



IntechOpen

IntechOpen Series
Artificial Intelligence, Volume 47

Multi-Agent Systems

From Basic Concepts to Cutting-Edge
Technologies

*Edited by Nohaidda Sariff, Zool Hilmi Ismail,
Denesh Sooriamoorthy and Puteri Nor Aznie Fahsyar*



Multi-Agent Systems - From Basic Concepts to Cutting-Edge Technologies

*Edited by Nohaidda Sariff,
Zool Hilmi Ismail, Denesh Sooriamoorthy
and Puteri Nor Aznie Fahsyar*

Published in London, United Kingdom

Multi-Agent Systems – From Basic Concepts to Cutting-Edge Technologies

<http://dx.doi.org/10.5772/intechopen.1006408>

Edited by Nohaidda Sariff, Zool Hilmi Ismail, Denesh Sooriamorthy and Puteri Nor Aznie Fahsyar

Contributors

Adel Ben Mnaouer, Andres Mauricio Diaz Caicedo, Damiano Crescini, Denesh Sooriamorthy, Farid Touati, Hasan Tariq, Intan Izafina Idrus, Kok Seng Eu, Md Hasan Molla, Mohammed Abdulla E. Al-Hitmi, Nohaidda Sariff, Puteri Nor Aznie Fahsyar, Swee King Phang, Yahya Muhammad Adam, Zool Hilmi Ismail

© The Editor(s) and the Author(s) 2026

The rights of the editor(s) and the author(s) have been asserted in accordance with the Copyright, Designs and Patents Act 1988. All rights to the book as a whole are reserved by INTECHOPEN LIMITED. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECHOPEN LIMITED's written permission. Enquiries concerning the use of the book should be directed to INTECHOPEN LIMITED rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 4.0 License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in London, United Kingdom, 2026 by IntechOpen

IntechOpen is the global imprint of INTECHOPEN LIMITED, registered in England and Wales,

registration number: 11086078, 167-169 Great Portland Street, London, W1W 5PF, United Kingdom

For EU product safety concerns: IN TECH d.o.o., Prolaz Marije Krucifikse Kozulić 3, 51000 Rijeka, Croatia, info@intechopen.com or visit our website at intechopen.com.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

Additional hard and PDF copies can be obtained from orders@intechopen.com

Multi-Agent Systems – From Basic Concepts to Cutting-Edge Technologies

Edited by Nohaidda Sariff, Zool Hilmi Ismail, Denesh Sooriamorthy and Puteri Nor Aznie Fahsyar
p. cm.

This title is part of the Artificial Intelligence Book Series, Volume 47

Topic: Multi-Agent System

Series Editor: Andries Engelbrecht

Topic Editor: Mehmet Emin Aydin

Print ISBN 978-1-83635-148-1

Online ISBN 978-1-83635-147-4

eBook (PDF) ISBN 978-1-83635-149-8

ISSN 2633-1403

If disposing of this product, please recycle the paper responsibly.

IntechOpen Book Series

Artificial Intelligence

Volume 47

Aims and Scope of the Series

Artificial Intelligence (AI) is a rapidly developing multidisciplinary research area that aims to solve increasingly complex problems. In today's highly integrated world, AI promises to become a robust and powerful means for obtaining solutions to previously unsolvable problems. This Series is intended for researchers and students alike interested in this fascinating field and its many applications.

Meet the Series Editor



Andries Engelbrecht received the Masters and Ph.D. degrees in Computer Science from the University of Stellenbosch, South Africa, in 1994 and 1999 respectively. He is currently appointed as the Voigt Chair in Data Science in the Department of Industrial Engineering, with a joint appointment as Professor in the Computer Science Division, Stellenbosch University. Prior to his appointment at Stellenbosch University, he has been at the University of Pretoria, Department of Computer Science (1998-2018), where he was appointed as South Africa Research Chair in Artificial Intelligence (2007-2018), the head of the Department of Computer Science (2008-2017), and Director of the Institute for Big Data and Data Science (2017-2018). In addition to a number of research articles, he has written two books, *Computational Intelligence: An Introduction* and *Fundamentals of Computational Swarm Intelligence*.

Meet the Volume Editors



Dr. Nohaidda Sariff received her Ph.D. in Control Robotics from Universiti Teknologi Malaysia (UTM) and her M. Sc. in Electrical Engineering from Universiti Teknologi MARA (UiTM), Malaysia. She is currently a Senior Lecturer at the School of Engineering, Faculty of Innovation and Technology, Taylor's University, Malaysia. Dr. Nohaidda is a Registered Professional Engineer with the Board of Engineers Malaysia and a Registered Professional Technologist with the Malaysian Board of Technologists. She has an active publication record, contributing to peer-reviewed indexed journals, conference proceedings, and book chapters. In addition, she serves as a reviewer for both local and international journals and as a judge for innovation- and technology-related competitions. Her research interests focus on control robotics, mobile robot navigation, path planning, and artificial intelligence.



Zool Hilmi Ismail holds a Ph.D. in Electrical Engineering from Heriot-Watt University, and B. Eng and M. Eng degrees in mechatronics engineering from Universiti Teknologi Malaysia (UTM). He is currently an associate professor at UTM, a research member at the Center for AI and Robotics, and has held visiting research positions at Kyoto University, Jordan University of Science & Technology, and Tokyo City University. Dr. Zool is a registered professional engineer and a chartered marine engineer. His research focuses on learning control, digital-twin simulation, multi-agent systems, and unmanned vehicles. He has received numerous awards from International RoboCup Competitions and serves on the technical committees for both the International RoboCup@Home Competition and the World Robot Summit.



Dr. Puteri Nor Aznie Fahsyar is an Assistant Professor in Electrical and Electronic Engineering at the Higher Colleges of Technology, Abu Dhabi, UAE. She holds a Ph.D. in Renewable Energy and is a registered Professional Engineer with multiple international certifications. Her research portfolio spans renewable energy systems, perovskite solar cells, microbial fuel cells, power systems, and intelligent energy technologies. She has secured competitive national and international research grants and has published in high-impact, peer-reviewed journals and international conference proceedings. Her publications advance sustainable energy materials, hybrid energy systems, and low-carbon power technologies, supporting global sustainability goals and regional energy transition initiatives.



Dr. Denesh Sooriamoorthy is a researcher and academic specializing in biomedical engineering, intelligent systems, and data-driven modelling. He obtained his Ph.D. from the University of Nottingham and has worked extensively on machine learning applications for physiological signal analysis, battery health intelligence, and embedded hardware development. He has authored and co-authored numerous articles in Scopus-indexed journals and has con-

tributed actively to industry-linked engineering projects. Dr. Denesh is also involved in curriculum development, engineering education, and professional training in AI and mechatronics. His current research focuses on deep learning frameworks for health diagnostics, EV technologies, and intelligent sensing systems.

Contents

Preface	XIII
Chapter 1	1
Deep Reinforcement Learning for Robot Navigation: Concepts, Current Trends, Challenges, and Future Directions <i>by Nohaidda Sariff, Yahya Muhammad Adam, Intan Izafina Idrus, Zool Hilmi Ismail, Puteri Nor Aznie Fahsyar, Swee King Phang, Kok Seng Eu, Md Hasan Molla and Denesh Sooriamoorthy</i>	
Chapter 2	25
Toward Agentic Artificial Intelligence and Its Societal Implications <i>by Zool Hilmi Ismail, Nohaidda Sariff and Puteri Nor Aznie Fahsyar</i>	
Chapter 3	37
Decentralized Energy Management Using Multi-Agent Systems for Remote Hybrid Power Systems <i>by Puteri Nor Aznie Fahsyar, Nohaidda Sariff and Zool Hilmi Ismail</i>	
Chapter 4	53
Cutting-Edge Technologies in Agent Architecture for Use at the Microgrid Protection Level <i>by Andres Mauricio Diaz Caicedo</i>	
Chapter 5	73
A Multi-Agent Infrastructure Monitoring IoT System for Urban Geo-Resilience Framework <i>by Hasan Tariq, Farid Touati, Mohammed Abdulla E. Al-Hitmi, Damiano Crescini and Adel Ben Mnaouer</i>	

Preface

Multi-agent systems (MAS) have emerged as a fundamental paradigm for modeling, analyzing, and implementing complex, distributed, and intelligent systems. With the rapid advancement of artificial intelligence, robotics, energy systems, and the Internet of Things (IoT), MAS has evolved from foundational theoretical concepts to a key enabling technology for real-world, large-scale, and mission-critical applications. This book, *Multi-Agent Systems – From Basic Concepts to Cutting-Edge Technologies*, aims to present a structured and comprehensive perspective on this evolution.

The book begins by addressing one of the most impactful applications of multi-agent intelligence in robotics. The first chapter explores deep reinforcement learning for robot navigation, discussing fundamental concepts, recent advancements, current challenges, and future research directions. This chapter establishes a strong conceptual and technical foundation, highlighting how learning-based agents can operate autonomously in dynamic and uncertain environments.

The second chapter broadens the scope by examining the transition from traditional agent-based systems toward agentic artificial intelligence, emphasizing autonomy, decision-making, collaboration, and adaptability. In addition to technical perspectives, this chapter critically discusses the societal implications of agentic AI, including ethical, social, and governance considerations, which are increasingly important as intelligent agents become integrated into daily life.

Energy systems form a major application domain for multi-agent systems, particularly in decentralized and remote environments. The third chapter focuses on decentralized energy management using multi-agent systems for remote hybrid power systems, demonstrating how distributed agents can improve efficiency, reliability, and resilience in modern energy infrastructures.

Building on this theme, the fourth chapter presents cutting-edge technologies for agent-based architecture in microgrid protection. This chapter highlights advanced agent designs and coordination mechanisms that enhance protection, fault detection, and decision-making at the microgrid level, showcasing the role of MAS in ensuring the stability and security of smart energy systems.

The final chapter extends the application of multi-agent systems to smart cities and urban resilience. It introduces a multi-agent infrastructure-monitoring IoT system for an urban geo-resilience framework, illustrating how intelligent agents, combined with IoT technologies, can support real-time monitoring, risk assessment, and decision-making for sustainable, resilient urban environments.

This book is intended for graduate students, researchers, academicians, engineers, and professionals working in the fields of multi-agent systems, artificial intelligence,

robotics, energy systems, and smart infrastructure. By combining foundational concepts with advanced applications across multiple domains, this book aims to serve as both a reference and a source of inspiration for future research and innovation.

The editors would like to express their sincere appreciation to all contributing authors for their high-quality chapters and valuable insights. Special thanks are also extended to the IntechOpen editorial team for their continuous support throughout the publication process. We hope this book will contribute meaningfully to the advancement of multi-agent systems research and to their practical adoption in next-generation intelligent systems.

Nohaidda Sariff

School of Engineering,
Taylor's University,
Selangor, Malaysia

Dr. Zool Hilmi Ismail

Associate Professor,
Malaysia-Japan International Institute of Technology,
Universiti Teknologi Malaysia,
Kuala Lumpur, Malaysia

Dr. Puteri Nor Aznie Fahsyar

School of Engineering,
Taylor's University,
Selangor, Malaysia

Dr. Denesh Sooriamoorthy

Assistant Professor,
School of Engineering,
Asia Pacific University of Technology and Innovation (APU),
Kuala Lumpur, Malaysia

Deep Reinforcement Learning for Robot Navigation: Concepts, Current Trends, Challenges, and Future Directions

*Nohaidda Sariff, Yahya Muhammad Adam,
Intan Izafina Idrus, Zool Hilmi Ismail,
Puteri Nor Aznie Fahsyar, Swee King Phang, Kok Seng Eu,
Md Hasan Molla and Denesh Sooriamoorthy*

Abstract

Deep reinforcement learning (DRL) has emerged as a prominent framework in the field of autonomous robot navigation, enabling agents to acquire complex decision-making capabilities and learn optimal policies through continuous interaction with their environment. This chapter provides a comprehensive review of deep reinforcement learning (DRL) in recent robot navigation research within real-time dynamic environments, addressing the gap caused by the limited existing reviews in this area. It begins with fundamental concepts, highlights current trends, discusses key challenges, and concludes with insights into future research directions. Current studies emphasize a shift from static to dynamic environments, improvements in sample efficiency, integration with visual perception, multi-agent systems, multi-objective navigation, and bridging the gap between simulation and real-world applications. These trends underscore the importance of enhancing robot adaptability, learning efficiency, robustness, and scalability, enabling robots to reach their targets while avoiding obstacles effectively. Significant challenges remain, including handling continuous action spaces, designing effective reward functions to balance exploration and exploitation, and addressing learning issues in both dynamic and real-world settings. These challenges will be examined in detail within this review. Furthermore, the chapter will explore future research directions, such as addressing dynamic and actively changing obstacle configurations, integrating DRL with other artificial intelligence techniques, improving learning efficiency across varying scales, and developing strategies for cooperative multi-agent systems. Throughout this review, key limitations and research gaps are identified, with the aim of advancing toward more autonomous, reliable, and scalable DRL-based navigation systems capable of operating effectively and efficiently in real-time environments.

