwen/Qwen3-235B and WizardLM-2-8x22B consume significantly more energy, with Qwen/Qwen3-235B using 0.007103992 kWh and WizardLM-2-8x22B using 0.004770831 kWh. These models have the highest CO2 emissions per inference, with Qwen/Qwen3-235B producing 0.005493181 kg of CO2 and WizardLM-2-8x22B producing 0.001331077 kg of CO2. The high energy consumption and emissions of these large models suggest that while they may have certain advantages in scale, they are less efficient for deployment in resource-constrained or environmentally conscious environments.

D. Implications for Practical Use

The results indicate that larger models like GPT-4o offer the best performance and accuracy but at the expense of higher energy consumption and environmental impact (if we consider the approximate energy consumption during the training phase). These models are suitable for applications where accuracy is the primary concern, and there are sufficient computational resources. However, their high energy consumption makes them less ideal for environmentally conscious or resource-limited environments.

On the other hand, smaller models like Mistral-7B and Meta-Llama-3-8B are more energy-efficient but suffer from significantly lower accuracy, making them less suitable for complex tasks such as code generation. GPT-4o-mini shows promise with a balance of moderate energy consumption and relatively good accuracy at 30%. These models could be a viable option when moderate performance is acceptable, and energy efficiency is prioritized.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-311-8

VII. Conclusion and Future Work

This study provides a comprehensive evaluation of various LLMs for code generation tasks, focusing on performance, accuracy, energy efficiency, and environmental impact. The results reveal several key insights:

1) Larger models, such as GPT-4o and GPT-4-turbo offer superior accuracy but require significantly more execution time and consume higher amounts of energy, leading to increased CO2 emissions. These models are optimal for tasks where high accuracy is crucial, but their environmental impact makes them less suitable for resource-constrained environments.
2) Smaller models like Mistral-7B and Meta-Llama-3-8B offer better energy efficiency and lower CO2 emissions, but their accuracy is considerably reduced. These models may be suitable for scenarios where energy consumption is a priority and moderate performance is acceptable.
3) Models like GPT-4o-mini, with a balance of moderate energy consumption and reasonable accuracy, could serve as a compromise between large models and smaller, more efficient models.

In conclusion, the performance of LLMs in code generation tasks is a delicate balance between accuracy, energy consumption, and environmental impact. The choice of model should depend on the specific use case, with larger models being preferred for high-accuracy requirements, while smaller models offer better energy efficiency for more environmentally-conscious applications.

Future Work

Building on the insights from this study, several lines of future research are planned:

- Expansion to Other LLMs: The experiments can be extended to other LLMs beyond those evaluated in this study. Newer models or those from different providers may offer improvements in performance, energy efficiency, and environmental impact that could alter the current conclusions.
- Evaluation on Other Software Development Tasks: While the current study focused on code generation tasks in C programming, future experiments will expand to other types of software development tasks such as debugging, code optimization, and automatic code refactoring. This will help assess whether the trade-offs observed in this study hold true for a wider range of software engineering tasks.
- Incorporating Local Deployment: The main future direction involves repeating the experiments but this time using local LLMs. This will involve deploying the models on both local CPUs and local GPUs. By doing this, we can compare the performance and energy consumption when models are running locally, with the aim to identify the most efficient configurations for resource-constrained environments. The availability of local GPUs could offer a significant improvement in execution time and energy consumption, making it a valuable area of exploration.
- Optimization of Inference Efficiency: Future work will also include optimizing inference strategies for energy consumption and accuracy. Investigating different quantization, pruning, and distillation methods could provide potential pathways for improving the efficiency of LLMs without compromising their performance.

By addressing these future directions, the research will not only provide further insights into the trade-offs between performance and efficiency but also contribute to the development of more sustainable AI practices, particularly in the context of code generation and software development.

Acknowledgements

The authors are part of the Software and Systems Engineering research group of Mondragon Unibertsitatea (IT1519-22), supported by the Department of Education, Universities and Research of the Basque Country. This research was supported by the Ikerketa Taldeak funding (IT1519-22) and the GRECO Elkartek project (KK-2024/00090), both funded by Eusko Jaurlaritza.

References

[1] E. Strubell, A. Ganesh, and A. McCallum, *Energy and policy considerations for deep learning in nlp*, (visited on 09/03/2025), 2019. arXiv: 1906.02243 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1906.02243 (visited on 09/03/2025).

[2] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, *Green AI*, (visited on 09/03/2025), 2019. arXiv: 1907.10597 [cs.CY]. [Online]. Available: https://arxiv.org/abs/1907.10597 (visited on 09/03/2025).

[3] D. Patterson *et al.*, *The carbon footprint of machine learning training will plateau, then shrink*, (visited on 09/03/2025), 2022. arXiv: 2204.05149 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2204.05149 (visited on 09/03/2025).

[4] T. B. Brown *et al.*, *Language models are few-shot learners*, (visited on 09/03/2025), 2020. arXiv: 2005.14165 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2005.14165 (visited on 09/03/2025).

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, (visited on 09/03/2025), 2019. arXiv: 1810.04805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.04805 (visited on 09/03/2025).

[6] M. Chen *et al.*, *Evaluating large language models trained on code*, (visited on 09/03/2025), 2021. arXiv: 2107.03374 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2107.03374 (visited on 09/03/2025).

[7] J. Xu, W. Zhou, Z. Fu, H. Zhou, and L. Li, *A survey on green deep learning*, (visited on 09/03/2025), 2021. arXiv: 2111.05193 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2111.05193 (visited on 09/03/2025).

[8] D. Patterson et al., *Carbon emissions and large neural network training*, (visited on 09/03/2025), 2021. arXiv: 2104.10350 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2104.10350 (visited on 09/03/2025).

[9] MLCO2, *Codecarbon*, (visited on 09/03/2025), 2025. [Online]. Available: https://github.com/mlco2/codecarbon (visited on 09/03/2025).

[10] Y. Rangineeni and J. Pub, "End-to-end mlops: Automating model training, deployment, and monitoring", *Journal of Recent Trends in Computer Science and Engineering (JRTCSE)*, vol. 7, pp. 60–76, Sep. 2019.

ISBN: 978-1-68558-311-8

[11] A. Arora, J. Jang, and R. Zilouchian Moghaddam, *Setupbench: Assessing software engineering agents' ability to bootstrap development environments*, (visited on 09/03/2025), 2025. arXiv: 2507.09063 [cs.SE]. [Online]. Available: https://arxiv.org/abs/2507.09063 (visited on 09/03/2025).

Energy Management of a Surface Water Heat Pump Powered by Wind and a Battery System

Joyce Assaf, Mamadou-Baïlo Camara, Damien Guilbert, Brayima Dakyo
Groupe de Recherche en Electrotechnique et Automatique du Havre (GREAH)
Université Le Havre Normandie
Le Havre, Normandie, France
e-mail: joyce.assaf@univ-lehavre.fr, camaram@univ-lehavre.fr, damien.guilbert@univ-lehavre.fr, brayima.dakyo@univ-lehavre.fr

***Abstract*—The paper investigates power electronics interfaces for integrating two identical 30 kW Surface Water Heat Pumps (SWHPs) operating in parallel, with a 35 kW Wind Turbine (WT) and a 30 kWh Battery Energy Storage System (BESS), connected via a DC-bus. A simulation model was developed in MATLAB/Simulink to evaluate the behavior of the system under dynamic operating conditions, including measured data for the wind speed in 'Le Cano Ouistreham pilot site', France, and the thermal load based on data from an operational Aquathermal Energy (AE) site in Dijlemolens, Belgium. The paper addresses challenges related to efficient energy management and power control. This work presents new system-level insights, supported by simulation results. The proposed design offers flexibility and scalability, making it adaptable to integration with other Renewable Energy Sources (RESs) and a wide range of power capacities.**

Keywords-Wind energy; surface water heat pump; battery energy storage system; energy management; renewable energy.

I. Introduction

Aquathermal Energy (AE) is the extraction, storage, and distribution of heat from water bodies, including surface water, wastewater, and groundwater. It is an emerging Renewable Energy Source (RES) with strong potential for heating and cooling applications. In AE systems using surface water, heat is extracted from the water body using a Surface Water Heat Pump (SWHP) via a heat exchanger, which then increases or decreases the temperature as needed for domestic hot water supply and other thermal energy applications [1]. AE systems reduce carbon emissions in the heating and cooling sector and decrease its reliance on imported energy [2].

According to the latest IRENA report, 87 million Heat Pumps (HPs) are projected to be installed in buildings, with wind and solar expected to contribute 91% of the total Renewable Energy (RE) capacity by 2050 [3]. When coupled with RESs, such as wind or Photovoltaics (PVs) to power their compressors, SWHPs offer a promising pathway to decarbonize the building sector [4]. In Europe, the deployment of Wind Turbines (WTs) is spreading rapidly, with wind energy accounting for almost 39% of the total electricity generated from RESs in 2024 [5]. However, due to the intermittent nature of WTs, Battery Energy Storage Systems (BESSs) are integrated to effectively manage short-term fluctuations, while appropriate control strategies can significantly improve system reliability and efficiency [6].

Power electronics play an essential role in enabling the efficient integration of HPs with RESs. DC-links, incorporating AC-DC and DC-DC converters, allow for improved energy management, voltage regulation, and efficient power conversion across hybrid systems. Compared to traditional AC microgrids, DC-based systems offer lower conversion losses, improved energy management, and more straightforward control, particularly advantageous for hybrid configurations with large shares of DC loads or storage units [7].

In the literature, several studies have explored hybrid systems for HP applications [8]–[11]. Although these studies laid essential groundwork, they often overlooked power electronics integration, realistic thermal demand modeling, and dynamic power management between sources and loads. Few studies addressed power electronics control strategies and converter topologies for RE-powered HPs. For instance, in [12], the authors proposed an energy management strategy for a microgrid that combines a PV, a BESS, and an air-source HP. The main goal of the study was to reduce the cost of the BESS by leveraging the thermal storage of the HP, incorporating a double fuzzy logic that coordinates power fluctuation stabilization between the BESS and the air-source HP. However, HP operation was mainly governed by RE availability and system efficiency, not by actual, time-varying thermal demand.

In [13], the authors presented a hybrid system that includes PV, WT, and BESS powering a DC-HP. However, the approach is limited as it only accounts for the electrical power consumption of the HP, with the thermal load profile being derived from thermodynamic equations without incorporating its actual dynamics. As a result, the control and performance of the system are not realistically reflective of actual HP operation, which is typically governed by unpredictable and varying thermal loads over time. Furthermore, in [14], the authors proposed an advanced control strategy based on the model predictive control combined with a fractional short-circuit current approach to optimize power extraction from the PV system interfaced with a SWHP. However, this study modeled the SWHP as a resistive load, so results were only analyzed on the PV–BESS side. Alternatively, in [15], the authors proposed a high-voltage-gain DC-DC converter to improve PV-HP coupling efficiency. Although their converter showed promising results in handling variable solar irradiance, the system also lacked realistic thermal modeling of the HP. Although their work offers valuable insights, it remains unclear whether the results are scalable to real-world scenarios.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-311-8

This paper contributes to the growing body of research on HPs, particularly SWHPs, by presenting a realistic simulation conducted for an AE harvesting pilot site within the “Waterwarmth” project, funded by Interreg North Sea Region, in Le Cano Ouistreham, France. This pilot site is currently in the preparatory phase for constructing a SWHP system, making it an ideal case study to simulate and validate RE integration and control strategies under realistic site conditions. The proposed system employs a 35 kW WT and a 30 kWh BESS to power two identical 30 kW SWHPs operating in parallel, all connected via a DC-bus with realistic control architecture, using both measured wind speeds and thermal load profile. The originality of this study lies in its comprehensive simulation of component interactions between fluctuating RE generation and dynamic thermal demand, and its validation of a coordinated control strategy that ensures voltage stability under realistic fluctuating wind and load conditions. The proposed model provides a more accurate reflection of operational conditions and offers a pathway toward scalable, site-adapted AE solutions.

In summary, the novelty of this work lies in three aspects: (i) the use of a measured, time-varying thermal demand profile rather than a simplified one; (ii) the coordinated control of wind and battery subsystems to ensure DC-bus stability under realistic fluctuations; and (iii) the adoption of a DC-link topology that reduces conversion stages compared to conventional AC-based systems. The main objective of this study is to design and validate, through MATLAB/Simulink simulations under real site conditions, an energy management strategy that ensures reliable demand-driven operation of a hybrid WT–BESS–SWHP system.

This paper is organized as follows: Section I briefly reviews the current state of research. Section II presents the system configuration, detailing the integrated WT, SWHP, and power electronics on both the WT and BESS sides. Subsequently, the control strategies applied to each DC-DC converter are described in Section III. Section IV presents the results of the performance analysis at a pilot site within the framework of the “Waterwarmth” project. The conclusions of this work and a paper summary are presented in Section V.

II. SYSTEM CONFIGURATION

A. Wind Turbine

The system shown in Figure 1 is based on a variable-speed WT coupled with a permanent magnet synchronous generator, which converts mechanical energy to electrical energy. The rotating rotor blades extract kinetic energy from the wind and transform it into shaft torque, which is then converted into electricity by the generator. To maximize the energy extracted from the varying wind speeds, the system relies on control strategies that regulate the WT’s speed and blade pitch angle. A detailed description of the control strategy adopted is provided in Section III.

The wind speed data, which serve as input to the WT model, were collected for Le Cano Ouistreham, covering the period from March 7, 2024 at 11:00 AM to May 7, 2024 at 2:00 AM, with a sample time of one hour. This data is illustrated in Figure 2.