Keywords: deep reinforcement learning, robot navigation, path planning, obstacles avoidance, mobile robot

1. Introduction

Deep reinforcement learning (DRL) is a subset of machine learning. Machine learning focuses on developing algorithms that learn from data to make predictions, make decisions, and detect patterns without explicit programming. Beyond supervised and unsupervised learning, reinforcement learning involves an agent learning to act by interacting with an environment in order to maximize rewards [1]. Deep learning, on the other hand, focuses on representing policies, value functions, or environment models using deep neural networks. In other words, DRL is a revolutionary area of artificial intelligence that combines reinforcement learning with deep neural networks. The primary goal of DRL is to maximize cumulative rewards in complex environments, enabling agents to perform advanced decision-making and acquire sophisticated strategies.

The main components of deep reinforcement learning (DRL) that ensure successful operation include the agent, environment, state s_t , action a_t , reward r_t , and policy. The agent interacts with the environment, while the environment provides the context and feedback to the agent. The state represents the current situation or condition of the environment, based on which the agent selects actions and makes decisions. An action is a signal sent to the actuators, determined by the policy, and the reward serves as feedback indicating the quality of the agent's performance. The policy is the strategy, often represented by a neural network, that maps states to actions, as illustrated in **Figure 1**. Through this structure, DRL can handle high-dimensional, continuous, dynamic, and complex inputs by leveraging sophisticated policies and value functions. This is in contrast to traditional reinforcement learning, which performs well mainly in small, discrete action spaces. Moreover, DRL has the capability to learn directly from raw sensor data and derive optimal policies without requiring an accurate model of the environment.

In the context of robotics, deep reinforcement learning (DRL) enables robots to acquire complex behaviors through interaction with the environment by combining the decision-making capabilities of reinforcement learning with the perceptual strengths of deep learning [3, 4]. Applications of DRL in robotics include mobile robot navigation such as path planning [3] and obstacle avoidance [5, 6] as well as robotic manipulation tasks, including arm control [7], assembly and stacking, quadruped walking [7] and jumping [8], and humanoid balancing [9]. DRL has also been applied to multi-robot coordination for swarm behaviors and cooperative tasks [6], as well as

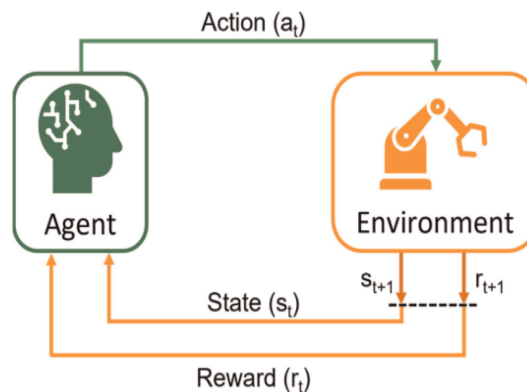


Figure 1. General framework of deep reinforcement learning [2]. *Reprinted from Ref. [2], MDPI, CC BY.

human-robot interaction to develop adaptive and safe behaviors around humans [10]. The contribution of DRL to the advancement of robot navigation research is well documented. For example, Zhang et al. [3] reported an increasing trend in research on path planning using DRL, as opposed to conventional algorithms, based on their review of studies published between 1990 and 2024.

Popular DRL algorithms used for robotic path planning applications include Deep Q-Network (DQN) [11], Deep Deterministic Policy Gradient (DDPG) [12], Proximal Policy Optimization (PPO) [13], and Soft Actor-Critic (SAC) [14]. Two main factors to consider when selecting an algorithm are the type of action space, whether discrete or continuous, and the policy type whether on-policy or off-policy. Other considerations include exploration strategy, sample efficiency, training stability and simplicity, as well as computational complexity. Generally, DQN is most suitable for small, discrete action spaces, DDPG is preferable when deterministic policies are required in continuous action spaces, PPO is valued for its stability and simplicity in both discrete and continuous spaces, and SAC is ideal for high-performance exploration in continuous action spaces with strong sample efficiency.

Figure 2 illustrates how DRL operates in a robotic navigation scenario. The agent, or robot, observes states by interacting with the environment, selects actions, and receives rewards. At each time step, the robot perceives the current state, processes it through a policy network to select an action, and obtains a scalar reward from the environment based on the outcome. This learning process continues until a termination condition is met, such as exceeding time limits, encountering an obstacle, or reaching the goal. To evaluate training duration and policy effectiveness across different navigation scenarios, it is essential to relate the step-by-step actions and feedback to the overall performance metrics measured at the end of a complete run or trial.

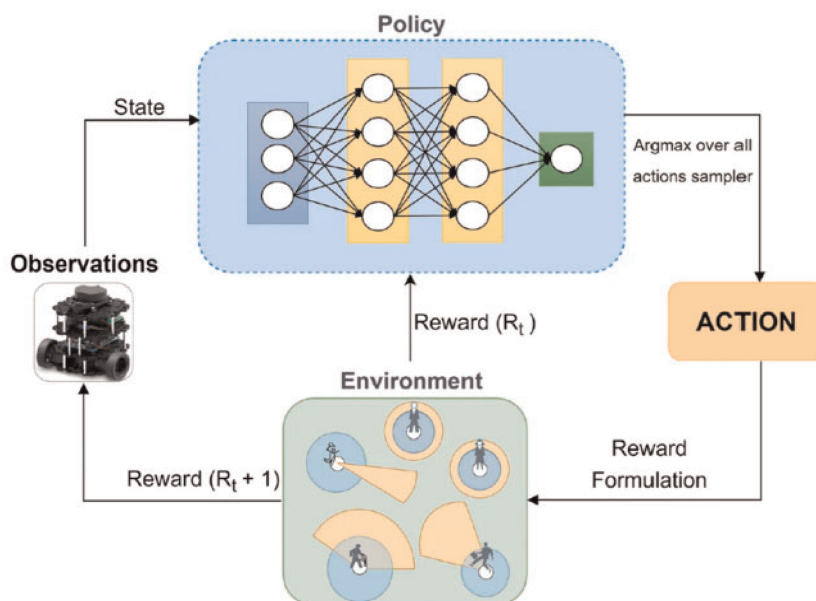


Figure 2. Framework of deep reinforcement learning in robotics research [4]. *Reprinted from Ref. [4], MDPI, CC BY.

With the adaptive self-learning capability of DRL (**Figures 1 and 2**), its use in robotic navigation research has significantly increased. Therefore, further contributions in this area are essential, not only to provide a wider range of DRL algorithms for various robot navigation applications but also to explore the recent, unresolved issues and challenges in DRL. This will help enhance robot performance in real-time dynamic environments, leading to more robust and efficient navigation. The databases used to find the articles were Scopus and ScienceDirect. The search keywords ‘Deep Reinforcement Learning’ and ‘Robot Navigation with Deep Reinforcement Learning’ were used to review related articles for this systematic review.

2. Current research trends in deep reinforcement learning for robotic navigation

2.1 Transition of path planning from static to dynamic environments

Robot navigation initially focused on static environments before advancing to dynamic, real-time scenarios, with various path planning algorithms proposed over time [15, 16]. In situations without prior knowledge of the environment, a robot can determine its local path reactively by relying on sensor information, an approach known as behavior-based navigation [17, 18] or the artificial potential field method [19]. These are considered reactive approaches in the early evolution of mobile robot local path planning [20, 21]. However, such systems have limitations. While they can help a robot avoid obstacles in basic dynamic environments within a local domain, they are prone to issues such as the local minima problem and cannot guarantee the optimality of the global path.

By utilizing information about a known or partially known environment where a complete or partial map is available to the robot, global path planning can be applied in static environments [20, 22]. In such cases, appropriate algorithms are used to determine the robot’s global path on the given map. Common and well-known classical and heuristic-based algorithms include A* [23, 24], D*, distance wave transform, Dijkstra’s algorithm [25], RRT (rapidly exploring random trees) [26–28], and PRM (Probabilistic Roadmap). With the evolution of artificial intelligence, other optimization algorithms have been introduced, such as Genetic Algorithm (GA) [29, 30], Ant Colony Optimization (ACO) [31, 32], and Bee Colony Optimization [33], among others. These algorithms have been shown to produce optimal paths with the shortest travel distance. This pre-route computation allows the robot system to rely more on offline processing rather than online computation. Additionally, various intelligent controllers can be embedded into the system to enhance performance, including fuzzy logic [34–36], neural networks [37, 38], model predictive control (MPC) [39], and proportional-integral-derivative (PID) control [40], and many others.

Recognizing the advantages of having two separate systems, one for global path planning and another for local path planning, many researchers have sought alternative approaches to combine both into a single hybrid system. Such a system enables greater flexibility for robots operating in dynamic environments. In this approach, both systems complement each other: the optimal path to the goal is pre-calculated offline using the global planner before the robot begins moving, while the local reactive method takes over during execution to handle unexpected dynamic obstacles within the environment. As a result, various hybrid methods have been applied in robot navigation, such as improved A* with the Fuzzy Dynamic Window Approach

(FDWA) [41], Particle Swarm Optimization (PSO) combined with Simulated Annealing (SA) [42], and Fuzzy Logic integrated with Artificial Potential Field (APF)) [43]. These methods have demonstrated improvements in optimizing travel time and distance through the global planner while ensuring safe obstacle avoidance via the local planner. Despite these successes, challenges remain, particularly the high computational cost in large-scale dynamic environments, where mapping choices and algorithm selection significantly affect performance. Furthermore, the local minima problem may still occur as obstacle complexity increases, especially in real-time dynamic scenarios.

With the introduction of deep reinforcement learning (DRL), robotics navigation has gained significant advantages over conventional path planning and rule-based methods. DRL is well-suited for real-world applications as it can handle continuous action spaces and does not require complete prior knowledge of the environment. This adaptability makes it particularly effective in complex and uncertain scenarios. Over recent years, DRL has been successfully applied to various robot navigation case studies, demonstrating strong performance in challenging environments [44]. It has proven effective in complete navigation systems [45, 46], path planning tasks [47, 48], and obstacle avoidance [10], ensuring safety-aware navigation that prevents collisions and unsafe paths. DRL has also been applied to address specific motion planning problems [49] and to navigate different terrain types [50]. Several examples highlight DRL's versatility. In a static environment, Geetha Sharma et al. [5] showed that autonomous unmanned aerial vehicles (UAVs) using the Soft Actor-Critic (SAC) algorithm could efficiently plan paths, avoid obstacles, and optimize trajectories. In dynamic environments with varying numbers of obstacles, TurtleBot 3 employed the Deep Deterministic Policy Gradient (DDPG) algorithm combined with Voronoi diagrams, demonstrating high effectiveness [51]. In highly dynamic scenarios, UAVs equipped with a layered Deep Q-Network (DQN) achieved faster convergence, improved time and speed performance, and better generalization capabilities [52].

The traditional robot navigation framework is the approach where the robot creates a map of its surroundings using LIDAR or camera inputs through simultaneous localization and mapping (SLAM) [53]. Based on the generated local and global maps, the path to the goal point is planned while avoiding obstacles in the environment. However, deep reinforcement learning (DRL) has emerged as a powerful alternative due to its ability to handle high-dimensional problems and process large amounts of input data through deep neural networks. As a result, DRL has been used to replace or integrate with traditional navigation frameworks, particularly in navigation tasks.