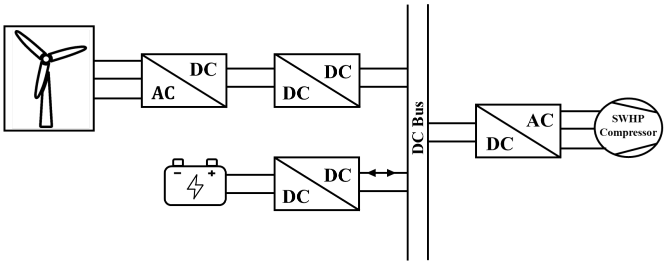


Figure 1. Electrical synoptic of the system.

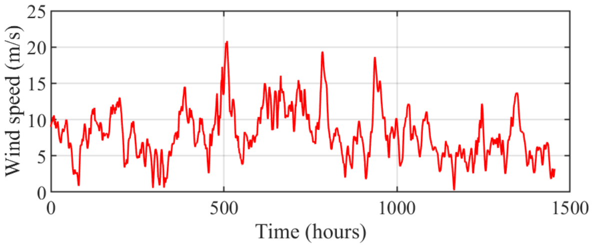


Figure 2. The wind speed profile in Le Cano Ouistreham, France.

The wind speed frequently drops below the 10.9 m/s nominal speed and occasionally falls near or below cut-in speed, reflecting the intermittent nature of wind energy. These fluctuations justify the need for both real-time WT speed control to maximize energy capture under partial load conditions and the integration of a BESS to stabilize the system’s power supply.

B. Heat pump

In this model, two identical SWHPs are used as parallel loads, ensuring that the total electrical demand is closely aligned with the rated capacities of the RE generation and BESS (35 kW WT and 30 kWh BESS). This configuration also reflects the actual thermal demand of the building and removes the need for a backup gas boiler, enabling fully RE-powered operation. The SWHP considered in the simulation is a CIAT DYNACIAT LG 300 A water-to-water HP, which provides a nominal heat capacity of 90.3 kW, a cooling capacity of 61.5 kW, and a rated electrical consumption of 29.4 kW.

The primary model inputs were the thermal power generated by the SWHP (Figure 3), and its coefficient of performance (COP), based on measurements from an installed system at an operational site in Dijlemolens, Belgium. This site also uses a backup gas boiler to satisfy the total building heat demand. The data covers the period from March 7, 2024 at 11:00 AM to May 7, 2024 at 2:00 AM, with a sample time of five seconds.

The electrical power consumption for the two SWHPs is calculated based on the thermal load profile of the building, as well as the SWHP’s technical datasheet. It is given by (1):

$$P_{\text{elec,SWHP}} = \frac{P_{\text{thermal}}}{COP} \quad (1)$$

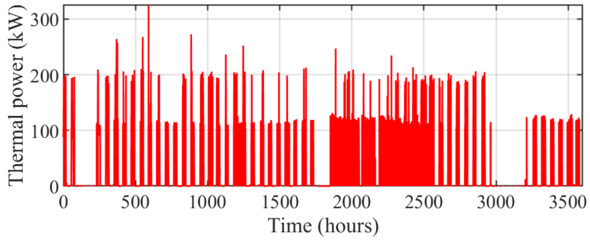


Figure 3. Thermal load profile of the SWHP.

As shown in Figure 3, the thermal load profile exhibits frequent peaks in high demand and varying on/off cycles, reflecting realistic building heating demand. Operating in a fully demand-driven mode, these dynamics emphasize the importance of a rapid control response from the supply side and a robust power balance between generation, storage, and load.

C. DC-DC boost converter on the WT side

Boost converters are commonly used in RE applications to raise the input voltage generated by the RESs, particularly when interfacing low-voltage sources, such as WTs with a higher-voltage DC-bus. In this study, the DC-bus voltage reference is 1000 V, which is a typical design choice in many RE microgrids and DC link systems, to ensure compatibility with industrial inverters and BESSs [16][17]. The converter raises the rectified WT's generated voltage to the bus voltage level, allowing efficient energy transfer under fluctuating wind conditions. Various DC-DC converter topologies are found in the literature, each with trade-offs in efficiency, control complexity, and component stress, depending on the application requirement [18]. The DC-DC boost topology used in this study is illustrated in Figure 4, with the sole purpose of validating the working principles of the overall system, providing a solid foundation for further extension and refinement.

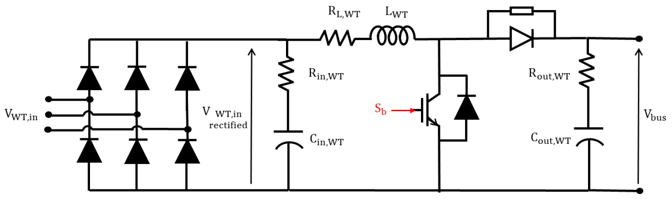


Figure 4. Topology of the AC-DC-DC converter on the WT side.

D. DC-DC bidirectional converter on the BESS side

To ensure stable operation of the DC-bus, a bidirectional buck-boost converter is implemented on the battery side. Bidirectional converters are pivotal components for energy management in RE applications, as they enable bidirectional power flow. This key feature allows excess energy generated by the RESs to be stored in the BESS during peak generation periods (buck mode) and retrieved during low-generation periods (boost mode). Therefore, their integration is vital for maximizing the utilization of RESs, as they ensure a consistent and reliable power supply. In the context of HPs, a few studies are found in the literature, including a BESS [19]. The bidirectional converter topology adopted in this study is illustrated in Figure 5 [20].

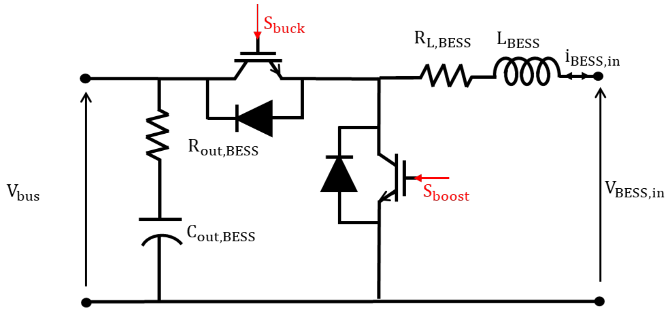


Figure 5. Topology of the AC-DC-DC converter on the WT side.

The equations governing the working principle of the buck and boost modes of this bidirectional converter are presented in (2) [21].

$$\begin{cases} \dfrac{di_{\text{BESS,in}}}{dt} = \dfrac{V_{\text{BESS,in}} - V_{\text{bus,S}_{\text{boost}}}}{L_{\text{BESS}}} \\[2ex] \dfrac{dV_{\text{bus}}}{dt} = \dfrac{i_{\text{BESS,S}_{\text{buck}}} - i_{\text{bus}}}{C_{\text{out,BESS}}} \end{cases} \tag{2}$$

where $i_{\text{BESS,in}}$ represents the current exchanged with the battery, accounting for both charging and discharging modes, V_{bus} is the measured DC-bus voltage, S_{boost} and S_{buck} are the switching signals generated by the control algorithm described in Section III, L_{BESS} is the inductance of the boost converter, and $C_{\text{out,BESS}}$ corresponds to the capacitor located at the output of the BESS, on the DC-bus side.

III. Micro-grid control methods

The control architecture of the system is designed with a clear decoupling between energy management on the source side and load operation. The SWHP functions as a passive load, turning on/off based on thermal demand. Predictive adjustment of the SWHP load is not considered; instead, the WT and the BESS are solely responsible for maintaining voltage stability and meeting demand.

A. WT Boost Converter Control Strategy

The control strategy for the WT's boost converter is based on maximizing the power extraction from the available wind-based Maximum Power Point Tracking (MPPT), while ensuring the protection of the components beyond nominal operation. Wind speed is measured and compared to a predefined nominal speed of 10.9 m/s, specific to the WT's model used. After cut-in wind speed, and if the measured wind speed is below this threshold, the reference power is computed as in (3), based on the WT power curve shown in Figure 6.

$$P = \tfrac{1}{2}\rho_{\text{air}} A C_{\text{p,max}} V_{\text{wind}}^3 \tag{3}$$

where ρ_{air} is the air density, A is the swept area of the WT blades, $C_{\text{p,max}}$ is the maximum power coefficient of the WT, and V_{wind} is the measured wind speed.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-311-8

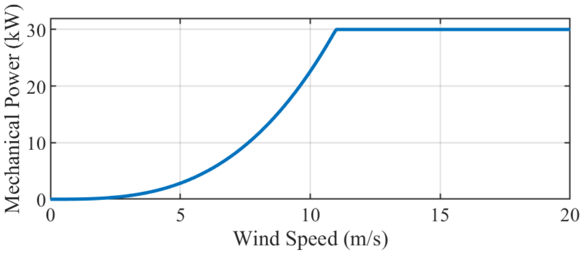


Figure 6. WT power curve.

The actual electrical power generated by the WT is measured via voltage and current sensors on the boost converter input. The power error is fed into a discrete Proportional Integral (PI) controller as illustrated in Figure 7. The error is then processed and used to generate the PWM signal for the boost converter at a switching frequency of 10 kHz, with a sample time of 10 µs.

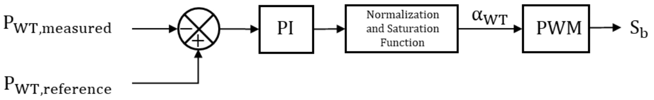


Figure 7. PWM control signal for boost switch on the WT side.

B. BESS Bidirectional Converter Control Strategy

The control of the battery-side bidirectional converter is based on maintaining the DC-bus voltage at 1000 V, while managing power flow directions (buck or boost mode) based on the instantaneous power balance between the sources and the load. For this purpose, a dual-loop bidirectional converter control was developed based on the power flow direction and the DC-bus voltage regulation. The control logic is illustrated in Figure 8 and Figure 9.

IV. Results and discussions

The system configuration previously defined in Figure 1 is simulated in Matlab/Simulink software, with the sole purpose of validating the architecture under real operating conditions. The parameters used in the simulation are listed in Table I.

Figure 10(a) shows the performance of the DC-bus voltage control system operating in buck and boost modes. The output voltage V_{bus} closely follows the constant reference voltage of 1000 V throughout the entire operation, first reaching the target value in 3.7 seconds. Afterwards, the voltage settles within a ±5% tolerance band after approximately 1.6 seconds and remains stable, showing a reliable and steady operation. The Root Mean Square Error (RMSE) of the tracking error is about 0.42%. Although occasional transient deviations occurred, the system operated within acceptable limits, with a voltage overshoot of 4.76%, which indicates a controlled transient response without excessive voltage spikes. These results demonstrate that the controller successfully maintains the voltage regulation with minimal deviation despite mode transitions and dynamic conditions.

Figure 10(b) illustrates the performance of the WT power control loop defined in Section III. The reference curve includes rapid changes and fluctuations to reflect both realistic wind and load conditions. The measured electrical power output of the WT closely follows its reference throughout the dynamic operating period, with an RMSE of almost 1.036 kW, indicating accurate and efficient tracking of the controller. Minor transient deviations occur during abrupt transitions, as expected, due to system inertia and converter response delays. However, the overall tracking behavior validates the effectiveness of the proposed control algorithm in maintaining the desired power output under varying operating conditions.

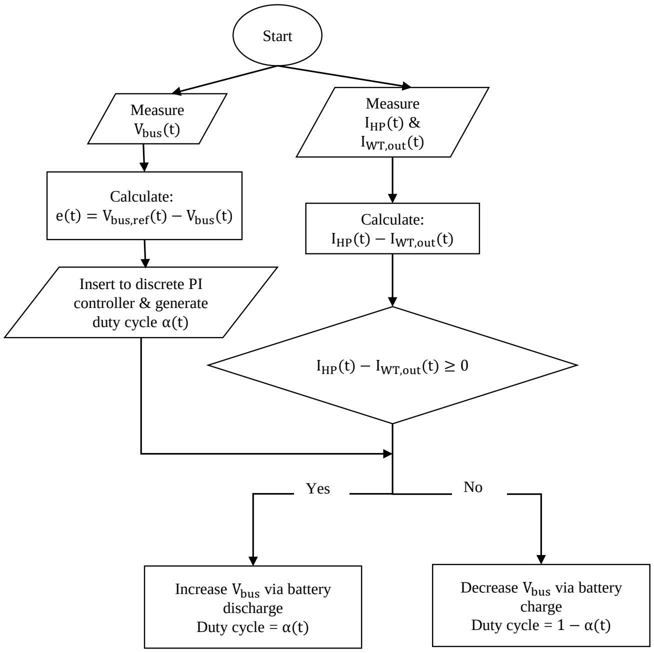

Start
Measure $V_{bus}(t)$
Measure $I_{HP}(t)$ & $I_{WT,out}(t)$
Calculate: $e(t) = V_{bus,ref}(t) - V_{bus}(t)$
Calculate: $I_{HP}(t) - I_{WT,out}(t)$
Insert to discrete PI controller & generate duty cycle $\alpha(t)$
$I_{HP}(t) - I_{WT,out}(t) \geq 0$
Yes
No
Increase V_{bus} via battery discharge Duty cycle = $\alpha(t)$
Decrease V_{bus} via battery charge Duty cycle = $1 - \alpha(t)$

Figure 8. Buck and boost modes algorithm for the BESS bidirectional converter.

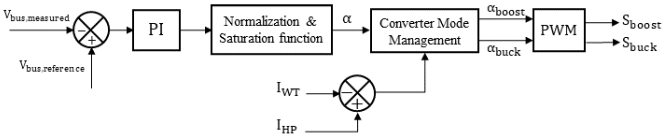


Figure 9. PWM control signals for buck and boost switches for the BESS bidirectional converter.

The energy management between the WT, BESS and SWHPs is illustrated in Figure 11 over an approximately 800-second simulation period. The WT output presented in Figure 11(a) fluctuates significantly between 0 and 35 kW due to wind speed variability, while the electrical power demand of the SWHPs is intermittent, varying between 0 and 52 kW, reflecting the dynamics of the thermal load profile, previously shown in Figure 3. The BESS operates bidirectionally, charging (negative power values) when the load demand is low, and discharging (positive power values) when the load demand exceeds the WT generation, as illustrated in Figure 11(b). The BESS provides up to 45 kW to maintain power balance. Therefore, the selected 100 Ah (35 kWh) battery is largely sufficient to cover the

TABLE I. USED PARAMETERS IN THE SIMULATION

Simulation parameters	Values
Wind Turbine	
P_{MPPT}: Maximum power	35 kW
R: Radius	5 m
v_{nom}: Nominal wind speed	10.9 m/s
ρ_{air}: Air density	1.225 kg/m^3
C_{p}: Power coefficient	0.47
WT's Boost Converter	
$C_{\text{in,WT}}$: Capacitance at rectifier's output	100 μF
$C_{\text{out,WT}}$: DC-bus capacitance	3300 μF
L_{WT}: Input inductance	1 mH
$R_{\text{L,WT}}$: Inductor's resistance	10 mΩ
BESS – Lithium-Ion	
V_{nom}: Nominal Voltage	350 V
Ah: Rated Capacity	100 Ah
BESS's Buck-Boost Converter	
$C_{\text{out,BESS}}$: DC-bus capacitance	3300 μF
L_{BESS}: Inductance	1 mH
$R_{\text{L,BESS}}$: Inductor's resistance	10 mΩ
Other parameters	
T_{s}: Sampling time	10 μs
f: Switching frequency	10 kHz

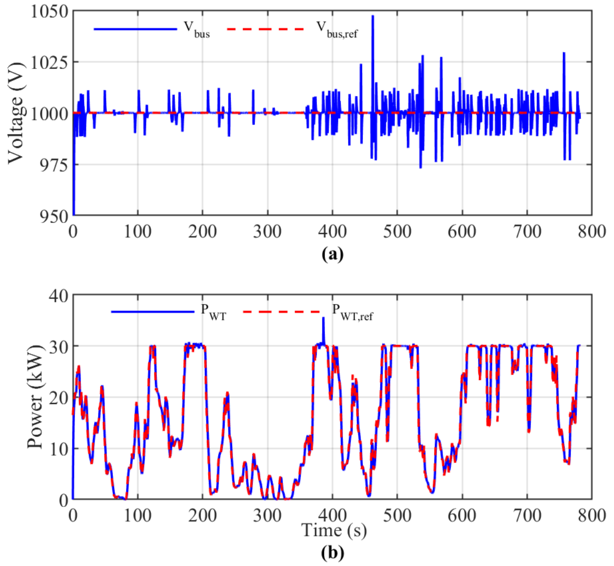
observed demand–generation imbalance under the simulated conditions. Furthermore, the overall power balance RMSE, which compares the combined generation of the WT and BESS with the electrical demand of the SWHP (Figure 11c), is almost 4 kW, indicating effective coordination between the components of the system.

Table II summarizes the key performance metrics that were derived from the simulation to quantitatively assess the dynamic performance of the proposed control strategy. These include voltage regulation accuracy, power tracking quality, and system response characteristics under variable wind speed and load conditions. The RMS tracking error represents the average percentage deviation between the measured DC-bus voltage and its reference 1000 V, computed during steady-state operation. The maximum voltage overshoot corresponds to the highest percentage by which the DC-bus voltage exceeds its reference during dynamic transients. The settling time is the duration required for the measured DC-bus voltage to return and remain within ±5% of its reference after a significant disturbance.

The simulation results confirm that the WT–BESS mix reliably tracks the HP's dynamic demand, validating control strategies under real-time conditions. In addition, the proposed DC-based architecture reduces conversion stages, achieving 95–96% efficiency—about 3–5% higher than conventional AC systems.

V. Conclusion and Future Work

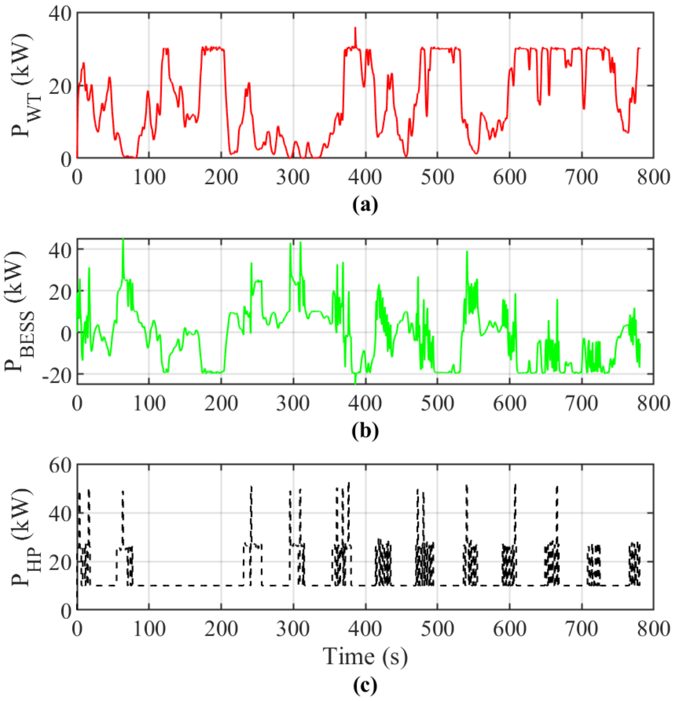
In this study, a hybrid system consisting of a Wind Turbine (WT), Battery Energy Storage System (BESS), and two Surface Water Heat Pumps (SWHPs) was developed. A complete description of the components of the proposed system is presented, along with the physical model associated with these components. The proposed control methods and their features



(a)

(b)

Figure 10. Comparison of measured and reference (a) DC-bus voltage (b) WT power output during control operation.



(a)

(b)

(c)

Figure 11. Instantaneous power profiles of the WT, BESS, and HP.

TABLE II. System Control Performance Metrics

Metric	Value
RMS voltage tracking error of DC-bus voltage	0.42%
Max voltage overshoot of DC-bus voltage	4.76%
Settling time of DC-bus voltage	1.6 s
WT electric power tracking RMSE	1.036 kW
Power balance RMSE	3.981 kW

are then highlighted. The simulation results demonstrated that the proposed control strategy achieves a fast dynamic response, stable voltage regulation, and accurate power tracking performance. This study is a proof of the feasibility and effectiveness of using such hybrid configurations to power thermal systems in a sustainable manner, offering scalable and site-adaptable solutions. Although the present study relied on classical PI-based control loops to validate feasibility, future work will investigate more innovative control strategies, such as model predictive controllers, in combination with alternative DC–DC converter topologies, to further enhance dynamic performance and robustness.

ACKNOWLEDGMENT

This work is done at GREAH-laboratory, University of Le Havre Normandie, particularly by the research team based renewable energies and storage systems (MERS). This work is done within the framework of Interreg Waterwarmth project co-funded by the European Union. The authors would like to thank the European Union and the University of Le Havre Normandie for their financial support.

REFERENCES

[1] J. D. Spitler and M. S. Mitchell, "Surface Water Heat Pump Systems", in *Advances in Ground-Source Heat Pump Systems*, Elsevier, 2016, pp. 225–246.

[2] E. Brasser, "Heating Houses using Surface Water: A Sustainable Alternative to Natural Gas", Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 2020.

[3] International Renewable Energy Agency (IRENA), "Regional Energy Transition Outlook: European Union", IRENA, Abu Dhabi, Report, 2025. DOI: ISBN:978-92-9260-658-9.

[4] A. Żelazna and A. Pawłowski, "Review of the Role of Heat Pumps in Decarbonization of the Building Sector", *Energies*, vol. 18, no. 13, 2025, ISSN: 1996-1073. DOI: 10.3390/en1813 3255.

[5] Eurostat, *Net electricity generation by type of fuel – monthly data*, [Online], Accessed: Jul. 07, 2025, 2025.

[6] D. Rekioua, "Hybrid Renewable Energy Systems Overview", in *Hybrid Renewable Energy Systems: Optimization and Power Management Control*, Springer, 2019, pp. 1–37.

[7] N. Bayati, A. Hajizadeh, and M. Soltani, "Protection in DC Microgrids: A Comparative Review", *IET Smart Grid*, vol. 1, no. 3, pp. 66–75, 2018. DOI: 10.1049/iet-stg.2018.0024.

[8] S. Baraskar, D. Günther, J. Wapler, and M. Lämmle, "Analysis of the Performance and Operation of a Photovoltaic-Battery Heat Pump System Based on Field Measurement Data", *Solar Energy Advances*, vol. 4, p. 100 047, 2024. DOI: 10.1016/j.sea dv.2023.100047.

[9] A. V. Klokov, A. S. Tutunin, E. S. Sharaborova, A. A. Korshunov, and E. Y. Loktionov, "Inverter Heat Pumps as a Variable Load for Off-Grid Solar-Powered Systems", *Energies*, vol. 16, no. 16, p. 5987, 2023, ISSN: 1996-1073. DOI: 10.3390 /en16165987.

[10] M. Bojić, N. Nikolić, D. Nikolić, J. Skerlić, and I. Miletić, "Toward a Positive-Net-Energy Residential Building in Serbian Conditions", *Applied Energy*, vol. 88, no. 7, pp. 2407–2419, 2011, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2011.01.015.

[11] C. Roselli, M. Sasso, and F. Tariello, "Dynamic simulation of a solar electric driven heat pump for an office building located in southern italy", *International Journal of Heat and Technology*, vol. 34, no. S2, S496–S504, 2016, ISSN: 0392-8764. DOI: 10.18280/ijht.34S243.

[12] L. Yang, N. Tai, C. Fan, and Y. Meng, "Energy Regulating and Fluctuation Stabilizing by Air Source Heat Pump and Battery Energy Storage System in Microgrid", *Renewable Energy*, vol. 95, pp. 202–212, 2016, ISSN: 0960-1481. DOI: 10.1016/j.renene.2016.04.026.

[13] C. B. N. Fapi, M. L. Touré, M.-B. Camara, and B. Dakyo, "Control Strategy for DC Micro-Grids in Heat Pump Applications with Renewable Integration", *Electronics*, vol. 14, no. 1, p. 150, 2025, ISSN: 2079-9292. DOI: 10.3390/electronics14010150.

[14] C. B. N. Fapi, M. L. Touré, M.-B. Camara, and B. Dakyo, "MPPT Based Fractional Short-Circuit Current-Model Predictive Control for PV System in Real Weather Conditions for Heat-Pump Applications", in *Proc. of the 2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, IEEE, 2024, pp. 1–6. DOI: 10.1109/ISCV61718.2024.1069439 5.

[15] C. B. N. Fapi, M. L. Touré, M.-B. Camara, and B. Dakyo, "High Voltage Gain DC-DC Converter Based Maximum Power Tracking from Photovoltaic Systems for Heat-Pump Applications", in *Proc. of the 2024 12th International Conference on Smart Grid (icSmartGrid)*, IEEE, 2024, pp. 493–498. DOI: 10.1109/icSmartGrid61521.2024.10557289.

[16] K. B. Samal, S. Pati, and R. Sharma, "Power Management Using an Improved EMS Algorithm in a Stand-Alone Hybrid PV-PEMFC Microgrid with Reduced Converter Count", *Green Energy and Intelligent Transportation*, p. 100 302, 2025, ISSN: 2772-3755. DOI: 10.1016/j.get.2025.100302.

[17] K. Zafar, M. K. Kamaludeen, Y. Esa, A. A. A. Mohamed, and S. Odie, "Fault analysis for dc bus-integrated energy storage system, electric vehicle supply equipment, and photovoltaic systems", *Electric Power Systems Research*, vol. 234, p. 110 837, 2024. DOI: 10.1016/j.epsr.2024.110837.

[18] J. Assaf, J. S. Menye, M. B. Camara, D. Guilbert, and B. Dakyo, "Power Converter Topologies for Heat Pumps Powered by Renewable Energy Sources: A Literature Review", *Electronics*, vol. 13, no. 19, p. 3965, 2024, ISSN: 2079-9292. DOI: 10.3390 /electronics13193965.

[19] C. Lorenzo, L. Narvarte, R. H. Almeida, and A. B. Cristóbal, "Technical Evaluation of a Stand-Alone Photovoltaic Heat Pump System Without Batteries for Cooling Applications", *Solar Energy*, vol. 206, pp. 92–105, 2020, ISSN: 0038-092X. DOI: 10.1016/j.solener.2020.05.051.

[20] M. B. Camara, "Supercapacitors for Dynamic Energy Exchange Onboard Hybrid Electric Vehicles: Modeling, Converter Study, and Control", English, PhD Thesis, Université de Franche-Comté, France, 2007.

[21] S. Agrawal, L. Umanand, and B. S. Reddy, "Bidirectional Current-Fed Converter for High Gain DC–DC and DC–AC Applications", in *Proc. of the Symposium on Power Electronic and Renewable Energy Systems Control (PERESC 2020)*, ser. Lecture Notes in Electrical Engineering, vol. 731, Singapore: Springer, 2021, pp. 101–111. DOI: 10.1007/978-981-3 3-4081-9_10.