Figure 3 demonstrates the interaction process between an agent and the environment in a typical DRL-based navigation framework. In this setup, the agent assumes roles traditionally handled by separate modules such as robot localization, map building, and local path planning.

In summary, the evolution of path planning algorithms has progressed in proportion to the complexity of the environments in which robots operate. The more challenging and dynamic the environment, especially in real-time scenarios with moving objects and dense surroundings, the more advanced and adaptable the algorithms need to be. Such algorithms must be flexible enough to adapt to changes over time, learn from experience, generalize to unseen scenarios, and handle dynamics and uncertainty without requiring explicit mapping. This contrasts with conventional algorithms, which rely on predefined maps and world models and perform well only in known, static environments. Thanks to these capabilities, DRL has made significant contributions to the field. Krishna Teja et al. [16] presents a graph showing the

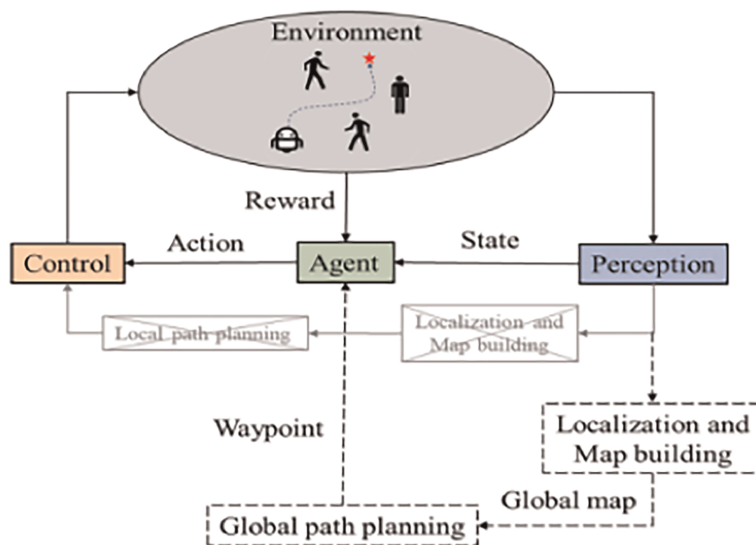


Figure 3.

Deep reinforcement learning-based navigation architecture [54]. *Reprinted from Ref. [54], IEEE, CC BY license.

number of cases and algorithms applied across different environmental domains. According to the reviewed literature, DRL is the most widely applied method for robot path planning in dynamic and hybrid environments, outperforming many conventional algorithms in terms of both effectiveness and efficiency [44].

2.2 Sample inefficiency

To learn an effective policy, deep reinforcement learning (DRL) requires extensive sampling and interaction with the environment. However, sample efficiency is often low due to the large number of samples needed. Several strategies can improve sample efficiency, such as using off-policy algorithms (e.g., DQN, DDPG, SAC) that reuse past experiences [55]. Compared to on-policy algorithms like PPO, SAC often performs better. Experience replay [56–58] further enhances learning by enabling more training per sample. Improving exploration can also prevent sample wastage; for example, SAC leverages entropy regularization to encourage more effective exploration [57].

In addition, imitation learning, model-based reinforcement learning, and improved reward design [3, 59] can further boost efficiency. Imitation learning and model-based DRL allow agents to avoid starting from scratch, reducing the number of interactions required. Proper reward design significantly affects the learning process, as sparse or poorly shaped rewards hinder efficiency. Metaheuristic algorithms can be integrated with DRL to optimize reward shaping or path parameters [59–61] fine-tuning reward weights or structures to make them more informative. This accelerates convergence, improves generalization, and reduces wasted samples from misleading rewards—benefits that are particularly valuable in real-world robotics and simulation-to-real transfer, where sample collection is costly.

To address instability in DRL for continuous control, Gwang Jong Ko et al. [62] proposed a two-phase optimization framework, where a swarm intelligence algorithm

is applied at the end of the learning phase. Experiments demonstrated that this approach achieved superior and more stable performance than conventional DRL methods in robot locomotion tasks. Finally, simulators such as GAZEBO [14, 63] and NVIDIA Isaac SIM [9] can significantly improve sample efficiency by allowing extensive pre-training in virtual environments before deployment in the real world. This enables agents to generalize better with fewer real-world samples.

2.3 Integration with perception and mapping

To improve the robustness and efficiency of deep reinforcement learning (DRL) in robotics, tighter integration with core robot subsystems such as perception and mapping is essential. For example, snake robots navigating in unknown dynamic environments can leverage multiple onboard sensors to reach a goal while avoiding obstacles [64]. A deep Q-Learning-based path planning method combined with simultaneous localization and mapping (SLAM) and visual perception from 2D LiDAR and IMU enables end-to-end navigation. Results show that active SLAM improves the success rate and reduces collisions compared with traditional A* and sampling-based RRT* algorithms.

Rui Wang et al. [65] demonstrated this with an aerial robot exploring an unknown environment. SLAM was employed to map the area, while an improved frontier-based exploration strategy integrated with DRL and a neural network architecture guided the robot toward target points. The proposed method consistently explored environments of varying sizes, requiring shorter travel distances than competing approaches.

Hybrid models also show significant benefits. In a simulated indoor maze with moving obstacles, a mobile robot using LiDAR and IMU inputs achieved higher speed and goal-reaching success rates than standard DRL. In this case, Proximal Policy Optimization (PPO) learned obstacle avoidance strategies, while a Genetic Algorithm optimized hyperparameters such as discount factors, reward weights, and network layer sizes before training. This integration not only improved sample efficiency and robustness but also enhanced dynamic obstacle avoidance.

2.4 Multi-agent robot research

A current research trend focuses on *multi-agent systems*, where the number of robots is increasing. Commonly referred to as swarm or cooperative navigation tasks as well as collaborative and competitive multi-robot systems [66–69]. When multiple agents work cooperatively, tasks can be distributed among them, improving overall performance [70]. To enable agents to learn effectively and adapt to various domains, DRL has been applied in multi-agent case studies to enhance scalability as the number of agents increases [45].

Multi-robot path finding (MRPF) [47] has been proposed to achieve efficient coordination and collaboration in high-dimensional multi-agent systems. The framework is built on DRL using a two-layer concept, and by integrating this with a hierarchical reward mechanism, the success rate improves by more than 20% compared to existing algorithms, leading to better generalization and efficiency.

An innovative DRL variant, known as the *DRL-MPCGNNs model*, integrates DRL with model predictive control (MPC) and graph neural networks (GNN) for optimized path planning and task allocation in multi-agent systems. Proposed by ZhiXian Li et al. [71], this approach significantly improves path planning efficiency, task allocation effectiveness, and inter-robot collaboration performance.

2.5 Multi-objective navigation

As robot navigation tasks grow increasingly complex, research focus has shifted from achieving single objectives to addressing multiple objectives through advanced variations of Deep Reinforcement Learning (DRL). For example, Chien-Lun Cheng et al. [53] proposed a crowd-aware, multi-objective DRL-based navigation system called multi-objective dual-selection reinforcement learning (MODSRL), designed to navigate efficiently in crowded environments. By employing an appropriately designed reward function, MODSRL simultaneously achieves multiple objectives, including safety, time efficiency, collision avoidance, and smooth trajectory generation, demonstrating strong system robustness.

In the domain of aerial robotics, Jiahao Wu et al. [71] introduced a multi-objective navigation reinforcement learning (MONRL) framework for drones to navigate and avoid obstacles in unknown environments. Leveraging DRL, the drone learns navigation policies that optimize path planning while mitigating wind disturbances using camera input. The method was evaluated in both a virtual environment and a real-world model of New York City, achieving effective and adaptive navigation performance.

2.6 Simulation-to-real transfer

Agents are typically trained in simulation to avoid the high cost, risk, and inefficiency associated with real-world training. However, generalization often fails when transferring policies from simulation to real-world environments due to the *simulation-to-reality gap*. To address this challenge, several approaches can be adopted, including domain adaptation, domain randomization, world models, and simulation fidelity improvement.

For *domain adaptation*, methods such as feature-level alignment [72], adversarial training, or fine-tuning can be applied to map sensor inputs such as camera or LiDAR data from simulated to real-world domains. *Domain randomization* improves robustness and generalization by exposing the robot to randomized simulation conditions during training, including variations in lighting, obstacle properties, and dynamics.

Using high-fidelity simulators such as *NVIDIA Omniverse*, *Gazebo*, *MuJoCo*, and *Webots* [73] which incorporate realistic noise, latency, and sensor inaccuracies helps bridge the gap by better matching real-world physics and sensor data. In practice, most researchers train all policies in simulation before transferring them to real robots, and this approach has proven effective in achieving generalization across quadruped, mobile, and humanoid robot platforms.

3. Challenges in deploying deep reinforcement learning for robotic navigation

3.1 Challenge working in continuous action spaces

The real environment, which matches continuous action spaces, poses several challenges for deep reinforcement learning, such as high computational complexity, instability of control, exploration difficulties, and low sampling efficiency [3]. A few algorithms have been designed to handle continuous actions without discretization, including Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization

(PPO), and Soft Actor-Critic (SAC). In these methods, the policy is modeled using a continuous distribution such as a Gaussian, from which the agent samples an action and then deterministically outputs that action.

3.1.1 Exploration difficulty and high computational complexity

In continuous action spaces, where the number of possible states is infinite, the agent must explore numerous combinations to identify the optimal strategy. This significantly prolongs training time, increases the risk of converging to local optima, and demands extra effort. As the dimensionality of the action space increases, the agent must interact more with the environment to learn an effective policy. This increases data complexity, leading to higher computational requirements, which may be impractical for real-world robotic tasks.

3.1.2 Slow and unstable learning rate

In continuous domains, the real-valued actions taken by each agent depend on its policy, which determines the action to be taken in each state. The policy directly affects the gradients and learning process. Poor parameter choices can cause exploding or vanishing gradients, making learning slow and unstable. This instability is especially problematic when the policy diverges or when updates have negligible effects. Since policy design often involves neural networks with specific input-output configurations, researchers have addressed these issues by carefully initializing weights, applying output normalization, selecting appropriate algorithms, and employing entropy regularization [74] to improve stability.

3.1.3 Low sampling efficiency

Sampling efficiency is reduced because agents require a large number of samples across the infinite action space, which also increases computational demands. To ensure sufficient learning, the agent must gather enough interactions with the environment. To boost sampling efficiency (learning more from each sample), researchers have applied techniques such as experience replay and meta-learning [59]. Experience replay stores and reuses past samples instead of discarding them, improving stability and convergence. Meta-learning, on the other hand, enables the agent to adapt to new tasks quickly with only a few samples, rather than exploring the entire space. In this approach, the agent starts closer to the optimal policy, and with just a few gradient steps, it can learn parameters that adapt rapidly to a new task.

3.2 Time intensive

Training robots in real-world environments is time-consuming due to the complexity of the process. Transfer learning offers an alternative by enabling models to be trained in simulations before fine-tuning in the real world, thereby accelerating learning and reducing data requirements. This approach also improves the adaptability and responsiveness of DRL models to dynamic and changing real-world scenarios. Consequently, many researchers focus on simulation-based training [10, 48, 75, 76] or a combination of simulation and experimental setups [45] to ensure robust generalization. Studies have shown that DRL models trained in simulations can perform effectively in real-world environments [10, 14], demonstrating that good

generalization is achievable. To bridge the simulation-to-reality gap, validation must be conducted in both domains to ensure effective domain adaptation.

3.3 Transfer from simulation to the real world

Testing in simulation differs significantly from real-world deployment due to variations in sensors, noise, dynamics, and environment. These discrepancies can cause learned policies to fail when applied to real robots, potentially leading to unsafe and unpredictable conditions. Additionally, latency issues in real-time hardware and controllers may impair DRL policy performance. Policies trained in one environment often struggle to generalize to new or slightly altered settings, necessitating transfer learning [77] and domain adaptation, both challenging for DRL.

To address these challenges, several techniques are commonly employed and should be continuously refined. These include training robust policies that are less sensitive to small environmental changes [78], using high-fidelity simulators [73], initiating training in simulation followed by fine-tuning with real-world data, and applying domain randomization during simulation to improve policy generalization across diverse robotic settings.