Energy Management of PV-Batteries System for a Rural Micro-Grid Application in Guinea

Mohamed Lamine Touré[1,2], Mamadou-Baïlo Camara[1], Claude Bertin Nzoundja Fapi[3], Alireza Payman[1] and Brayima Dakyo[1]

[1] GREAH Laboratory, University of Le Havre Normandie, 75 Rue Bellot, 76600 Le Havre, France
[2] Université Gamal Abdel Nasser Conakry, Guinea
[3] LIED Laboratory, UMR 8236 CNRS, Université Paris Cité, 35 rue Hélène Brion, 75006 Paris, France
e-mail: mohamed-lamine.toure@etu.univ-lehavre.fr; mamadou-bailo.camara@univ-lehavre.fr;
claude-bertin.nzoundja-fapi@u-paris.fr; alireza.payman@univ-lehavre.fr; brayima.dakyo@univ-lehavre.fr

Abstract—**Integration of solar energy in stand-alone micro-grid applications is an attractive solution for improving access to electrification in remote rural areas and reducing dependence on the main power grid. This paper proposes a control strategy to manage the energy of a hybrid system comprising photovoltaic (PV) solar panels and energy storage batteries. The aim is to maximize the use of PV solar energy by using a PV voltage model that considers the variations of the series resistance as a function of solar irradiance and operating temperature. This ensures optimum operations of the micro-grid and improves energy efficiency. The strategy adopted is based on a dual approach that combines a maximum power point tracking algorithm with incremental conductance to extract the maximum power from the solar panels, and an energy management system based on rules control. The proportional-integral controller regulates the batteries power flow to maintain a stable DC-bus voltage within the micro-grid. System performance is evaluated in the Matlab/Simulink software environment for different load profiles and sunlight conditions. The simulations results show a significant efficiency of the control system with PVs power fluctuations mitigating by the batteries to reduce the current stress for the load.**

Keywords—Photovoltaic; ernergy management system; energy storage batteries; DC-bus voltage regulation; micro-grids.

I. Introduction

The global climate crisis and the gradual depletion of fossil fuels are making the development of sustainable energy solutions an imperative. The integration of renewable energy sources into stand-alone micro-grid systems, in particular, offers a promising alternative for reducing greenhouse gas emissions and increasing energy independence. Among these sources, solar photovoltaic (PV) energy is abundant in West-Africa (Guinea), clean and increasingly accessible thanks to the falling cost of conversion technologies.

However, optimal operation of a PV source remains complex due to its variable and non-linear nature. Solar irradiance and temperature have a strong influence on the power generated, requiring dynamic Maximum Power Point Tracking (MPPT) to ensure optimum energy yield. Several MPPT techniques have been developed, including perturb and observe (P&O), incremental conductance (InC) and adaptive step methods, such as those explored in [1], [2], [3]. The InC algorithm, in particular, is recognized for its ability to better converge to the maximum power point under conditions of high solar radiation variability.

Furthermore, energy management based PVs and batteries Energy Storage Systems (ESS) requires an Energy Management Strategy (EMS) capable of making the fast decisions to direct energy to Electric Vehicles (EVs) or to the batteries. Several studies have proposed EMS based on fuzzy rules, decision logic or hierarchical approaches. For example, Swetha et al. [4] proposes a strategy based on a centralized model for EV applications in a micro-grid environment. Oukkacha et al. [5] present an energy management method for electric vehicles based on frequency sharing between a fuel cell, lithium-ion batteries and the supercapacitors, each connected via DC-DC converters. Badawy and Sozer [6] developed and implemented an Energy Management Strategy Control (EMSC) for renewable source with the batteries micro-grid system. Baqar et al. [7] conducted a comparison of various EMSC. They are done on a hybrid system based on fuel cell, supercapacitors and batteries, highlighting the importance of effective coordination to extend the batteries life. Similarly, Bonkile and Ramadesigan. [8] proposed a solution for autonomously managing a PV-batteries hybrid system for electrical energy storage by minimizing overload constraints using the Runge-Kutta method with high-stability time steps.

Concerning the DC-bus voltage regulation, a crucial element in on-board or stationary vehicle architectures, the use of conventional controllers (PI, PID) is still common to guarantee system voltage stability [9], [10]. Yaqoob et al. [11] proposes a power EMS based on platitude control for a stand-alone photovoltaic-battery hybrid system, aimed at stabilizing the DC-bus voltage and optimizing power sharing between sources. Similarly, an EMSC for a DC micro-grid integrating a PV module, batteries and the load, aimed at optimizing energy flow while preserving battery's life, is proposed in [12]. Benzaouia et al. [13] experimentally evaluated a control strategy for a PV/battery system dedicated to water pumping applications using neural network for maximum power extraction on the PV side and fuzzy logic on the battery side to maintain the balance between supply and demand.

Despite these advances, few studies have combined an InC-type MPPT strategy with a PI control dedicated to DC-bus voltage regulation, applied to a PV/battery architecture oriented towards stand-alone micro-grids with emphasis on the voltage model of the PV system with variable series resistance, all set under realistic weather conditions. In this work, the authors propose a PV/batteries system dedicated to stand-alone micro-grid applications. Proposed EMS includes InC MPPT algorithm, which ensures the maximum power extraction of PV energy under variable irradiance and the PV series resistance. The MPPT is assisted by two PI controllers, one ensuring stability of the DC-bus voltage at 1000V through efficient control of the batteries charge/discharge via a bidirectional DC-DC converter, and the second, via a unidirectional DC-DC Boost converter, transferring power from the PV system to the DC-bus.

The system is simulated using Matlab/Simulink with realistic climatic profiles (irradiance and temperature based on pilot site of Dialakoro in Guinea). The aim is to analyze the system's performance in a variety of production and consumption scenarios, focusing on DC-bus voltage stability, battery safety and overall energy conversion efficiency. The

ISBN: 978-1-68558-311-8

results obtained demonstrate efficient regulation, optimized use of solar energy, and optimal batteries utilization, making this approach particularly suitable for stand-alone micro-grid solutions.

Following this introduction, the paper is organized as follows: Section 2 describes the micro-grid architecture, focusing on the modeling of PV sources and batteries storage systems. Section 3 is devoted to energy management strategies; it first introduces the principle of MPPT control, then details the energy management strategy. Section 4 presents the simulation results and proposes an in-depth analysis. Finally, Section 5 concludes this study.

II. Description of Micro-grid

Figure 1 illustrates the overall architecture of the hybrid system studied, which integrates a variable series resistance PV source, a battery storage system and a load represented by a home. The PV generator is connected to a boost converter, driven by a control signal (*Duty Cycle PV*) from the Maximum Power Point Tracking (MPPT) control strategy. The latter applies a MPPT algorithm to optimize the power extracted from the PV field, dynamically adapting the operating point. The storage system consists of the batteries, connected to the DC-bus via a bidirectional DC-DC converter, controlled by a dual signal (*Duty Cycle BatBus*) generated by the energy management system. This converter enables the batteries to be charged when PV production is in excess, and discharged to feed the load when solar production is low. The micro-grid, as the main load, is supplied by the DC-bus, whose voltage regulation is ensured by the coordinated power management through the two converters. This architecture enables intelligent, adaptive energy management, making it possible to integrate the PV system into a micro-grid, contributing to energy flexibility and reducing dependence on the large interconnected grid.

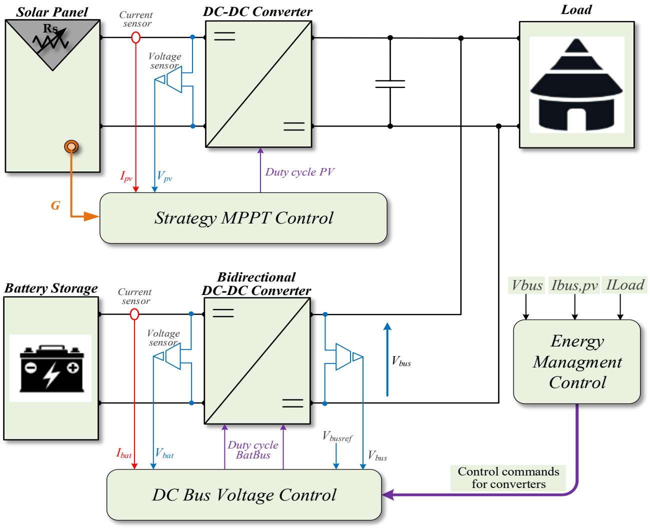


Figure 1. Proposed power system topoloy.

A. Modeling of Solar PV

Several mathematical models of PV panels have been developed with the aim of accurately representing their electrical properties and operation, which derive directly from the physical structure of PV cells as in [14], [15]. In this study, the PV model used is based on the methods described in [2]. The PV voltage model with variable series resistance is connected directly to the DC-bus via a boost converter, thus guaranteeing power transfer from the PV string to the DC-bus. Table 1 lists the main technical characteristics of the PV module and DC-DC boost converter. In addition, the voltage-current (I-V) properties of the PV panel, and the links with power output, can be explained using (1), which summarizes the electrostatic behavior of the module under standard irradiance and temperature conditions.

$$\begin{cases} V_{pv} = \alpha \cdot \dfrac{n \cdot k \cdot T}{q} \cdot \ln\left(\dfrac{I_{ph} - \frac{I_{pv}}{N_{p,ch}} \cdot k_{\lim i}}{N_{sc} \cdot I_o} \cdot \beta + 1 \right) - I_{pv} \cdot R_s \cdot \lambda \\ \alpha = N_{s,ch} \cdot N_{sc} \\ \beta = \left(\dfrac{T}{T_{ref}} \right)^3 \cdot \exp\left(\dfrac{q \cdot E_g \left(\frac{1}{T_{ref}} - \frac{1}{T} \right)}{n \cdot k} \right) \\ \lambda = \dfrac{N_{s,ch} \cdot N_{sc} \cdot k_{\lim v}}{N_{p,ch}} \\ k_{\lim i} = \dfrac{I_{sc} - I_{mp}}{I_{mp}} \\ k_{\lim v} = \dfrac{V_{oc} - V_{mp}}{V_{mp}} \end{cases} \qquad (1)$$

where I_{pv}, V_{pv}, I_o and I_{ph} represent PV current, voltage, saturation current and photocurrent, respectively. The series and parallel resistances are R_p and R_s. N_s and N_p are the total number of solar modules connected in series and parallel respectively. T is the PV surface temperature. n, E_g and k are the ideality factor, gap energy and Boltzmann constant, respectively.

TABLE I. Characteristics of PV panel and Boost converter

Electrical Parameters	Values
Solar PV source	
P_{mpp}: Maximum power	*345 W*
I_{mpp} : Maximum Current	*9.05 A*
I_{sc}: Short-circuit current	*9.52 A*
V_{mpp}: Maximum voltage	*38.14 V*
V_{oc}: Open-circuit Voltage	*46.52 V*
k_{sc}: Temperature coefficient of current	+0.049 *% /°C*
k_{oc}: Temperature coefficient of voltage	-0.315 *% /°C*
N_{sc}: Series cells	72
$N_{s,ch}$: Number of modules in series in a PV string	7
$N_{p,ch}$: Number of parallel strings in a PV string	80
DC-Dc boost converter	
C_{out}: Capacitor	*3300 µF*
L: Inductance	1 *mH*
f: Switching frequency	10 *kHz*

B. Behavior modeling of the batteries

The batteries are high energy density, quick dynamic response, and low rate of self-discharge make it a promising technology for storing renewable energy in hybrid systems. As depicted in Figure 1, the batteries are connected to the DC-bus via a bidirectional DC-DC converter, enabling efficient DC power supply to the load. The considered model of the batteries is presented by (2) [16], [17].

$$V_{bat} = E_0 - K \cdot \frac{Q}{Q - i_t} - R_b \cdot i + A_b \cdot e^{(-B \cdot i_t)} - K \cdot \frac{Q}{Q - i_t} \cdot i_f \qquad (2)$$

In (2), the open-circuit voltage is denoted by E_0. Q is the capacity (Ah) of a typical battery. i_t is the battery's current charge (Ah). The polarization constant is denoted by K. The exponential zone amplitude (in V) is shown by A_b. B represents the exponential zonetime constant inverse in the exponential

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-311-8

zone (Ah^{-1}). The internal resistance (in Ω) is denoted by R_b. The battery's current is denoted by i, and the filtered current (in A) by i_f.

The battery design was modeled using simulation based on the technical specifications listed in Table 2, which also includes the parameters of the bidirectional DC-DC converter. The model describing the battery's state of charge SoC_{bat}, is evaluated by (3) [17], [18], where SoC_{bat} is the battery state of charge (%), Q_{bat} is the maximum battery capacity (Ah).

$$SOC_{bat} = 100 \cdot \left(1 + \frac{1}{Q_{bat}} \cdot \int_{0}^{t} i_{bat}(t)\,dt\right) \tag{3}$$

The battery's charge-discharge cycle is mainly determined by the amount of power available and the power level required by the system. This cycle depends on both available energy capacity and demand dynamics. The battery's State of Charge (SoC) limits is used to define operational constraints, framing minimum and maximum operating thresholds. These limits are used to set safety limits and optimize battery use according to the actual power capacities that can be supplied or absorbed at any given time, as described in (4), where SoC_{batmin} and SoC_{batmin} are respectively the state of charge minimum and maximum.

$$SOC_{bat\,\min} \leq SOC_{bat} \leq SOC_{bat\,\max} \tag{4}$$

TABLE II. CHARACTERISTICS OF BATTERY AND BIDIRECTIONAL CONVERTER

Electrical Parameters	Values
Battery source	
Q	*648 Ah*
E_0	*273.2 V*
K	*0.0029*
R_b	*0.0038*
A_b	*21.16*
B	*0.094*
SOCbat	*50%*
DC-DC bidirectional converter	
C_{bus}: Capacitor	*3300 µF*
L_{bat}: Inductance	*1 mH*
f: Switching frequency	*10 kHz*

III. ENERGY MANAGEMENT STRATEGIES

This section provides an in-depth explanation of the MPPT control principle and the energy management strategy proposed in this paper.

A. Principle of MPPT Control for PV

The photovoltaic (PV) panel, as a generator of electrical energy, converts solar irradiance into electricity through a direct conversion process. Nevertheless, this conversion is subject to the intrinsic variability of solar resources, which induces intermittent and non-linear behavior in energy production. This characteristic adversely affects the stability and performance of the PV system. To mitigate these effects and optimize energy capture, the integration of a dedicated control system is essential. To do this operation , the MPPT algorithms play a crucial role, adapting the operating point of the PV grid in real time to extract the maximum available power. Several MPPT strategies have been proposed in the literature [2], [19], [20], aiming to autonomously control the voltage of each PV module, in order to maximize energy efficiency under varying sunlight and temperature conditions. In the present work, we have opted for the control strategy shown in Figure 2 with the integrated Incremental Conductance (InC) algorithm, whose principle is illustrated in Figure 3, as the method for tracking the maximum power point. The InC algorithm is based on an analysis of the power gradient as a function of voltage (dP/dV), enabling precise estimation of the optimum operating point. In combination with the irradiance sensor, it provides the reference current that is calculated using (5).

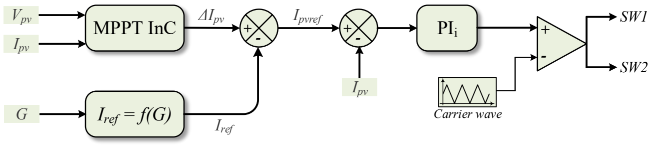


Figure 2. Strategy MPPT Control.

$$I_{ref} = \left(I_{sc} + k_{sc} \cdot (T - 298.15)\right) \cdot \left(\frac{G}{1000}\right) \tag{5}$$

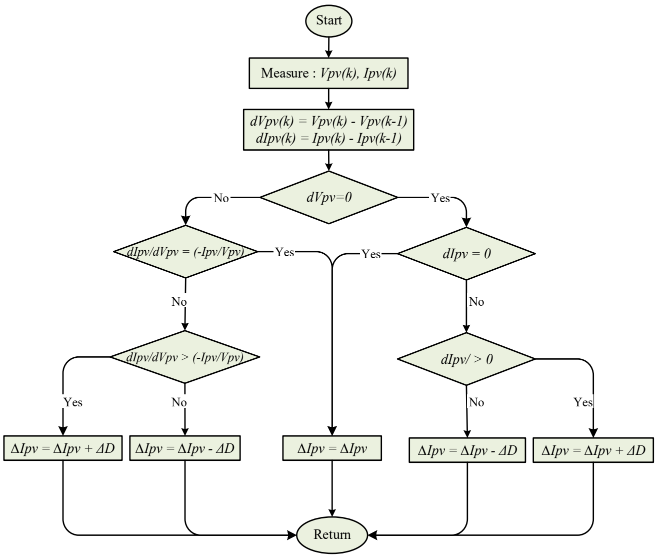


Figure 3. MPPT InC.

B. Energy Management Strategy Control (EMSC)

The main objective of this study is to propose an innovative energy management strategy in a micro-grid, based on the use of a Proportional-Integral (PI) regulator, allowing for optimal energy transfer between the different components of the system. A fundamental aspect of this approach lies in maintaining the stability of the direct current DC-bus voltage, which must remain at a predefined reference level, regardless of external disturbances or inherent system uncertainties. The stability of this voltage is indeed crucial to ensure the performance, reliability, and operational safety of the micro-grid.

The proposed method is based on minimizing the error between the measured DC bus voltage (V_{bus}) and the setpoint (V_{busref}) using a robust and systematic control scheme. The micro-grid's electrical system must constantly adapt to variations in the output of PV sources. In this context, battery loading becomes essential to avoid system instability in the event of production deficits and overproduction, thus ensuring balance between supply and demand.

Given that battery is coupled to the DC-bus via a bidirectional DC-DC converter, the PI controller's main task is to drive this converter in buck or boost mode, depending on energy requirements. The reference current, generated by the PI voltage controller, is used to determine the duty cycle applied to the pulse-width modulation (PWM) module, thus effectively regulating converter operation. As shown in Figure

1, PV energy is mainly directed to the loads on the micro-grid and, in the event of excess, to the battery. This strategy ensures precise current and voltage regulation, while maintaining an overall energy balance. It maximizes the use of solar energy while ensuring optimum battery performance during charging and discharging.

The structure of the control circuit is shown in Figure 4, and is based on a PI control loop which generates the appropriate duty cycle to track V_{ref} (fixed at 1000 V). This mechanism allows control of battery charging and discharging operations by leaving the battery current free, unlike a cascade loop controller where this degree of freedom is restricted. The coefficients k_p and k_i of the proportional-integral controllers for PV current and DC bus voltage were determined by (6) to optimize the dynamic response of the system. Table 3 shows the optimal values found for these parameters. The performances obtained were validated by simulation.

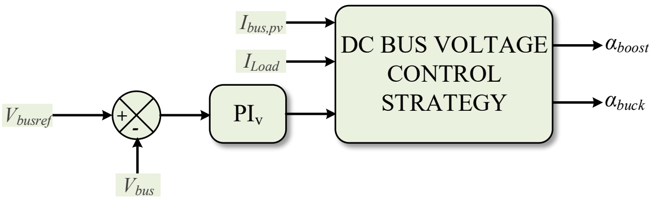


Figure 4. DC-bus voltage control.

$$\begin{cases} k_{pi} = L\cdot\omega_n^2 \\ k_{ii} = L\cdot\omega_n^2\cdot 2\cdot\varepsilon \\ k_{pv} = C\cdot\omega_n^2 \\ k_{iv} = C\cdot\omega_n^2\cdot 2\cdot\varepsilon \end{cases} , \quad \begin{cases} \omega_n = 2\cdot\pi\cdot\dfrac{f}{10} \\ \varepsilon = \dfrac{\sqrt{2}}{2} \\ \tau_{(i,v)} = \dfrac{k_{p(i,v)}}{k_{i(i,v)}} \end{cases} \qquad (6)$$

The system's ω bandwidth is restricted to 10% of the control frequency [5], [7].

TABLE III. PI CONTROLLER PARAMETERS

Coefficients	Values
k_{pi}	0.1
k_{ii}	100
k_{pv}	0.1
k_{iv}	1

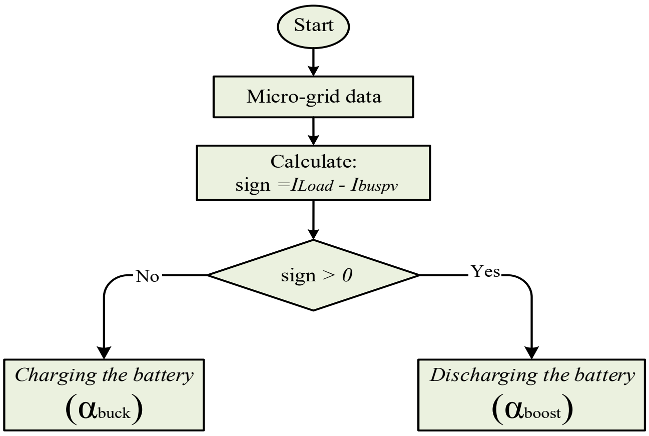


Figure 5. Voltage control strategy.

Figure 5 illustrates the flowchart of the control strategy developed. The battery storage system plays a dual role, both supplying energy and absorbing excess energy. In stand-alone mode, when production exceeds demand, the algorithm commands the battery to charge at a rate compatible with its safety limits, in particular its maximum permissible current. Conversely, in the event of a shortfall in photovoltaic production, the battery supplies the energy required to meet load demand, until it is completely discharged. The energy management system continuously monitors the battery's storage status to ensure safe and efficient operation of the micro-grid.

IV. Simulation Results and Discussion

Following the design of the photovoltaic system, storage device and control strategy, Matlab/Simulink software was used to model and simulate the hybrid power system, with a view to validating the effectiveness of the proposed energy management strategy.

A controlled DC voltage source has been adopted to represent the main DC bus, enabling analysis of system behavior in the presence of line or load disturbances. The simulated weather conditions, illustrated in Figure 6, include the evolution of solar radiation (in W/m²), ambient temperature (in °C) during and the load demanded over a period of 1000 hours. These parameters have a direct influence on the energy output of the photovoltaic modules. The simulated climate profile incorporates realistic scenarios, ranging from optimal sunshine to cloudy episodes and sunset, to assess the robustness of the control strategy in the face of environmental variations, an essential criterion in micro-grid applications.

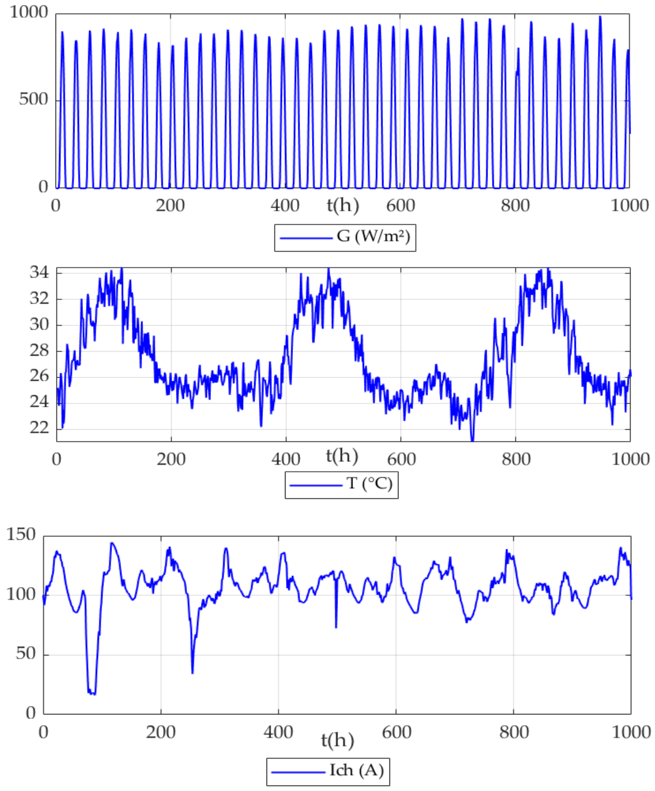


Figure 6. Variations in experimental weather conditions, solar radiation, temperature and load.

The curves in Figure 7 illustrate the temporal evolution of the voltage (V_{pv}) and current (I_{pv}) supplied by the photovoltaic modules. It can be seen that the system efficiently follows variations in irradiation and temperature, ensuring optimum production. This demonstrates the ability of the MPPT algorithm and its control strategy to dynamically adapt the operating point of the PV generator.

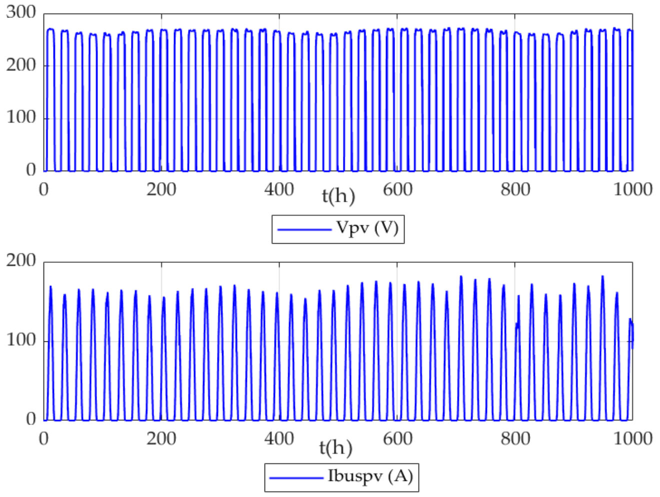


Figure 7. Waveform of the current and voltage on the PV side.

The result of the waveforms shown in the Figure 8 compares the current measured at the output of the photovoltaic array with the reference current generated by the MPPT controller. The precise alignment between the two curves demonstrates the performance of the incremental conductance algorithm, which rapidly adjusts the reference current in response to changes in sunlight, ensuring high energy efficiency.

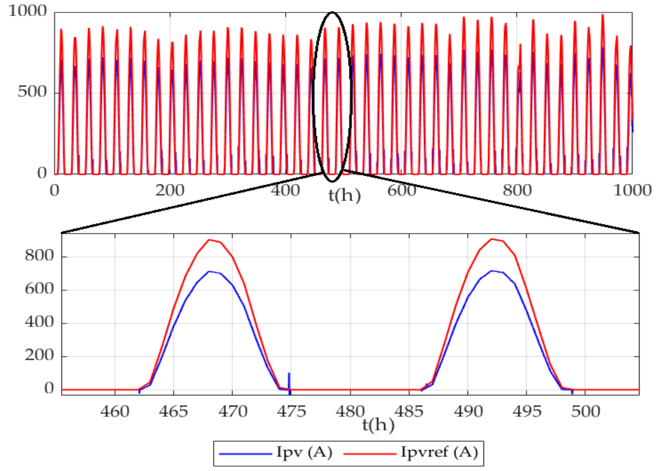


Figure 8. Waveform of the current and its reference on the PV side.