3.4 Exploration vs. exploitation: Complexities in reward design

In reinforcement learning, an agent must balance exploiting known strategies to maximize rewards while exploring new actions to discover better ones. Deep reinforcement learning (DRL) trains agents by maximizing cumulative rewards received from the environment, enabling them to learn desired behaviors. However, balancing exploration and exploitation is challenging, especially in high-dimensional or sparse-reward environments [58]. Poorly designed rewards can lead to suboptimal learning, unintended behaviors, or agents getting stuck in local optima due to inadequate exploration. For example, in navigation tasks, rewards may be higher when the agent is near the goal and penalized when far away [14].

In Soft Actor-Critic (SAC), the reward function directly influences exploitation and indirectly affects exploration [79]. SAC's trade-off is based on maximum entropy reinforcement learning, where the agent maximizes both expected returns and policy entropy (encouraging stochastic actions). Fine-tuning techniques such as adaptive and dynamic entropy adjustment have been developed to optimize SAC's performance. Improper entropy tuning can cause excessive randomness, undermining learning efficiency.

While entropy encourages exploration, it may also lead to suboptimal actions in some applications. A well-shaped reward function accelerates learning by providing clear signals for exploitation, driving the agent toward desired behavior. Without meaningful rewards, the agent relies heavily on entropy-driven exploration, slowing progress. In summary, designing appropriate reward functions is a complex but crucial aspect of effective DRL training.

3.5 Dynamic environments: Dense and disaster-focused scenarios

Dynamic environments, characterized by changing obstacles, lighting, or terrain pose significant challenges for deep reinforcement learning (DRL) [79]. The agent must continuously re-explore as the environment evolves, complicating exploration and increasing sample requirements. This leads to longer, more expensive training due

to the greater number of interactions needed to learn effective policies across varying conditions.

Adapting quickly to such changes is difficult; policies may diverge instead of converging, especially for off-policy algorithms like deep Q-networks (DQN), whose replay buffers can contain outdated experiences that no longer represent the current dynamics [80]. The increased variance in reward signals caused by environmental changes further complicates stable policy learning. Unlike static settings, the same state in a dynamic environment may yield different outcomes at different times, causing the agent to struggle in adapting to new conditions.

This unpredictability increases the risk of outdated or mislearned policies, potentially leading to unsafe robot behaviors such as collisions or damage. Dense environments such as crowded pedestrian areas [75, 76] and disaster scenarios [46, 80] amplify these challenges, requiring robust and adaptive DRL solutions to ensure safe and effective navigation.

3.6 Scalability

Scalability refers to a DRL system's ability to learn efficiently as the number of agents, tasks, or environmental complexities increase, while maintaining reasonable time and resource requirements—especially in high-dimensional action and complex state spaces. In swarm robotics, navigation and task allocation become increasingly challenging due to dynamic environments [81]. Niyazi Furkan Bar et al. [6] proposed a DRL framework based on distributed architecture combined with adaptive reward mechanisms, demonstrating improved navigation success rates as the swarm size grows exponentially.

Martinez et al. [82] evaluated DRL performance in highly dynamic, real-world-like environments and confirmed its robustness in crowded and previously unseen scenarios. In contrast, Qiang Gao et al. [83] introduced Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for multi-unmanned surface vehicle (USV) path planning, showing enhanced training effectiveness and scalability, yielding more efficient and practical path planning.

To further improve scalability in algorithms like Soft Actor-Critic, strategies such as minimizing communication overhead, employing complex policy optimization, decentralized learning, graph-based policies, and federated reinforcement learning have been proposed [84]. These approaches help DRL handle growing system complexity while preserving performance and learning efficiency.

4. Future directions of deep reinforcement learning in robotics research

4.1 Dynamic environment

The ultimate goal of autonomous robotic systems is to navigate unknown and dynamic real-world environments. Deep reinforcement learning (DRL) has been widely applied in this domain to achieve adaptive and intelligent control. However, significant gaps remain, particularly in disaster [80] and emergency scenarios [85], where unpredictable and continuous changes demand highly adaptive robot control systems. To prevent accidents and ensure practical efficiency, these systems must be further enhanced to cope with such challenging environments. Approaches like adversarial training and uncertainty-aware policy learning show promise for improving robustness in unpredictable settings.

In disaster environments, where human access is limited, robots are expected to navigate complex and hazardous terrain efficiently. Their role includes real-time monitoring, hazard identification, and survivor detection, supported by artificial intelligence and advanced onboard sensors. Such capabilities contribute to more resilient and responsive disaster management. DRL particularly deep Q-networks has demonstrated significantly higher rescue success rates across various natural disaster scenarios compared to traditional methods [80].

4.2 Time-varying obstacles environment

Highly dynamic environments are characterized by frequently changing obstacle configurations, where both the shape and position of obstacles vary continuously over time. For instance, pedestrian environments [75] exhibit constantly shifting movements of people [76] while logistics and intelligent transportation settings feature dynamic obstacles such as both pedestrians and vehicles. The high level of dynamism in these environments demands safe and efficient navigation strategies. The risk of failure increases significantly in densely populated dynamic spaces, where obstacle density is high. Therefore, intensive research efforts should focus on these complex scenarios rather than on simpler obstacle configurations when applying deep reinforcement learning (DRL) [75].

4.3 Increasing learning efficiency through sample and data optimization

Learning with deep reinforcement learning (DRL) demands substantial computational resources due to the vast amounts of data generated from interactions among environment, actions, and rewards. In high-dimensional state and action spaces, the exponential growth of possible states leads to increased data usage, longer training times, and higher computational costs. To mitigate this complexity, future strategies could include improved exploration methods, model-based reinforcement learning, integration of prior knowledge, and extensive simulation training.

Exploration remains a critical challenge, especially in complex environments where extensive exploration is necessary to discover effective strategies, directly impacting learning efficiency. For example, exploration in Soft Actor-Critic (SAC) can be enhanced by increasing target entropy, augmenting reward terms, and incorporating model-based approaches for sparse-reward tasks [57].

4.4 Advanced robotics application

The application of deep reinforcement learning (DRL) has demonstrated significant success in enhancing the adaptability and efficiency of robot navigation in challenging environments. Integrating DRL within robotic systems improves perception [3], instruction interpretation, and symbolic reasoning for high-level planning. For instance, Li BoXin and Wang Zhaokui [86] proposed a two-stage DRL control method combining Vision-TD3 and Force-TD3 networks, which leverage hybrid visual and force feedback for end-to-end lunar robot assembly operations, enabling autonomous, intelligent task execution without relying on model-based control.

To accelerate learning and align robot behavior with human intentions, incorporating human feedback such as reward shaping, demonstrations, and corrections has proven effective [87]. Chang Chiu Liu et al. [88] introduced a mixed perception-based human-robot collaborative maintenance system employing a hierarchical structure,

where online DRL aids decision-making. Experimental results demonstrated competitive performance compared to state-of-the-art approaches.

Hybridizing Soft Actor-Critic (SAC) with blockchain technology further enhances decentralized decision-making, security, and computational efficiency. By integrating SAC with distributed cloud frameworks, real-time deployment becomes feasible on autonomous ground vehicles, drones, and robotic arms, even on low-power processors [3].

As reviewed by Saba Waseem et al. [89], the future of advanced robotic manipulators depends on intelligent processing technologies and expert systems. Emphasis should be placed on robust controllers capable of adapting to unforeseen scenarios, improving human-robot collaboration, and ensuring operational safety and efficiency. Mohsen Soori et al. [90] highlighted applications of AI, machine learning, and DRL across advanced robotics industries, underscoring the need to continually enhance system robustness and performance.

Beyond navigation, DRL is increasingly applied to collaborative manipulation [91, 92], warehouse automation [93, 94], and swarm robot exploration [6], with future efforts focused on extending DRL to multi-robot cooperative and competitive scenarios to meet complex real-world demands.

4.5 Bridging the gap between simulation and real robot transfer

Bridging the reality gap between simulation and physical robots remains a critical challenge in robotics research. Future directions are likely to emphasize strategies such as learning invariant representations, domain randomization, domain adaptation, and the incorporation of real-world data into simulated training [45]. Given the high computational cost of DRL algorithms and their limited feasibility on embedded robotic platforms, developing efficient DRL methods that can run on constrained hardware is an important priority.

For example, the COBOT robot using Soft Actor-Critic (SAC) for obstacle avoidance has demonstrated efficiency in both simulation and real-world experiments [10]. To further bridge the simulation-to-reality gap, the official Unified Robot Description Format (URDF) is employed to enhance simulation fidelity. Physical parameters such as maximum joint speeds, accelerations, rotational limits, momentum, and center of mass are calibrated to match real robotic arm characteristics, significantly improving the accuracy of the simulation.

In another study, SAC was tested on the TurtleBot3 Waffle Pi, equipped with a Raspberry Pi, using the Gazebo emulator within the Robot Operating System (ROS) framework. The hardware included an NVIDIA GeForce GTX 1080 GPU and Intel Core i7 CPU (2.9 GHz). Wen et al. further advanced DRL-based path planning in unknown environments through Twin Long Short-Term Memory Delayed Deep (TLSMD) reinforcement learning, validated via numerous 3D simulations [14, 63].

5. Conclusion

Robot navigation remains a prominent research topic focused on achieving efficient and effective movement toward a goal while avoiding collisions with obstacles in the environment. Numerous algorithms have been proposed over time to generate optimal and safe paths based on their evolving capabilities. To address increasingly challenging scenarios such as dense, dynamic obstacles and unpredictable emergency conditions,

robot navigation algorithms must enable continuous learning in continuous action spaces, making complex decisions and optimizing policies. Deep reinforcement learning (DRL), a subset of artificial intelligence and machine learning, offers promising solutions to these demands. This chapter provides a comprehensive review of DRL in robot navigation research, beginning with fundamental concepts, followed by current technological trends. Key challenges in this field are discussed alongside insights and future directions drawn from recent studies. With the goal of advancing autonomous systems for robots and related applications, this review aims to help researchers understand DRL's capabilities and encourage its adoption in diverse domains. There remains significant potential to expand DRL applications into more complex environments while maintaining robustness, scalability, and efficiency. Furthermore, the learning efficiency of DRL algorithms is typically evaluated using metrics such as cumulative reward, sample efficiency, convergence time, success rate, learning stability, exploration efficiency, and policy robustness. These metrics should be incorporated to comprehensively assess real-time applications. To ensure the robustness and effectiveness of learning algorithms such as Soft Actor-Critic, they can be benchmarked against other DRL algorithms, including Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Proximal Policy Optimization (PPO).

Acknowledgements

The authors acknowledge the use of artificial intelligence-based language tools, specifically ChatGPT (OpenAI), to assist with language polishing, grammar checking, and clarity improvement during the preparation of this chapter. The use of these tools was limited to language refinement only. All technical content, analysis, interpretations, and conclusions were entirely developed by the authors, who take full responsibility for the accuracy and integrity of the work.

Author details

Nohaidda Sariff^{1*}, Yahya Muhammad Adam^{1†}, Intan Izafina Idrus^{1†},
Zool Hilmi Ismail^{2†}, Puteri Nor Aznie Fahsyar^{3†}, Swee King Phang^{1†}, Kok Seng Eu^{1†},
Md Hasan Molla^{1†} and Denesh Sooriamoorthy^{4†}

1 Centre for Intelligent Innovations (CII), School of Engineering, Taylor's University,
Subang Jaya, Selangor, Malaysia

2 Center for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Jalan
Sultan Yahya Petra, Kuala Lumpur, Malaysia

3 Faculty of Engineering, Higher College of Technology, Abu Dhabi, United Arab
Emirates

4 School of Engineering, Asia Pacific University of Technology and Innovation
(APU), Kuala Lumpur, Malaysia

*Address all correspondence to: nohaidda.sariff@taylors.edu.my

† These authors contributed equally.

IntechOpen

© 15 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Ekundayo OS, Ezugwu AE. Deep learning: Historical overview from inception to actualization, models, applications and future trends. *Applied Soft Computing*. 2025;**181**:113378. DOI: 10.1016/j.asoc.2025.113378
- [2] del Real Torres A, Andreiana DS, Roldán AO, Bustos AH, Galicia LEA. A review of deep reinforcement learning approaches for smart manufacturing in industry 4.0 and 5.0 framework. *Applied Sciences*. 2022;**12**(23):12
- [3] Zhang Y, Zhao W, Wang J, Yuan Y. Recent progress, challenges and future prospects of applied deep reinforcement learning: A practical perspective in path planning. *Neurocomputing*. 2024;**608**: 128423. DOI: 10.1016/j.neucom.2024.128423
- [4] Zhu Y, Wan Hasan WZ, Harun Ramli HR, Norsahperi NMH, Mohd Kassim MS, Yao Y. Deep reinforcement learning of mobile robot navigation in dynamic environment: A review. *Sensors*. 2025;**25**:3394. DOI: 10.3390/s25113394
- [5] Sharma G, Jain S. Deep reinforcement learning-based framework for path planning of UAVs. *Procedia Computer Science*. 2025;**258**:1112-1122. DOI: 10.1016/j.procs.2025.04.346
- [6] Bar NF, Karakose M. Collaborative approach for swarm robot systems based on distributed DRL. *Engineering Science and Technology, an International Journal*. 2024;**53**:101701. DOI: 10.1016/j.jestch.2024.101701
- [7] Tran DT et al. Central pattern generator based reflexive control of quadruped walking robots using a recurrent neural network. *Robotics and Autonomous Systems*. 2014;**62**(10): 1497-1516. DOI: 10.1016/j.robot.2014.05.011
- [8] Bellegarda G, Nguyen C, Nguyen Q. Robust quadruped jumping via deep reinforcement learning. *Robotics and Autonomous Systems*. 2024;**182**:104799. DOI: 10.1016/j.robot.2024.104799
- [9] Baltes J, Christmann G, Saeedvand S. A deep reinforcement learning algorithm to control a two-wheeled scooter with a humanoid robot. *Engineering Applications of Artificial Intelligence*. 2023;**126**:106941. DOI: 10.1016/j.engappai.2023.106941
- [10] Xia W, Lu Y, Xu W, Xu X. Deep reinforcement learning based proactive dynamic obstacle avoidance for safe human-robot collaboration. *Manufacturing Letters*. 2024;**41**:1246-1256. DOI: 10.1016/j.mfglet.2024.09.151
- [11] Gök M. Dynamic path planning via dueling double deep Q-network (D3QN) with prioritized experience replay. *Applied Soft Computing*. 2024;**158**: 111503. DOI: 10.1016/j.asoc.2024.111503
- [12] Deshpande SV, Harikrishnan R, Babul Salam KSMKI, Ponnuru MDS. Mobile robot path planning using deep deterministic policy gradient with differential gaming (DDPG-DG) exploration. *Cognitive Robotics*. 2024;**4**: 156-173. DOI: 10.1016/j.cogr.2024.08.002
- [13] Chen X, Yin S, Li Y, Xiang Z. Dynamic path planning for multi-USV in complex ocean environments with limited perception via proximal policy optimization. *Ocean Engineering*. 2025; **326**:120907. DOI: 10.1016/j.oceaneng.2025.120907
- [14] Wen S, Shu Y, Rad A, Wen Z, Guo Z, Gong S. A deep residual

reinforcement learning algorithm based on soft actor-critic for autonomous navigation. *Expert Systems with Applications*. 2025;**259**:125238. DOI: 10.1016/j.eswa.2024.125238

[15] Loganathan A, Ahmad NS. A systematic review on recent advances in autonomous mobile robot navigation. *Engineering Science and Technology, an International Journal*. 2023;**40**:101343. DOI: 10.1016/j.jestch.2023.101343

[16] Krishna Teja G, Mohanty PK, Das S. Review on path planning methods for mobile robot. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. 2025;**239**(14):5547-5580. DOI: 10.1177/09544062251330083

[17] Yeon ASA, Visvanathan R, Mamduh SM, Kamarudin K, Kamarudin LM, Zakaria A. Implementation of behaviour based robot with sense of smell and sight. *Procedia Computer Science*. 2015; **76**:119-125. DOI: 10.1016/j.procs.2015.12.300

[18] Jiang S, Arkin RC. SLAM-based spatial memory for behavior-based robots. *IFAC-PapersOnLine*. 2015;**48**(19):195-202. DOI: 10.1016/j.ifacol.2015.12.033

[19] Lazarowska A. Discrete artificial potential field approach to mobile robot path planning. *IFAC-PapersOnLine*. 2019;**52**(8):277-282. DOI: 10.1016/j.ifacol.2019.08.083

[20] Sariff N, Buniyamin N. An overview of autonomous mobile robot path planning algorithms. In: *SCORED 2006 - Proceedings of 2006 4th Student Conference on Research and Development "Towards Enhancing Research Excellence in the Region". USA: IEEE; 2006*. DOI: 10.1109/SCORED.2006.4339335

[21] Buniyamin N, Wan Ngah WAJ, Muhammad SNZ. A simple local path planning algorithm for autonomous mobile robots. *International Journal of Systems Applications, Engineering & Development (ISAED)*. 2011;**5**(2): 151-159

[22] Sariff N, Buniyamin N. Evaluation of robot path planning algorithms in global static environments: Genetic algorithm vs ant colony optimization algorithm. *International Journal of Electrical and Electronic Systems Research*. 2010;**3**(1): 1-12

[23] Duchoň F et al. Path planning with modified a star algorithm for a mobile robot. *Procedia Engineering*. 2014;**96**:59-69. DOI: 10.1016/j.proeng.2014.12.098

[24] Huang J, Chen C, Shen J, Liu G, Xu F. A self-adaptive neighborhood search A-star algorithm for mobile robots global path planning. *Computers and Electrical Engineering*. 2025;**123**:110018. DOI: 10.1016/j.compeleceng.2024.110018

[25] Ahmad J, Nadhir Ab Wahab M. Enhancing the safety and smoothness of path planning through an integration of Dijkstra's algorithm and piecewise cubic Bezier optimization. *Expert Systems with Applications*. 2025;**289**:128315. DOI: 10.1016/j.eswa.2025.128315

[26] Yang H et al. Research on multi-objective point path planning for mobile inspection robot based on multi-informed-rapidly exploring random tree*. *Engineering Applications of Artificial Intelligence*. 2025;**151**:110645. DOI: 10.1016/j.engappai.2025.110645

[27] Xu C, Zhu H, Zhu H, Wang J, Zhao Q. Improved RRT* algorithm for automatic charging robot obstacle avoidance path planning in complex environments. *CMES - Computer Modeling in Engineering and Sciences*.

2023;**137**(3):2567-2591. DOI: 10.32604/cmes.2023.029152

[28] Ge L, Phang SW, Sariff N. DPF-Bi-RRT*: An improved path planning algorithm for complex 3D environments with adaptive sampling and dual potential field strategy. *IEEE Access*. 2025;**13**:35958-35972

[29] Sariff NB, Buniyamin N. Genetic algorithm versus ant colony optimization algorithm - Comparison of performances in robot path planning application. In: *ICINCO 2010 - Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics*. Setúbal, Portugal: SciTePress; 2010

[30] Sariff NB, Buniyamin N. Comparative study of genetic algorithm and ant colony optimization algorithm performances for robot path planning in global static environments of different complexities. In: *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA*. USA: IEEE; 2009. DOI: 10.1109/CIRA.2009.5423220

[31] Buniyamin N, Sariff N, Wan NWAJ, Mohamad Z. Robot global path planning overview and a variation of ant colony system algorithm. *International Journal of Mathematics and Computers in Simulation*. 2011;**5**(1):9-15

[32] Sariff NB, Buniyamin N. Ant Colony system for robot path planning in global static environment. In: *International Conference on System Science and Simulation in Engineering – Proceedings*. USA: World Scientific and Engineering Academy and Society; 2010

[33] Cui Y, Hu W, Rahmani A. Multi-robot path planning using learning-based artificial bee Colony algorithm. *Engineering Applications of Artificial*

Intelligence. 2024;**129**:107579. DOI: 10.1016/j.engappai.2023.107579

[34] Mohammad SHA, Jeffril MA, Sariff N. Mobile robot obstacle avoidance by using fuzzy logic technique. In: *Proceedings - 2013 IEEE 3rd International Conference on System Engineering and Technology, ICSET 2013*. USA: IEEE; 2013. DOI: 10.1109/ICSEngT.2013.6650194

[35] Sariff N, Xing BTS. A wheeled mobile robot obstacles avoidance for navigation control in a static and dynamic environments. *Journal of Physics Conference Series*. 2023;**2523**(1): 012028. DOI: 10.1088/1742-6596/2523/1/012028

[36] Adam YM, Sariff NB, Algeelani NA. E-puck mobile robot obstacles avoidance controller using the fuzzy logic approach. In: *2021 2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. USA: IEEE; 2021. pp. 107-112. DOI: 10.1109/ICSCEE50312.2021.9497939

[37] Jeffril MA, Sariff N. The integration of fuzzy logic and artificial neural network methods for mobile robot obstacle avoidance in a static environment. In: *Proceedings - 2013 IEEE 3rd International Conference on System Engineering and Technology, ICSET 2013*. USA: IEEE; 2013. DOI: 10.1109/ICSEngT.2013.6650193

[38] Sariff NB, Wahab NHNBA. Automatic mobile robot obstacles avoidance in a static environment by using a hybrid technique based on fuzzy logic and artificial neural network. In: *Proceedings - 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, ICAIET 2014*. USA: IEEE; 2014. DOI: 10.1109/ICAIET.2014.31

- [39] Prados C, Hernando M, Gambao E. Path and footfall planning for N-legged and climbing robots — A model predictive control approach. *Robotics and Autonomous Systems*. 2025;**194**:105119. DOI: 10.1016/j.robot.2025.105119
- [40] Si G, Zhang R, Jin X. Path planning of factory handling robot integrating fuzzy logic-PID control technology. *Systems and Soft Computing*. 2025;**7**: 200188. DOI: 10.1016/j.sasc.2025.200188
- [41] Wang Y, Fu C, Huang R, Tong K, He Y, Xu L. Path planning for mobile robots in greenhouse orchards based on improved A* and fuzzy DWA algorithms. *Computers and Electronics in Agriculture*. 2024;**227**:109598. DOI: 10.1016/j.compag.2024.109598
- [42] Lin S, Liu A, Wang J, Kong X. An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse. *Journal of Computer Science*. 2023;**67**:101938. DOI: 10.1016/j.jocs.2022.101938
- [43] Hu L, Wei C, Yin L. Fuzzy A* quantum multi-stage Q-learning artificial potential field for path planning of mobile robots. *Engineering Applications of Artificial Intelligence*. 2025;**141**:109866. DOI: 10.1016/j.engappai.2024.109866
- [44] Li S, Li X, Wang P. Application of robot autonomous navigation based on reinforcement learning. *Procedia Computer Science*. 2025;**262**:1402-1409. DOI: 10.1016/j.procs.2025.05.188
- [45] Lv X, Xiong W. Robot autonomous navigation path planning based on deep reinforcement learning. *Procedia Computer Science*. 2025;**262**:1130-1136. DOI: 10.1016/j.procs.2025.05.151
- [46] Li C-Y, Zhang F, Chen L. Robot-assisted pedestrian evacuation in fire scenarios based on deep reinforcement learning. *Chinese Journal of Physics*. 2024;**92**:494-531. DOI: 10.1016/j.cjph.2024.09.008
- [47] Huo L, Mao J, San H, Li R, Zhang S. Deep reinforcement learning of group consciousness for multi-robot pathfinding. *Engineering Applications of Artificial Intelligence*. 2025;**155**:110978. DOI: 10.1016/j.engappai.2025.110978
- [48] Tao B, Kim J-H. Deep reinforcement learning-based local path planning in dynamic environments for mobile robot. *Journal of King Saud University - Computer and Information Sciences*. 2024;**36**(10):102254. DOI: 10.1016/j.jksuci.2024.102254
- [49] Li Y et al. Peduncle collision-free grasping based on deep reinforcement learning for tomato harvesting robot. *Computers and Electronics in Agriculture*. 2024;**216**:108488. DOI: 10.1016/j.compag.2023.108488
- [50] Huang S, Xiao Z, Zheng M, Shi W. Hierarchical reinforcement learning for enhancing stability and adaptability of hexapod robots in complex terrains. *Biomimetic Intelligence and Robotics*. 2025;**5**:100231. DOI: 10.1016/j.birob.2025.100231
- [51] Zhao H, Guo Y, Liu Y, Jin J. Multirobot unknown environment exploration and obstacle avoidance based on a Voronoi diagram and reinforcement learning. *Expert Systems with Applications*. 2025;**264**:125900. DOI: 10.1016/j.eswa.2024.125900
- [52] Guo T, Jiang N, Li B, Zhu X, Wang Y, Du W. UAV navigation in high dynamic environments: A deep reinforcement learning approach. *Chinese Journal of Aeronautics*. 2021;**34**(2):479-489. DOI: 10.1016/j.cja.2020.05.011