The DC bus voltage is regulated around a setpoint of 1000 V. The evolution of its result over the course of the experiment is illustrated in Figure 9. Thanks to the PI controller and the control strategy, effective regulation is achieved even in the presence of disturbances on the load or generation side. The stability of this voltage is essential to ensure the smooth operation of all equipment interacting in the system, particularly the micro-grid receivers.

The waveform results shown in Figure 10 demonstrate the battery's response to system requirements. Current variations indicate the charging and discharging phases in relation to solar energy availability and load demand. Voltage remains within nominal ranges, reflecting the correct sizing of the bi-directional converter and the reliability of the management system.

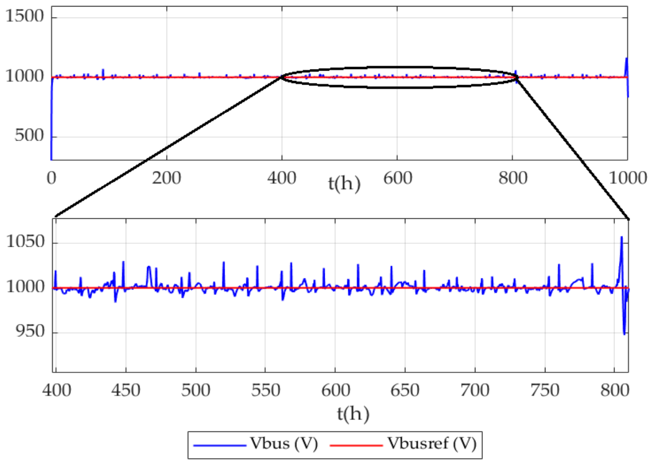


Figure 9. DC-bus voltage.

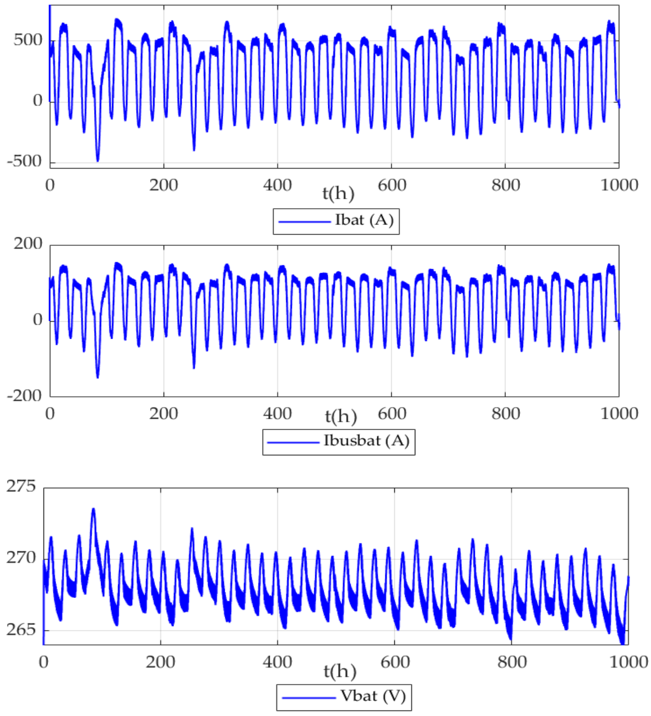


Figure 10. Waveform of the voltage and current on the battery side.

Figure 10 shows the state of charge of the battery pack. It can be seen that during the period of overproduction, the battery's state of charge increases due to the excess solar energy production, conversely, when we are in deficit of production, the battery contributes by discharging to ensure the balance between supply and demand. This confirms that the proposed DC bus voltage control strategy responds effectively to system constraints, with better performance in terms of stability.

The results in Figure 9 and Figure 10 confirm that the proposed DC bus voltage control strategy effectively meets the system's constraints, with better performance in terms of stability.

The resulting waveforms in Figure 11 illustrate the power flows involved in the system. We can see how the energy produced by the PV is shared between the load (micro-grid) and the battery. When PV energy is insufficient, the battery takes over. The EMSC thus ensures efficient dynamic management, avoiding losses and maintaining energy balance.

ISBN: 978-1-68558-311-8

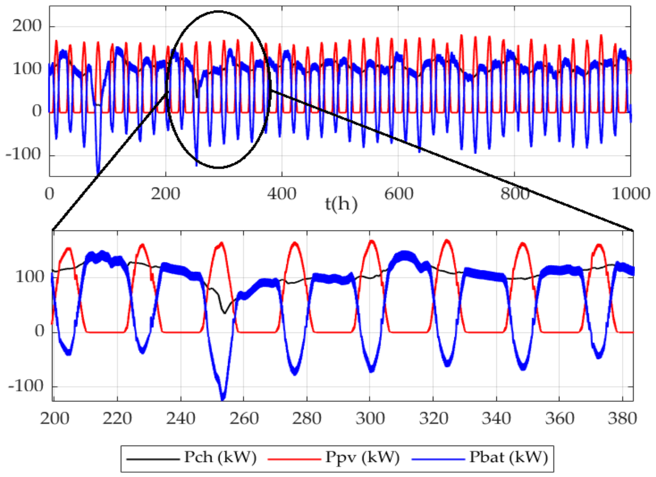


Figure 11. Waveform of the power.

V. Conclusion and Future Work

In this paper, a control and energy management strategy for a stand-alone PV–battery hybrid micro-grid has been proposed, specifically designed for rural electrification in Guinea. By combining an Incremental Conductance MPPT algorithm with Proportional–Integral controllers, the system ensures both maximum photovoltaic energy extraction and robust DC-bus voltage regulation. The proposed approach enables efficient coordination between PV generation and battery storage, thereby improving energy utilization, enhancing system stability, and reducing current stress on the load. Simulation results in Matlab/Simulink under realistic irradiance and load conditions confirm the effectiveness of the proposed strategy, demonstrating stable DC-bus operation at 1000 V, optimized battery charge/discharge management, and improved overall efficiency of the micro-grid. These findings indicate that the proposed control system is well suited for off-grid rural applications where reliability and energy autonomy are critical. Future work will focus on experimental validation using a physical micro-grid test bench, as well as the integration of additional renewable sources and advanced predictive energy management techniques to further enhance system resilience and scalability.

Acknowledgment

This work was carried out at the GREAH laboratory, Université du Havre Normandie, by the Renewable Energies and Storage Systems (MERS) research team, as part of the 1000PhD and 5000 Masters train-the-trainer project initiated by the Ministry of Higher Education, Scientific Research and Innovation of the Republic of Guinea (MESRSI-GUINEE). The authors would like to thank the Guinean authorities for their financial support, and the University of Le Havre Normandie for its pedagogical and institutional support in ensuring the success of this project.

References

[1] A. Chellakhi and S. El Beid, “A Real-Time Investigation of an Enhanced Variable Step PO MPPT Controller for Photovoltaic Systems Using dSPACE 1104 Board”, *Energies*, vol. 18, no. 13, p. 3343, 2025.

[2] M. L. Touré, M. B. Camara, and B. Dakyo, “Symmetrical Multilevel High Voltage-Gain Boost Converter Control Strategy for Photovoltaic Systems Applications”, *Electronics*, vol. 13, no. 13, p. 2565, 2024.

[3] C. B. Nzoundja Fapi et al., “Fuzzy Logic-BasedMaximum Power Point Tracking Control for Photovoltaic Systems: A Review and Experimental Applications”, *Archives of Computational Methods in Engineering*, vol. 32, no. 4, pp. 2405-2428, 2025.

[4] C. H. Swetha, N. S. Jayalakshmi, K. M. Bhargavi, and P. B. Nempu, 2019, “Control strategies for power management of PV/battery system with electric vehicle”, In *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)* pp. 1-6, 2019, IEEE.

[5] I. Oukkacha, M. B. Camara, and B. Dakyo, “Energy Management in Electric Vehicle based on Frequency sharing approach, using Fuel cells, Lithium batteries and Supercapacitors”, in *Inter. Conf. on Renewable Energy Research and Applications (ICRERA2018),* Paris, France, pp. 986-992, 2018, IEEE.

[6] M. O. Badawy and Y. Sozer, “Power flow management of a grid tied PV-battery system for electric vehicles charging”, *IEEE Transactions on Industry Applications*, vol. 53, no. 2, pp. 1347-1357, 2026.

[7] A. Baqar, M. B. Camara, and B. Dakyo, “Supercapacitors Fast Ageing Control in Residential Microgrid Based Photovoltaic/Fuel Cell/Electric Vehicle Charging Station”, *Energies*, vol. 16, no. 13, p. 5084, 2023.

[8] M. P. Bonkile and V. Ramadesigan, “Power management control strategy using physics-based battery models in standalone PV-battery hybrid systems”, *Journal of Energy Storage*, vol. 23, pp. 258-268, 2019.

[9] S. Ahmad, M. Shafiullah, C. B. Ahmed, and M. Alowaifeer, “A review of microgrid energy management and control strategies”, *IEEE Access*, vol. 11, pp. 21729-21757, 2023.

[10] J. Hu, Y. Shan, K. W. Cheng, and S. Islam, “Overview of power converter control in microgrids—Challenges, advances, and future trends”, *IEEE Transactions on Power Electronics*, vol. 37, no. 8, pp. 9907-9922, 2022.

[11] S. J. Yaqoob et al., “An optimal energy management strategy for a photovoltaic/li-ion battery power system for DC microgrid application”, *Frontiers in energy research*, vol. 10, p. 1066231, 2023.

[12] Y. Alidrissi, R. Ouladsine, A. Elmouatamid, and M. Bakhouya, “An energy management strategy for DC microgrids with PV/battery systems”, *Journal of Electrical Engineering & Technology*, vol. 16, pp. 1285-1296, 2021.

[13] M. Benzaouia, A. Rabhi, B. Hajji, S. Benzaouia, H. Midavaine, and B. K. Oubbati, “Real-Time Control and Power Management Strategies of PV/Battery Standalone System”, *IFAC-Papers OnLine*, vol. 56, no. 2, pp. 9135-9140, 2023

[14] C. B. Nzoundja Fapi, M. L. Touré, M. B. Camara, and B. Dakyo, “Control Strategy for DC Micro-Grids in Heat Pump Applications with Renewable Integration”, *Electronics*, vol. 14, no. 1, p. 150, 2025.

[15] S. R, Fahim, H. M. Hasanien, R. A. Turky, S. H. A. Aleem, and M. Ćalasan, “A comprehensive review of photovoltaic modules models and algorithms used in parameter extraction”, *Energies*, vol. 15, no. 23, p. 8941, 2022.

[16] M. B. Camara, H. Gualous, F. Gustin, A. Berthon, and B. Dakyo, “DC/DC converter design for supercapacitor and battery power management in hybrid vehicle applications—Polynomial control strategy”, *IEEE transactions on industrial electronics*, vol. 57, no. 2, pp. 587-597, 2009.

[17] H. Rezk, A. M. Nassef, M. A. Abdelkareem, A. H. Alami, and A. Fathy, “Comparison among various energy management strategies for reducing hydrogen consumption in a hybrid fuel cell/supercapacitor/battery system”, *International Journal of Hydrogen Energy*, vol. 46, no. 8, pp. 6110-6126, 2021.

[18] B. Benlahbib et al., “Experimental investigation of power management and control of a PV/wind/fuel cell/battery hybrid energy system microgrid”, *International Journal of Hydrogen Energy*, vol. 45, no. 53, pp. 29110-29122, 2020.

[19] M. Danandeh and S. M. Mousavi, “Comparative and comprehensive review of maximum power point tracking methods for PV cells”, *Renewable and Sust. Energy Reviews*, vol. 82, pp. 2743-2767, 2018.

[20] M. L. Katche, A. B. Makokha, S. O. Zachary, and M. S. Adaramola, “A comprehensive review of maximum power point tracking (mppt) techniques used in solar pv systems”, *Energies*, vol, 16, no. 5, p. 2206, 2023.

Electric Behavior Characterization of Lithium Iron Phosphate Batteries as a Function of Operating Temperature

Ibrahima Touré[1,2,] Joselyn Stephane Menye[1], Alireza Payman[1], Mamadou-Baïlo Camara[1,2] and Brayima Dakyo[1]
[1]GREAH-Laboratory, University of Le Havre Normandie; [2]Research Laboratory in Applied Sciences of Mamou (LaReSA)
Le Havre, France; Mamou, Guinea
ibrahima.toure@etu.univ-lehavre.fr (I.T.); joselyn-stephane.menye@univ-lehavre.fr (J-S.M); alireza.payman@univ-lehavre.fr (A.P.); mamadou-bailo.camara@univ-lehavre.fr (M.B.C.); brayima.dakyo@univ-lehavre.fr (B.D.)

***Abstract*— This paper presents evaluation of temperature variations on electric behavior of iron phosphate batteries. Indeed, the aim of this study is to show the effects of operating temperature on the series resistances and the capacities of the electrical model. More precisely, the paper's contribution focuses on the study of the degradation of lithium iron phosphate batteries parameters as a function of the temperature for increasing and decreasing phases of temperature. To determine the batteries, charge and discharge capacities and series resistance, experimental characterization is carried out by using different predetermined protocols. That leads to determination of electrical model parameters under various temperature conditions.**

Keywords*— *Lithium iron phosphate batteries; Electrical characterization; Temperature; Series resistance; Battery capacity.

I. INTRODUCTION

The energy crisis underlies many of the challenges and opportunities facing the world today. All energy production sources have drawbacks, including air pollution, accidents, and greenhouse gas emissions [1]. Renewable energy sources are emerging as serious contenders for fossil fuel substitution, helping to reduce greenhouse gas emissions. In a global context marked by ambitious renewable energy targets, their deployment has intensified across industrial, commercial, public, and residential sectors [2][3]. However, the intermittent nature of these resources, heavily dependent on weather conditions and changing seasons, underscores the crucial role of efficient energy storage systems in promoting the widespread adoption of renewable energy technologies in homes [2]. Storage systems now play a central role in integrating renewable energy sources into the traditional energy market while ensuring a stable and reliable power supply in smart grids [4].