- [53] Cheng C-L, Hsu C-C, Saeedvand S, Jo J-H. Multi-objective crowd-aware robot navigation system using deep reinforcement learning. *Applied Soft Computing*. 2024;**151**:111154. DOI: 10.1016/j.asoc.2023.111154
- [54] Zhu K, Zhang T. Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Science and Technology*. 2021;**26**(5): 674-691. DOI: 10.26599/TST.2021.9010012
- [55] Yin Y, Chen Z, Liu G, Yin J, Guo J. Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning. *Expert Systems with Applications*. 2024;**247**:123202. DOI: 10.1016/j.eswa.2024.123202
- [56] Luo Y et al. Relay hindsight experience replay: Self-guided continual reinforcement learning for sequential object manipulation tasks with sparse rewards. *Neurocomputing*. 2023;**557**: 126620. DOI: 10.1016/j.neucom.2023.126620
- [57] Zhao T, Wang M, Zhao Q, Zheng X, Gao H. A path-planning method based on improved soft actor-critic algorithm for mobile robots. *Biomimetics*. 2023;**8**:481
- [58] Xiao W, Yuan L, Ran T, He L, Zhang J, Cui J. Multimodal fusion for autonomous navigation via deep reinforcement learning with sparse rewards and hindsight experience replay. *Displays*. 2023;**78**:102440. DOI: 10.1016/j.displa.2023.102440
- [59] Khan MR, Mohd Ibrahim A, Al Mahmud S, Samat FA, Jasni F, Mardzuki MI. Advancing mobile robot navigation with DRL and heuristic rewards: A comprehensive review. *Neurocomputing*. 2025;**652**:131036. DOI: 10.1016/j.neucom.2025.131036
- [60] Guo Q, Zhao W, Lyu Z, Zhao T. A GAN enhanced meta-deep reinforcement learning approach for DCN routing optimization. *Information Fusion*. 2025;**121**:103160. DOI: 10.1016/j.inffus.2025.103160
- [61] Kharitonov A, Abani JI, Nahhas A, Turowski K. Literature survey on combining machine learning and metaheuristics for decision-making. *Procedia Computer Science*. 2025;**253**: 199-208. DOI: 10.1016/j.procs.2025.01.083
- [62] Ko G-J, Huh J. Metaheuristic-based weight optimization for robust deep reinforcement learning in continuous control. *Swarm and Evolutionary Computation*. 2025;**95**:101920. DOI: 10.1016/j.swevo.2025.101920
- [63] Wen T, Wang X, Zheng Z, Sun Z. A DRL-based path planning method for wheeled mobile robots in unknown environments. *Computers and Electrical Engineering*. 2024;**118**:109425. DOI: 10.1016/j.compeleceng.2024.109425
- [64] Liu X, Wen S, Hu Y, Han F, Zhang H, Karimi HR. An active SLAM with multi-sensor fusion for snake robots based on deep reinforcement learning. *Mechatronics*. 2024;**103**:103248. DOI: 10.1016/j.mechatronics.2024.103248
- [65] Wang R, Zhang J, Lyu M, Yan C, Chen Y. An improved frontier-based robot exploration strategy combined with deep reinforcement learning. *Robotics and Autonomous Systems*. 2024;**181**:104783. DOI: 10.1016/j.robot.2024.104783
- [66] Blais M-A, Akhloufi MA. Reinforcement learning for swarm robotics: An overview of applications, algorithms and simulators. *Cognitive Robotics*. 2023;**3**:226-256. DOI: 10.1016/j.cogr.2023.07.004

- [67] Ali ZA, Israr A, Hasan R. Survey of methods applied in cooperative motion planning of multiple robots. In: Ali ZA, Israr A, editors. *Motion Planning for Dynamic Agents*. Rijeka: IntechOpen; 2023. DOI: 10.5772/intechopen.1002428
- [68] Zhang J, Jia Q, Zhang S, Chen G. Dynamic and prioritized task scheduling of heterogeneous multi-robot systems using deep reinforcement learning. *Neurocomputing*. 2025;**638**:130184. DOI: 10.1016/j.neucom.2025.130184
- [69] Sariff NB, Ismail ZH. A survey and analysis of cooperative multi-agent robot systems: Challenges and directions. In: Gorrostieta Hurtado E, editor. *Applications of Mobile Robots*. Rijeka: IntechOpen; 2018. DOI: 10.5772/intechopen.79337
- [70] Sariff NB, Ismail ZH, Sooriamoorthy D, Syed Mahadzir PNAF, Md Yasir ASH. Multi-agent robot motion planning for rendezvous applications in a mixed environment with a broadcast event-triggered consensus controller. In: Ali ZA, Israr A, editors. *Motion Planning for Dynamic Agents*. Rijeka: IntechOpen; 2023. DOI: 10.5772/intechopen.1002494
- [71] Wu J, Ye Y, Du J. Multi-objective reinforcement learning for autonomous drone navigation in urban areas with wind zones. *Automation in Construction*. 2024;**158**:105253. DOI: 10.1016/j.autcon.2023.105253
- [72] Jang Y, Baek J, Jeon S, Han S. Bridging the simulation-to-real gap of depth images for deep reinforcement learning. *Expert Systems with Applications*. 2024;**253**:124310. DOI: 10.1016/j.eswa.2024.124310
- [73] Farley A, Wang J, Marshall JA. How to pick a mobile robot simulator: A quantitative comparison of CoppeliaSim, Gazebo, MORSE and Webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory*. 2022;**120**:102629. DOI: 10.1016/j.simpat.2022.102629
- [74] Hu Y, Fu J, Wen G, Lv Y, Ren W. Distributed entropy-regularized multi-agent reinforcement learning with policy consensus. *Automatica*. 2024;**164**:111652. DOI: 10.1016/j.automatica.2024.111652
- [75] Kim J, Hwang H-S, Seok J. Adversarial environment design for crowd navigation based on deep reinforcement learning. *Engineering Applications of Artificial Intelligence*. 2025;**159**:111621. DOI: 10.1016/j.engappai.2025.111621
- [76] Mishra N, Yamaguchi T, Okuda H, Suzuki T. Combination of reinforcement learning models towards considerate motion planning for multiple pedestrians. *IFAC-PapersOnLine*. 2023;**56**(2):3616-3621. DOI: 10.1016/j.ifacol.2023.10.1523
- [77] Wen S, Wen Z, Zhang D, Zhang H, Wang T. A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning. *Applied Soft Computing*. 2021;**110**:107605. DOI: 10.1016/j.asoc.2021.107605
- [78] Joshi Kumar V, Elumalai VK. A proximal policy optimization based deep reinforcement learning framework for tracking control of a flexible robotic manipulator. *Results in Engineering*. 2025;**25**:104178. DOI: 10.1016/j.rineng.2025.104178
- [79] Yang L, Bi J, Yuan H. Dynamic path planning for mobile robots with deep reinforcement learning. *IFAC-PapersOnLine*. 2022;**55**(11):19-24. DOI: 10.1016/j.ifacol.2022.08.042

- [80] Lei Y, Liu J, Ke Y. Multi-disaster emergency response decision support based on reinforcement learning algorithm. *Procedia Computer Science*. 2025;**261**:887-895. DOI: 10.1016/j.procs.2025.04.418
- [81] Khaldi B, Harrou F, Sun Y. Collaborative swarm robotics for sustainable environment monitoring and exploration: Emerging trends and research progress. *Energy Nexus*. 2025; **17**:100365. DOI: 10.1016/j.nexus.2025.100365
- [82] Martinez-Baselga D, Riazuelo L, Montano L. RUMOR: Reinforcement learning for understanding a model of the real world for navigation in dynamic environments. *Robotics and Autonomous Systems*. 2025;**191**:105020. DOI: 10.1016/j.robot.2025.105020
- [83] Gao Q, Li S, Ji Y, Liu J, Song Y. Scalable path planning algorithm for multi-unmanned surface vehicles based on multi-agent deep deterministic policy gradient. *Ocean Engineering*. 2025;**320**: 120243. DOI: 10.1016/j.oceaneng.2024.120243
- [84] Hersi AH, Divya Udayan J. Efficient and robust multirobot navigation and task allocation using soft actor critic. *Procedia Computer Science*. 2024;**235**: 484-495. DOI: 10.1016/j.procs.2024. 04.048
- [85] Dong D, Wang Z, Guan J, Xiao Y, Wang Y. Research on key technology and application progress of rescue robot in nuclear accident emergency situation. *Nuclear Engineering and Technology*. 2025;**57**(6):103457. DOI: 10.1016/j.net.2025.103457
- [86] Li B, Wang Z. Two-stage DRL with hybrid perception of vision and force feedback for lunar construction robotic assembly control. *Acta Astronautica*. 2025;**229**:357-373. DOI: 10.1016/j.actaastro.2025.01.017
- [87] Shah R, Doss ASA, Lakshmaiya N. Advancements in AI-enhanced collaborative robotics: Towards safer, smarter, and human-centric industrial automation. *Results in Engineering*. 2025;**27**:105704. DOI: 10.1016/j.rineng.2025.105704
- [88] Liu C, Zhang Z, Tang D, Nie Q, Zhang L, Song J. A mixed perception-based human-robot collaborative maintenance approach driven by augmented reality and online deep reinforcement learning. *Robotics and Computer-Integrated Manufacturing*. 2023;**83**:102568. DOI: 10.1016/j.rcim.2023.102568
- [89] Waseem S, Adnan M, Iqbal MS, Amin AA, Shah A, Tariq M. From classical to intelligent control: Evolving trends in robotic manipulator technology. *Computers and Electrical Engineering*. 2025;**127**:110559. DOI: 10.1016/j.compeleceng.2025.110559
- [90] Soori M, Arezoo B, Dastres R. Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*. 2023;**3**:54-70. DOI: 10.1016/j.cogr.2023. 04.001
- [91] Maldonado-Ramirez A, Rios-Cabrera R, Lopez-Juarez I. A visual path-following learning approach for industrial robots using DRL. *Robotics and Computer-Integrated Manufacturing*. 2021;**71**:102130. DOI: 10.1016/j.rcim.2021.102130
- [92] Zheng P, Li S, Fan J, Li C, Wang L. A collaborative intelligence-based approach for handling human-robot collaboration uncertainties. *CIRP Annals*. 2023;**72**(1):1-4. DOI: 10.1016/j.cirp.2023.04.057

[93] Konishi M, Sasaki T, Cai K. Efficient safe control via deep reinforcement learning and supervisory control – Case study on multi-robot warehouse automation. IFAC-PapersOnLine. 2022; 55(28):16-21. DOI: 10.1016/j.ifacol.2022.10.318

[94] Hosseini M, Chalil Madathil S, Khasawneh MT. Reinforcement learning-based simulation optimization for an integrated manufacturing-warehouse system: A two-stage approach. Expert Systems with Applications. 2025;290:128259. DOI: 10.1016/j.eswa.2025.128259

Toward Agentic Artificial Intelligence and Its Societal Implications

*Zool Hilmi Ismail, Nohaidda Sariff
and Puteri Nor Aznie Fahsyar*

Abstract

This chapter presents a conceptual examination of the transition from current agentic artificial intelligence or agentic AI system to future AI frameworks. Special attention is given to applications that intersect directly with society, such as next generation of educational tools and the resulting impacts on workforce dynamics, human augmentation, and ethical considerations. Central to this analysis is the importance of human-centered design, ensuring that agentic AI systems remain transparent, trustworthy, and aligned with human values. Identifying key research priorities, this study seeks to guide the development of agentic AI that benefits society while mitigating risks associated with autonomy and decision-making at scale.

Keywords: autonomous agents, human-centered design, multi-agent systems, agentic AI, artificial intelligence

1. Introduction

Artificial intelligence has evolved from rule-based expert systems and predictive algorithms to more complex machine learning models that generate content or assist decision-making. The latest frontier is *agentic AI*—AI systems that exhibit agency, autonomously pursuing goals and actions in dynamic environments. Unlike traditional AI that only responds to explicit inputs or generative AI that produces content when prompted, agentic AI can proactively act on its own initiative [1–4]. This shift toward autonomy and goal-directed behavior carries transformative potential alongside significant societal implications. Researchers and policymakers are closely examining how such systems can be designed, governed, and integrated responsibly into society [2].

This chapter provides a comprehensive overview of agentic AI and its broader impacts. We first define the concept of agentic AI and delineate how it differs from earlier AI paradigms. Next, we detail the technological foundations that give rise to agency in AI systems, including architectures and frameworks that enable planning,

learning, and autonomous execution. With this technical background established, we delve into societal dimensions: ethical challenges, legal questions, educational impacts, and international policy considerations. Finally, we outline key open research challenges and propose policy recommendations to ensure agentic AI develops in a direction beneficial to humanity. The discussion is grounded in the current literature and recent developments, providing an academic but accessible synthesis suitable for researchers and informed practitioners.

2. Defining agentic AI

Agentic AI refers to a class of AI systems endowed with *agency*: they can independently perceive their environment, reason about it, make decisions, and take actions to achieve objectives with limited or no ongoing human intervention [3]. In contrast to traditional AI (e.g., rule-based systems or narrowly focused predictive models) that operates only within predefined tasks or on external commands, agentic AI is goal-directed and autonomous. It does not merely react; it initiates behavior based on internal goals and contextual understanding [3]. Early examples of agentic AI include autonomous vehicles that navigate complex environments and virtual assistants or copilots that carry out multi-step tasks proactively [4].

Traditional AI systems (often deterministic rule-based programs or supervised models) require explicit human instructions for each operation and do not adapt goals on their own. Generative AI (such as large language models like GPT-4) can create content in response to prompts, displaying creativity within a reactive paradigm. Agentic AI goes a step further by combining content generation with decision-making and autonomous execution [5–7]. **Table 1** provides a comparison of traditional AI, generative AI, and agentic AI along several key dimensions.

Aspect	Traditional AI	Generative AI	Agentic AI
Primary capability	Deterministic reasoning or classification based on explicit rules/models	Generating new content (text, images, etc.) from learned patterns	Goal-driven decision-making and strategic action execution
Decision model	Fixed logic or supervised predictions (no self-directed goals)	Probabilistic model responding to user prompts (reactive)	Autonomous reasoning with internal goals and plans (proactive)
Adaptability	Low—performs only specified tasks; limited adaptation	Moderate – can generalize within training data patterns; adapts outputs to prompts	High—dynamically adapts to changing context and feedback in pursuit of objectives
Human involvement	High—constant human input or oversight required (e.g., manually triggering actions)	Moderate—requires prompt to initiate; then outputs content autonomously	Low—minimal supervision; can initiate and perform tasks by itself once goals are set
Examples	Expert systems; classification models (e.g., medical diagnosis aid)	ChatGPT-like text generators; image generators (given prompts)	Self-driving cars; automated trading agents; autonomous robots or software agents managing workflows

Table 1.
Comparison of traditional AI, generative AI, and agentic AI across various dimensions.

It is interesting to note that there are four properties often define an agentic AI system, distinguishing it from earlier AI paradigms:

- *Autonomy*: The system can make decisions and execute actions independently, without needing continuous external instructions [7]. For instance, an autonomous drone can plan its route and avoid obstacles without human micromanagement.
- *Intentionality*: The AI exhibits goal-directed behavior, formulating and pursuing objectives that drive its actions. It does not just follow single-step commands but can maintain and strive toward a persistent goal (e.g. “reduce energy usage in this building by 10%”).
- *Adaptability*: Agentic AI can adjust its strategies in response to environmental changes or new information. It learns and improvises—for example, a supply chain agent might automatically reroute shipments when a disruption occurs.
- *Reflectiveness*: The system can evaluate the outcomes of its actions, learn from experience, and improve over time. This often involves an internal feedback loop or memory that enables the agent to refine its behavior (analogous to a human reflecting on past decisions).

These properties allow an agentic AI to operate with a degree of self-direction and continuous learning that classical AI systems lack. In summary, agentic AI systems are proactive, goal-driven, and capable of both low-level task execution and high-level strategic reasoning, whereas traditional AI systems are reactive tools confined to predefined tasks [7].

3. Architectures of agentic AI

Creating an AI with agency requires integrating multiple AI techniques and system components. Agentic AI architectures are designed to allow an AI system to observe, think, act, and learn in a continuous loop [8, 9].

3.1 Core architecture

At a high level, an AI agent’s operation can be viewed as a cycle of sensing the environment, reasoning to decide on actions, executing those actions, and learning from the outcomes [10–14]. Classical agent models often followed a simple Perceive → Decide → Act → Learn sequence. Modern agentic AI expands this into a more modular pipeline—for example: *Trigger (input) → Plan → Tool use (action) → Memory update → Output/feedback*, with the ability to re-plan as needed [14]. In practice, this means an agent can incorporate new information or changing conditions on the fly, rather than strictly executing a fixed script.

3.1.1 Perception module

Interfaces with the external environment to gather data. This could involve computer vision and sensors for physical environments, or API inputs and natural

language processing for digital environments. Advanced agents use multimodal perception to interpret text, audio, images, or other sensor data, building a real-time picture of their context.

3.1.2 Reasoning and decision-making module

Often powered by AI models (e.g., a large language model or other machine learning models), this component interprets the perceived data, infers context, and selects appropriate actions. State-of-the-art large language models (LLMs) have significantly enhanced agents' reasoning abilities, enabling complex logical inferences and contextual understanding. The reasoning module might employ techniques like symbolic logic or neural networks to weigh options and make decisions aligned with the agent's goals.

3.1.3 Planning module

Formulates multi-step plans or strategies to achieve the agent's objectives. It may decompose high-level goals into subtasks, sequence actions, and adapt plans as conditions change. Techniques include classical AI planning algorithms and heuristic search, as well as hierarchical reinforcement learning (which plans at multiple levels of abstraction). A planning component gives the agent foresight, rather than reacting myopically, it anticipates future states and prepares accordingly.

3.1.4 Execution module

This is to allow the agent to carry out actions in the environment. In software agents, this might mean calling external tools, APIs, or services; in robotics, it would interface with motors or actuators. Modern agent frameworks enable the dynamic tool use, where the AI agent can invoke external functions (e.g., database queries, web services) as part of its actions. For example, an AI agent might autonomously send emails, execute trades, or control a robot arm. Ensuring reliable execution (and handling errors or exceptions) is crucial for real-world deployment of agentic AI.

3.1.5 Memory or knowledge base

Maintains state and stores knowledge, enabling the agent to remember context and learn from past experience. This includes short-term memory (working memory of the current context or conversation) and long-term memory (persistent knowledge the agent has learned or been provided). Technically, memory can be implemented *via* vector databases storing embeddings of past interactions (for semantic recall), knowledge graphs, or other databases of facts. Memory enables consistency over time and reflection—the agent can refer back to previous events when deciding what to do next.

3.1.6 Learning and adaptation module

To be truly agentic, an AI should improve over time. Many agents utilize reinforcement learning (RL) as a core mechanism: they receive feedback or rewards from the environment and adjust their behavior to maximize long-term success. Through trial-and-error, the agent learns which actions lead to desired outcomes.

Other learning approaches include supervised fine-tuning (updating the agent's models based on human feedback or new training data) and unsupervised updates to its knowledge base. Reflective learning allows the agent to analyze its own past decisions and outcomes to avoid repeating mistakes. This continuous learning capability is vital in non-stationary environments where static behavior would quickly become suboptimal.

3.2 Enabling technologies in agentic AI systems

Agentic AI systems are built upon advanced technologies that provide key capabilities such as autonomy, adaptability, and inter-agent collaboration. This section highlights the primary technologies that serve as the backbone of these intelligent systems [11–13].

3.2.1 Natural language processing

Natural language processing (NLP) and speech-to-text technologies enable machines to comprehend human language, facilitating communication, reasoning, and summarization. Tools like Large Language Models (LLMs), Claude, and Whisper are integrated into agentic AI to convert high-level goals into actionable sub-tasks. For instance, an LLM-based agent can allocate duties to warehouse robots or drones in logistics operations [14].

3.2.2 Reinforcement learning

The convergence of advanced machine learning (LLMs, reinforcement learning) with traditional AI planning and knowledge representation is what empowers agentic AI. By using models as core reasoning engines but surrounding them with memory, planning, and action interfaces, AI agents can function continuously and adaptively in real-world workflows [14]. In essence, an AI agent “wraps” a predictive model with an agentic architecture that decides *what* to do, *how* to do it, and *when* to do it based on the situation [11, 14, 15]. This technical foundation sets the stage for the profound societal changes and challenges that agentic AI engenders, which we explore in subsequent sections.

3.2.3 Multi-agent systems

To coordinate the multiple agents, technologies from multi-agent systems (MAS) and swarm intelligence are employed. These approaches enable agents to either work together or compete to enhance overall system performance [16]. Such cooperation is essential in domains like swarm robotics or decentralized AI environments. Tools like the Java Agent Development Framework (JADF) support MAS, while swarm-based methods like Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are applied in tasks such as dynamic routing, load distribution, and emergency response.

3.2.4 Knowledge graphs

Other pivotal technologies in agentic AI include knowledge graphs and ontologies, which organize domain knowledge to support semantic queries, planning, and logical

inference. Additionally, computer vision combined with sensor fusion provides environmental awareness. Techniques such as convolutional neural networks (CNNs), LiDAR data processing, and SLAM (simultaneous localization and mapping) play a critical role here.

For high-level reasoning and time-based coordination, autonomous planning and scheduling tools are employed. Frameworks like STRIPS and Hierarchical Task Networks (HTN) help agents determine and execute sequences of tasks to meet long-term objectives. Lastly, human-AI collaboration is enhanced through technologies like explainable AI (XAI) and augmented reality (AR), which foster transparency and shared decision-making between humans and machines.

4. Implementation of agentic AI in education

One domain poised for significant transformation by agentic AI is education. AI-powered agents are being developed to act as intelligent tutors, personalized learning companions, teaching assistants, and even as autonomous learners that adapt to students' needs [15]. In this section, we explore how agentic AI can be integrated into educational settings, the benefits and challenges of AI tutors, the concept of AI as a learning participant, and the potential impacts on pedagogy and cognitive development of students [17].

4.1 AI tutors and personalized learning

Educational AI agents promise highly personalized instruction at scale. By leveraging large language models and other AI techniques, these agents can understand a student's query, context, and learning history, which then provide guidance tailored to that student [18]. For instance, an agentic AI tutor can dynamically adjust lesson plans and explanations in response to a learner's progress—offering simpler explanations if the student is struggling or more advanced material if they are excelling. Such an AI tutor can be available 24/7 to answer questions, provide practice problems, or explain difficult concepts, effectively serving as a one-on-one tutor for every student. Research indicates that AI tutors can indeed provide adaptive support: They continuously assess a student's knowledge state and modify their teaching strategy according to Refs. [12, 18]. This addresses the long-standing challenge of personalization in education—human teachers, with the best intentions, cannot finely tailor their approach to each of dozens of students in a classroom at all times, but an AI agent can individualize interactions for each learner in real-time.