Of all energy storage technologies, lithium batteries (LIBs) are the most widely used in industry today. They serve a broad range of applications, from smartphones to aerospace and electric vehicles [5]-[7]. Among the various types, the lithium iron phosphate (LiFePO4, LFP) battery is particularly popular due to its thermal stability and low cost compared to other technologies.

Despite their central role, LIBs face several limitations and constraints. Numerous studies have demonstrated that temperature plays a critical role in the aging and failure of LIBs [8][9]. These effects are often characterized by failures such as thermal runaway and aging, which are based on variations in the components' capacitance and internal resistances. Many studies have evaluated the impact of operating temperature on battery series resistances and capacity, primarily focusing on test protocols involving charge and discharge cycles.

In the work conducted by Yue et al. [10], tests were carried out using three types of batteries over a wide temperature range, from -50°C to 50°C. The results indicate that ohmic resistance increases significantly as the temperature decreases, particularly below -30°C, a phenomenon attributed to the increased viscosity of the electrolyte. Among the technologies evaluated, LFP batteries exhibited the lowest resistance and were the least sensitive to temperature variations.

In another study [11], the authors focused on characterizing and modeling the aging of LFP batteries under the combined effects of temperature and DC current ripple frequency. The tests were based on experimental data covering 4,800 cycles, with frequency variations from 50 to 500 mHz and temperature variations from 10 to 80°C. The results reveal that series resistance increases with frequency but decreases with rising temperature. Conversely, energy capacity increases with both temperature and frequency.

In [12], Ahmed et al. studied two types of batteries at different temperatures, demonstrating that cell resistance increases significantly at low temperatures. Their analysis also revealed that the interfacial resistance of the anode is nearly twice that of the cathode, highlighting its predominant role in ohmic losses.

In [13], the research investigates the influence of cathode material and temperature on the discharge capacity of LIBs. They found that as temperature rises, electrolytic activity changes, leading to an initial increase followed by a decrease in discharge capacity. For LFP technology, correlations have been established between capacity, internal resistance, ambient temperature, and state of charge. At extreme temperatures ($T \geq 50°C$ or $T \leq 20°C$), capacity decreases. However, as long as the temperature remains above 0°C, capacity stays above 93.4%, before dropping significantly below this threshold. Previous work [14][15] has highlighted the contribution of temperature to battery aging, with its effects often studied alongside other factors, such as charge/discharge current or State of Charge (SoC).

In this paper, we propose an analysis focused exclusively on the influence of temperature on ohmic resistance as well as on charge and discharge capacities. A comparative approach is adopted between two thermal profiles: one with increasing temperature and the other with decreasing temperature.

The structure of this paper is designed to present the experimental approach and the results obtained in a progressive and methodical manner. Section II details the experimental setup and the general conditions under which the

ISBN: 978-1-68558-311-8

tests were conducted, ensuring the reproducibility and reliability of the measurements. Section III outlines the methodology used to estimate the main electrical parameters of LIBs, specifically series resistance and capacity, from the experimental data. Section IV presents the results obtained and provides an in-depth analysis of the observed effects, highlighting the correlations between the measured parameters and the test conditions. Finally, Section V summarizes the main conclusions of the study and suggests perspectives for future work in the modeling and characterization of electrochemical storage systems.

II. Description of Experimental Setup and General Tests Conditions

A. Description of the test environment:

The experimental test bench used in this study is illustrated in Figure 1 and comprises a climatic chamber (model ARS-0220), a battery cycler (BT2000/ARBIN BT-ML) and a real-time control system (MITS-PRO). The NI cDAQ-9174 module acquires and records the temperature, the batteries currents and the cells terminal voltages.

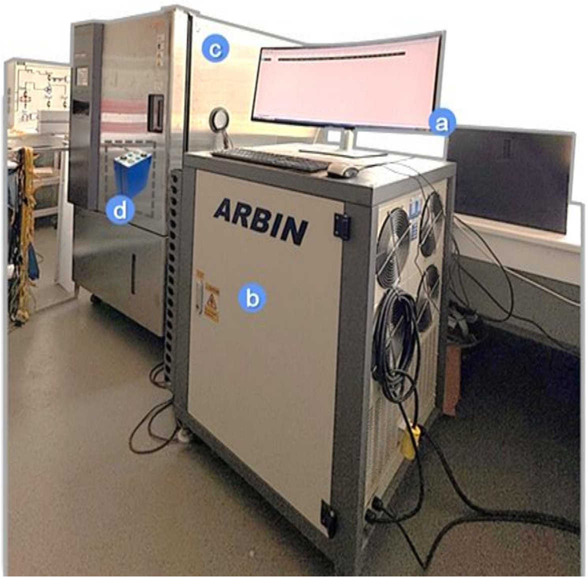


Figure 1. Test bench: (a) Computer running MITS-Pro software, (b) Battery cycler, (c) Climatic chamber (d) Battery pack.

The batteries are subjected to thermal stress during the charging and discharging phases, imposed by a 120-liter environmental chamber with an operating range from -75°C to 180°C. The test bench is controlled via an RS232 interface, allowing for synchronization of events between the BT2000 cycler and the environmental chamber.

Tests were conducted on LF50K cells (3.2V, 50Ah), which have a nominal capacity of 50 Ah. The proposed analysis method is generic and can be adapted to other battery technologies without modification of the experimental protocol, provided the temperature ranges recommended by manufacturers are adhered to.

Tests consist of periodic voltage cycles applied to the batteries, ranging from 2.5 V to 3.5 V and vice versa, under the influence of either fluctuating or constant DC current. These charge and discharge cycles are continuously applied, with no interruptions between different phases.

To ensure accurate estimation of the battery's capacity and its series resistance, the measurements are carried out according to the following protocols:

- Temperatures are measured using high-precision sensors (error ≤ ±0.1%).
- Voltages are measured directly at the cell terminals, thus avoiding disturbances induced by power cables.

B. Test protocol

It is crucial to design an accurate test protocol; otherwise, incorrect results may arise. In our study, we developed a characterization algorithm to determine the battery's ampere-hour capacity (Q[Ah]) and series resistance. To ensure that the tests are conducted properly and that the correct parameters are accurately collected, we designed an algorithm that incorporates all elements of the protocol. This algorithm is tailored to perform two types of tests:

1) A Constant Current/Constant Voltage (CCCV) charging method to charge and discharge the batteries.

2) In the same program, a second test allows us to determine the series resistance.

Both algorithm tests were implemented through MITS-PRO, an ARBIN Group data acquisition device used as a communication interface, as shown in Figure 1. Measurements are performed with a sampling time of one point per second for the capacity and one point per 0.001 second for the series resistance.

The characterization protocol was studied over an operating temperature range of [-5°C to 55°C] with a charging current of 32 A and a discharging current of -32 A. The two experiments were conducted as follows:

First, the battery is subjected to the test temperature for one hour to allow it to equilibrate with the temperature of the climatic chamber. It is then charged with a constant positive current until the voltage at its terminals reaches the maximum set voltage of 3.55 V, which has been established for both testing purposes and the safety of the module.

Before discharging, the battery is allowed to rest for 30 seconds. A negative current is then applied to discharge the battery until its voltage reaches 2.5 V, the minimum test voltage. At the end of the characterization test, a rest period of 2 hours is set to allow the battery voltage to stabilize. This stabilization period enables us to determine the capacity and resistance of the electrolyte, which are essential for calculating the time constant. Figure 2 provides an illustration of the general protocol of the tests.

As illustrated in Figure 2 and in accordance with the protocol defined in the test algorithm, three distinct phases can be identified, each corresponding to a specific stage in the battery characterization process.

Phase A represents the thermal stabilization period. During this phase, the battery is placed in the climate chamber and maintained at the set temperature for a sufficient duration to ensure thermal homogeneity within the cell. In our protocol, this duration is set to one hour, allowing the battery to reach thermal equilibrium with the environment, thereby guaranteeing the reliability of subsequent measurements.

Phase B corresponds to a complete battery charge/discharge cycle. The exact sequence (charge followed by discharge or vice versa) is determined by the algorithm's internal logic. The total duration of this cycle is highly dependent on ambient temperature, as the electrochemical properties of the battery, such as capacity and internal

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-311-8

resistance, vary significantly with temperature. This step is essential for assessing the battery's energy behavior under different operating conditions.

Phase C is dedicated to determining the battery's series resistance. Unlike the previous phase, this stage does not require a long-term test. It relies on the application of a constant charging current for a short period, followed by a rest phase. In our study, a constant current was applied for 10 minutes, followed by a 5-minute rest period between stages. This method allows us to measure series resistance efficiently and accurately, without placing excessive strain on the battery.

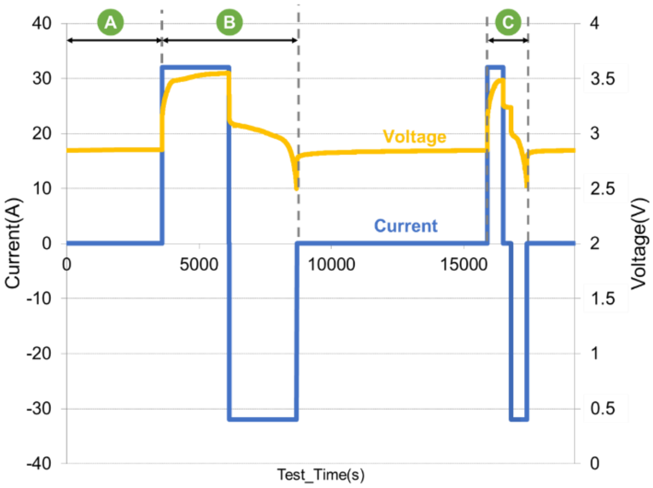


Figure 2. General protocol of the tests.

The sequence of these three phases constitutes the characterization protocol for obtaining the parameters needed to analyze the battery's dynamic behavior.

To determine series resistance, a short-term test was carried out. A constant charging current of 32 A was first applied for a period of 10 minutes. This was followed by a rest period of 5 minutes, before reversing the current to carry out the discharge phase under the same conditions.

III. Calculation Of Battery Electrical Parameters

A. *Behavior model of the battery*

Numerous battery models have been proposed in the literature to simulate their dynamic behavior. Most of these models are based on equivalent fixed-parameter electrical circuits, typically composed of constant resistances and capacitances [11][14]. However, such models may be insufficient to accurately represent the actual behavior of batteries, particularly when their characteristics—such as internal resistance, capacity, or SoC—vary over time.

A more realistic model of battery dynamic behavior can be achieved by considering the dependence of these parameters on SoC, as illustrated in Figure 3. Unfortunately, most existing models neglect the evolution of these parameters as a function of temperature during charge and discharge cycles, which limits their accuracy during prolonged simulations or under variable stress conditions.

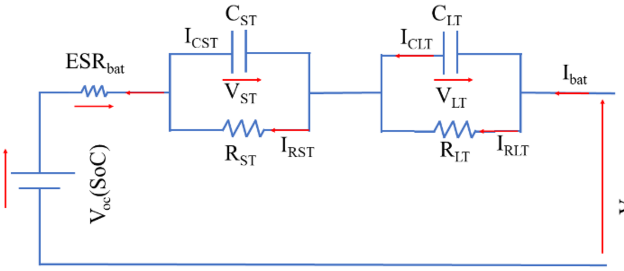


Figure 3. Basic model of the LFP-battery.

The time constants $R_{ST}C_{ST}$ and $R_{LT}C_{LT}$ are estimated based on a detailed analysis of the transient evolution of the battery terminal voltage, conducted immediately after the end of a charging phase or at the very beginning of a discharging phase. This approach accurately captures the dynamic response of the battery under these specific conditions, highlighting the resistive and capacitive components associated with various internal electrochemical phenomena [16][17].

B. *Battery's parameters identification*

To determine the parameters of the batteries, it is first necessary to extract the data recorded by the ARBIN system's acquisition software. Among the collected data, a column indicating capacity allows for the direct identification of the batteries' capacity in the table provided by the software. This section presents the method for calculating series resistance from experimental data (current and voltage) obtained during charging and discharging operations.

If the battery does not exhibit hysteresis behavior, the series resistance can be estimated based solely on the experimental data from charging, as the series resistance during discharging will be identical to that obtained during charging. To estimate series resistance from the charging operations, the experimental voltage and current data shown in Figure 4, combined with Equation 1, are used.

Conversely, if the battery exhibits hysteresis behavior, the series resistance must be estimated separately using the charge and discharge data presented in Figures 4 and 5, respectively. Thus, the voltage drops obtained from these figures, noted ΔV_{RC} and ΔV_{RD}, respectively, can be integrated into Equation 1 to calculate the series resistance corresponding to the battery's charge and discharge phases. The series resistance calculated during charging and discharging operations with a constant current of ±50 A for each temperature is shown in Figures 4 and 5.

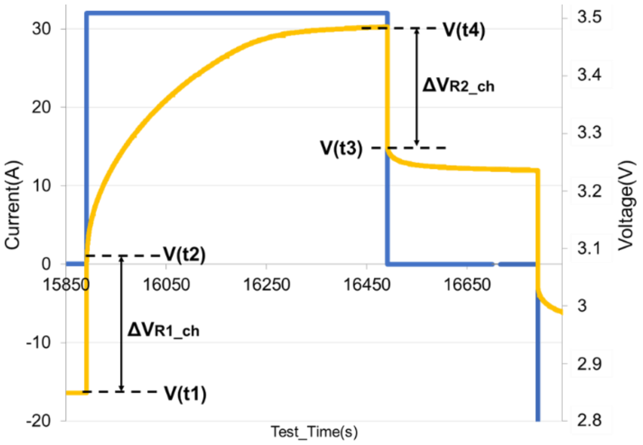


Figure 4. Voltage and current obtained during a charge operation with a constant current of $I_{bat} = 50$A.

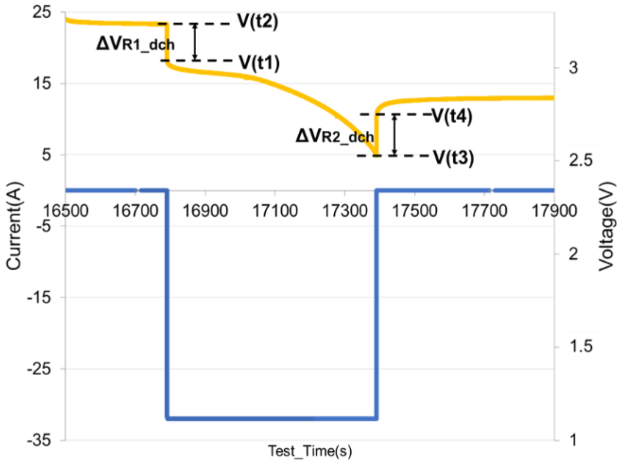


Figure 5. Voltage and current obtained during a discharge operation with a constant current of I_{bat} = -50A.

$$R_i = \begin{cases} R_C = \frac{\Delta V_{RC}}{I_{bat}} & if\ \ I_{bat} > 0 \\ R_D = \frac{\Delta V_{RD}}{I_{bat}} & if\ \ I_{bat} < 0 \end{cases} \quad (1)$$

Capacity identification is based on experimental battery voltage and current data, as a function of charging or discharging time. In order to consider into account any differences in parameters between charging and discharging phases, parameter identification must be separately carried out for each operating mode. Figure 6 illustrates a single charge case based on experimental data obtained by measuring voltage, current and response time.

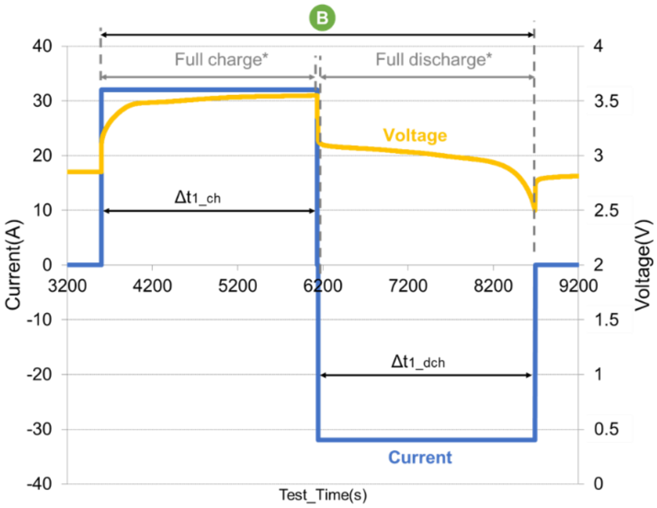


Figure 6. Calculation curve for load capacity/discharge

Battery capacities during charging Q_{cell_ch} and discharging Q_{cell_di} are determined using Equation 2.

$$Q_{cell_{ch}/di} \approx \int_{\tau_1}^{\tau_2} I_{bat}.dt \quad (2)$$

Where τ_1 corresponds to the start time of charging/discharging, and τ_2 to the final time of this operation.

IV. Results and Discussion

After identifying capacity values at different temperatures, a comparative analysis was carried out to assess the thermal effect on the battery's electrochemical behavior. Figure 7 illustrates the evolution of capacities in charge and discharge, as a function of the temperature, considered in both increasing and decreasing scenario. Observation of the curves reveals a tendency for increasing the capacity as temperature rises. This behavior can be explained by improved electrochemical kinetics. This trend is clearly visible in Figure 7, which shows that the capacity extracted is higher at higher temperatures, whether charging or discharging. Conversely, a decrease in temperature is accompanied by a significant drop in measured capacities, reflecting a marked thermal sensitivity of battery's performance.

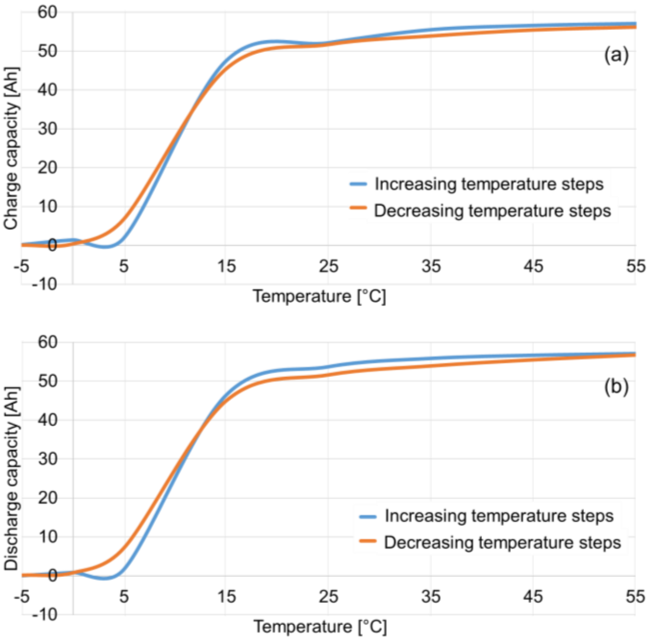


Figure 7. Battery charge and discharge capacity at rising and falling temperatures.

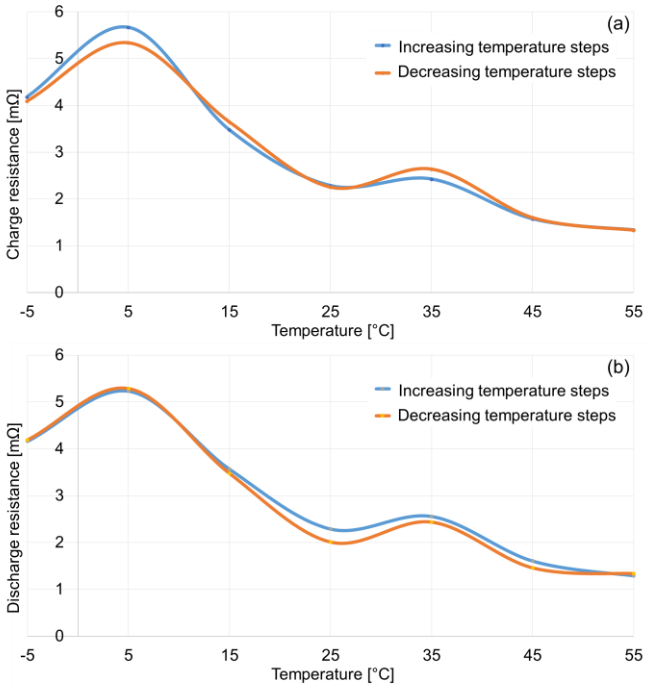


Figure 8. Battery charge and discharge series resistance at rising and falling temperatures

The carried-out calculations identify two values for series resistance, corresponding to the charging and the discharging phases, under the same temperature conditions as those used in the previous test. The analysis of the results, illustrated in Figure 8, reveals a tendency for series resistance to decrease with increasing temperature, reflecting the classic thermal behavior of electrochemical materials. This can be explained by a reduction in the cell's internal resistivity at high temperatures, facilitating charge transport. However, it is also observed that, for each given temperature, the resistance values obtained in charge and discharge are identical. This symmetrical behavior suggests the absence of any significant hysteresis effect on series resistance under the considered experimental conditions.

V. Conclusion and future work

The analysis of the experimental curves for series resistance and capacity, expressed in ampere-hours (Ah), clearly and unambiguously demonstrates the significant impact of the temperature on the electrical behavior of the battery. In particular, it is observed that at low temperatures, the battery's performance deteriorates considerably. This degradation is reflected in a sharp increase in internal resistance, indicating a reduction in ionic conductivity within the electrolyte, as well as a substantial decrease in the available capacity. In fact, under such conditions, the measured capacity remains well below the nominal value specified by the manufacturer, failing to reach even half of it. These findings confirm that low temperatures directly affect the battery's internal electrochemical processes, thereby limiting its energy efficiency and its ability to deliver adequate current during operation. Consequently, these results highlight the critical importance of accounting for temperature effects in the modeling, thermal management, and optimal operation of electrochemical energy storage systems.

References

[1] H. A. Gabbar, M. R. Abdussami, and Md. I. Adham, "Optimal Planning of Nuclear-Renewable Micro-Hybrid Energy System by Particle Swarm Optimization," *IEEE Access*, vol. 8, pp. 181049–181073, 2020, doi: 10.1109/ACCESS.2020.3027524.

[2] L. Apa, L. D'Alvia, Z. Del Prete, and E. Rizzuto, "A Characterization of the Uncertainties Associated With an Automated System for the Study of Lithium-Ion Cells: A Case-Study of a Domestic Grid 24-h Scenario," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024, doi: 10.1109/TIM.2024.3476559.

[3] F. Ricco Galluzzo *et al.*, "Electrical Characterization and Modeling of an Innovative Acid/Base Flow Battery," *IEEE Access*, vol. 12, pp. 185200–185211, 2024, doi: 10.1109/ACCESS.2024.3512994.

[4] K. Li, F. Wei, K. J. Tseng, and B.-H. Soong, "A Practical Lithium-Ion Battery Model for State of Energy and Voltage Responses Prediction Incorporating Temperature and Ageing Effects," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6696–6708, Aug. 2018, doi: 10.1109/TIE.2017.2779411.

[5] Y. Wang, H. Li, P. He, E. Hosono, and H. Zhou, "Nano active materials for lithium-ion batteries," *Nanoscale*, vol. 2, no. 8, pp. 1294–1305, Aug. 2010, doi: 10.1039/C0NR00068J.

[6] H. Sharma, S. Sharma, and P. K. Mishra, "A critical review of recent progress on lithium ion batteries: Challenges, applications, and future prospects," *Microchem. J.*, vol. 212, p. 113494, May 2025, doi: 10.1016/j.microc.2025.113494.

[7] R. Kumar and K. Das, "Lithium battery prognostics and health management for electric vehicle application – A perspective review," *Sustain. Energy Technol. Assess.*, vol. 65, p. 103766, May 2024, doi: 10.1016/j.seta.2024.103766.

[8] S. Ma *et al.*, "Temperature effect and thermal impact in lithium-ion batteries: A review," *Prog. Nat. Sci. Mater. Int.*, vol. 28, no. 6, pp. 653–666, Dec. 2018, doi: 10.1016/j.pnsc.2018.11.002.

[9] J. S. Menye, M.-B. Camara, and B. Dakyo, "Lithium Battery Degradation and Failure Mechanisms: A State-of-the-Art Review," *Energies*, vol. 18, no. 2, p. 342, Jan. 2025, doi: 10.3390/en18020342.

[10] Y. Yue *et al.*, "Effects of temperature on the ohmic internal resistance and energy loss of Lithium-ion batteries under millisecond pulse discharge," *J. Phys. Conf. Ser.*, vol. 2301, no. 1, p. 012014, July 2022, doi: 10.1088/1742-6596/2301/1/012014.

[11] K. Bellache, M. B. Camara, B. Dakyo, and S. Ramasamy, "Aging Characterization of Lithium Iron Phosphate Batteries Considering Temperature and Direct Current Undulations as Degrading Factors," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, pp. 9696–9706, Oct. 2021, doi: 10.1109/TIE.2020.3020021.

[12] S. Hossain, X. Kang, and S. Shrestha, "Effects of Temperature on Internal Resistances of Lithium-Ion Batteries," *J. Energy Resour. Technol.*, vol. 137, p. 031901, May 2015, doi: 10.1115/1.4028698.

[13] S. Lv, X. Wang, W. Lu, Z. Jiaqiao, and H. Ni, "The Influence of Temperature on the Capacity of Lithium Ion Batteries with Different Anodes," *Energies*, vol. 15, p. 60, Dec. 2021, doi: 10.3390/en15010060.

[14] B. Kosseila, M. B. Camara, and B. Dakyo, *Characterization and Electric Behavior Modeling of Lithium- Battery using Temporal Approach for Parameters Computing*. 2018, p. 1335. doi: 10.1109/ICRERA.2018.8566742.

[15] Z. Ling *et al.*, "Review on thermal management systems using phase change materials for electronic components, Li-ion batteries and photovoltaic modules," *Renew. Sustain. Energy Rev.*, vol. 31, pp. 427–438, Mar. 2014, doi: 10.1016/j.rser.2013.12.017.

[16] T. Mesbahi, N. Rizoug, P. Bartholomeus, R. Sadoun, F. Khenfri, and P. Le Moigne, "Dynamic Model of Li-Ion Batteries Incorporating Electrothermal and Ageing Aspects for Electric Vehicle Applications," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1298–1305, Feb. 2018, doi: 10.1109/TIE.2017.2714118.

[17] P. Pillai, J. Nguyen, and B. Balasingam, "Performance Analysis of Empirical Open-Circuit Voltage Modeling in Lithium-Ion Batteries, Part-2: Data Collection Procedure," *IEEE Trans. Transp. Electrification*, vol. 11, pp. 153-162, Jan. 2024, doi: 10.1109/TTE.2024.3386910.