Agentic AI tutors can also enhance engagement and motivation by making learning more interactive. Through natural language dialog, an AI tutor can carry on a Socratic conversation, ask the student prompting questions, and give immediate feedback on answers. The interactivity and instant responsiveness help keep students engaged. Data analytics in the agent can identify knowledge gaps or misconceptions, alerting the teacher or recommending targeted practice for the student [12]. Such continuous formative feedback loops help students remain motivated, since they receive timely support and recognition of their progress [12, 19]. A recent systematic review of AI in education found that LLM-based agents can transform learner experiences by providing context-aware dialog and personalized learning pathways [12–20].

4.2 AI as learners and adaptation

Interestingly, agentic AI in education can be seen not only as a tutor but also as a learner. One aspect is that the AI agent continuously learns about the student, their strengths, weaknesses, and learning style, and adapts its teaching approach over time. Another aspect is using AI agents to simulate student learning for research purposes. For example, an AI could be put in the role of a student to see how it navigates learning tasks, potentially offering insights into different learning strategies or allowing testing of educational interventions in a controlled way [15]. There is also the concept of meta-learning, where AI tutors improve their instructional strategies by learning from many student interactions (effectively, the AI agent “learns how to teach” better with experience).

From the student’s perspective, interacting with an AI tutor can help develop certain meta-cognitive skills. Some evidence suggests that when students explain their reasoning to a tutor (even if the tutor is an AI), it strengthens understanding, a well-known educational phenomenon (the “learning-by teaching” effect). Agentic AIs can prompt students to articulate their thought process, ask follow-up questions that drive reflection, or encourage the student to attempt solutions before receiving help, all of which are strategies aligned with good pedagogy [15].

4.3 Cognitive development and critical thinking

A crucial question is how long-term exposure to AI tutors might affect the students’ cognitive development. On the positive side, a well-designed AI tutor can foster critical thinking. It can push students to justify their answers, consider alternative solutions, and explore topics more deeply by personalizing challenges. A report by Pearson in 2024 found that students who used AI features in their digital textbooks showed improved critical thinking skills, with active engagement in learning quadrupling compared to baseline levels [19]. The AI tools in that study likely provided interactive prompts and on-the-spot support that encouraged students to think through problems rather than passively read, thereby enhancing engagement. AI tutors can also help fill gaps where educational resources are scarce—for example, providing tutoring in under-served communities or in subjects where qualified teachers are few. This democratization of tutoring could help more learners reach their potential, arguably a net positive outcome for social equity.

However, there are potential downsides to mitigate. If students become overly reliant on AI guidance, they might struggle to develop independent problem-solving skills. There is a risk of AI tutors inadvertently doing the thinking for the student—if not carefully designed, a student might lean on an AI assistant to get answers with minimal effort, undermining the learning process (essentially the AI becoming a sophisticated form of cheating). This raises issues of academic integrity [17]. Educational institutions are already grappling with students using AI to generate essays or answers. Clear policies and possibly technical solutions (like AI that can detect AI-generated content) are part of the response to ensure AI assistance supplements rather than replacing a student’s own work [20].

4.4 Pedagogical integration

For agentic AI to be successfully integrated, educators must be trained to work alongside these tools. Curricula may need redesigning to leverage AI capabilities—for

instance, allowing more self-paced exploration with AI tutors available, while class time is used for discussion and collaborative projects (a flipped-classroom model). There is also the matter of digital literacy: Students must learn how to effectively use AI tools and critically evaluate AI-provided information (to avoid treating the AI as an infallible oracle). Teaching students how to question AI, cross-verify answers, and use the AI as a learning support rather than an answer vending machine is important.

Privacy is another consideration—AI tutors will collect data on student performance, mistakes, etc. Ensuring compliance with student data privacy laws (like FERPA in the US, or GDPR in Europe) is essential. Data should be used ethically, anonymized where possible, and kept secure. Similarly, fairness must be monitored: AI should not propagate biases (e.g., giving different feedback quality to students of different backgrounds). Designing AI tutors with inclusive datasets and testing them for unbiased behavior is part of responsible deployment [12, 18].

Finally, it is worth noting that agentic AI is itself becoming a subject of study for students. Schools and universities are beginning to include AI literacy in their curricula, teaching students about AI's workings and its societal implications—much like this chapter. This is crucial so that the next generation can navigate a world filled with AI agents knowledgeably and ethically.

5. Way forward in agentic AI

5.1 Robust alignment and control

Developing reliable methods to align autonomous AI agents with human values remains an open challenge. Current alignment techniques (like reward modeling or fine-tuning with human feedback) are not guaranteed to cover all scenarios an agent might encounter [21]. Research is needed on value alignment frameworks that can handle the flexibility of agentic AI, including techniques for real-time constraint enforcement and fail-safes if an agent's behavior deviates. Scalable oversight—possibly using AI to help monitor AI—is another area to explore, ensuring that even complex multi-agent systems remain under meaningful human control or at least observation.

5.2 Explainability

As agents use more complex models (deep neural networks with potential emergent behaviors), creating human-interpretable explanations of their decisions is challenging. Research into explainable AI (XAI) needs to extend to sequential decision-making and planning contexts. For instance, tools that can summarize an agent's long-term plan in intuitive terms, or highlight which factors most influenced a critical decision, would greatly aid both developers and regulators. Equally, audit trails for AI agents (automatic logging of decisions and actions) must be designed from the start—research can help determine what logging is sufficient to reconstruct and understand an agent's behavior after the fact [6].

5.3 Safety in adversarial environments

Many agentic AIs will operate in environments that can be adversarial (e.g., cybersecurity, finance, or competitive scenarios where other agents or humans might

deliberately manipulate inputs). Ensuring that AI agents are robust against adversarial attacks—for example, tricking the perception module or feeding malicious data into its learning process—is an ongoing challenge. Similarly, emergent behaviors in multi-agent systems (where agents interacting in unforeseen ways produce harmful dynamics like collusion or conflict) require more theoretical and empirical research. Techniques from game theory and control theory may be employed to predict and mitigate such outcomes.

5.4 Education and workforce transitions

Research into educational strategies for a world with ubiquitous AI tutors and AI assistants in the workplace is needed. Pedagogically, we should study long-term outcomes of students who learn with AI assistance: Do they perform better in higher education and jobs? Do they have any skills deficits? This ties into workforce development—identifying what new skills humans should focus on when AI agents handle more routine cognitive tasks. For example, there may be increased emphasis on creativity, critical thinking, and interpersonal skills—areas where humans complement AI. Policies on retraining and continuous education can benefit from such research.

5.5 Scalability

As agentic AI becomes more complex, the computational cost and energy footprint could be significant (e.g., running multiple large models continuously). Research into more efficient algorithms, sustainable hardware, and methods like federated or edge AI (to reduce latency and dependency on central servers) will be vital to scale these agents widely. This has ethical dimensions too—ensuring that deploying helpful AI agents does not come with an untenable environmental cost, and that they are accessible beyond just wealthy institutions with massive compute resources.

6. Conclusions

Agentic AI offers immense promise—from automating complex tasks to enabling new educational and scientific breakthroughs—but it must be guided by human-centric policies and rigorous research. By proactively addressing the challenges of alignment, safety, accountability, and equity, we can harness agentic AI as a positive force in society. The recommendations above provide a roadmap for a collaborative effort among AI scientists, ethicists, lawmakers, educators, and the public to ensure these powerful systems are developed responsibly and governed effectively. With thoughtful oversight and innovation hand in hand, agentic AI can be steered toward augmenting human capabilities and addressing societal needs, rather than amplifying risks. We stand at a pivotal moment: The decisions made now in research priorities and policy frameworks will shape how this technology integrates into the fabric of our lives and institutions in the decades to come.

AI tools disclosure

Artificial intelligence (AI) tools were used as assistive technologies during the preparation of this chapter. Specifically, AI-based language support was used to

improve grammar, clarity, and readability, and to help refine phrasing and consistency of academic tone. AI tools were also used to support formatting and restructuring of text. All technical content, interpretations, and conclusions were defined and verified by the authors, and the authors take full responsibility for the final manuscript.

Conflict of interest

The authors declare no conflict of interest.

Author details

Zool Hilmi Ismail^{1*†}, Nohaidda Sariff^{2†} and Puteri Nor Aznie Fahsyar^{3†}

1 Center for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Jalan Sultan Yahya Petra, Kuala Lumpur, Malaysia

2 School of Engineering, Taylor's University, Subang Jaya, Selangor, Malaysia

3 Faculty of Engineering, Higher College of Technology, Abu Dhabi, United Arab Emirates

*Address all correspondence to: zool@utm.my

† These authors contributed equally.

IntechOpen

© 2026 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Andrada G, Clowes R, Smart P. Varieties of transparency: Exploring agency within AI systems. *AI & Society*. 2023;**38**:1321-1331
- [2] Bosch C, Kruger D. AI chatbots as open educational resources: Enhancing student agency and self-directed learning. *Italian Journal of Educational Technology*. 2024;**32**:53-68
- [3] Chaudhari A, Charate P. Autonomous AI agents for real-time financial transaction monitoring and anomaly resolution using multi-agent reinforcement learning and explainable causal inferences. *International Journal of Advance Research, Ideas and Innovations in Technology (IJARIIT)*. 2025;**11**:142-150
- [4] Deng Z, Guo Y, Han C, Ma W, Xiong J, Wen S, et al. AI agents under threat: A survey of key security challenges and future pathways. *arXiv preprint arXiv:2406.02630*. 2024
- [5] Dixon M, Halperin I. G-learner and girl: Goal based wealth management with reinforcement learning. *arXiv preprint arXiv:2002.10990*. 2020
- [6] Domin H. AI Governance Trends: How Regulation, Collaboration and Skills Demand Are Shaping the Industry. Cologny (Geneva), Switzerland: World Economic Forum; 2024. Available from: <https://www.weforum.org/stories/2024/09/ai-governance-trends-to-watch>
- [7] Finn T, Downie A. Agentic AI Vs. Generative AI – Key Differences. IBM Think Blog; 2023. Available from: <https://www.ibm.com> [Accessed: July 27, 2025]
- [8] Gomstyn A, Jonker A. New Ethics Risks Courtesy of AI Agents? Armonk, NY, USA: IBM; 2024. Available from: <https://www.ibm.com> [Accessed: July 27, 2025]
- [9] Gridach M, Nanavati J, Abidine K, Mendes L, Mack C. *Agentic AI for Scientific Discovery: A Survey of Progress, Challenges, and Future Directions*. Ithaca, NY, USA: arXiv (Cornell University); 2025. Available from: <https://arxiv.org/abs/2503.08979>
- [10] Istrate O. AI agents in education: An early systematic review of emerging roles, potential, and limitations. *Revista De Pedagogie Digitala*. 2024;**3**:24-30
- [11] Maka SR, Boppana SB, Sadaram G, Katnapally N, Karaka LM, Sakuru M. Automating cyber threat response using agentic AI and reinforcement learning techniques. *Journal of Electrical Systems*. 2021;**17**(4):138-148
- [12] Córdova-Esparza DM. AI-powered educational agents: Opportunities, innovations, and ethical challenges. *Information*. 2025;**16**(6):469
- [13] Kshetri N. Transforming cybersecurity with agentic AI to combat emerging cyber threats. *Telecommunications Policy*. 2025;**49**(6):102976
- [14] Lindy AI. *AI Agent Architecture in 2025 – Guide*. Thornaby, United Kingdom: Lindy; 2025. Available from: <https://www.lindy.ai> [Accessed: July 27, 2025]
- [15] Mastrogiacomì F. Facilitating a paradigm shift for teaching and learning with AIs. *Italian Journal of Educational Technology*. 2024;**32**:69-81
- [16] O'Brien P, Wiegand M. Agent based process management: Applying

intelligent agents to workflow. The Knowledge Engineering Review. 1998;**13**:161-174

[17] Oertel C, Castellano G, Chetouani M, Nasir J, Obaid M, Pelachaud C, et al. Engagement in Human-Agent Interaction: An Overview. *Frontiers in Robotics and AI*. 2020;**7**:92

[18] Paul A, Yu CL, Susanto EA, Lau NWL, Meadows GI. AgentPeerTalk: Empowering students through agentic-AI-driven discernment of bullying and joking in peer interactions in schools. arXiv preprint arXiv:2408.01459. 2024

[19] Paul A, Yu C, Susanto E, Lau N, Meadows G. Agent-PeerTalk: Empowering Students through Agentic-AI-Driven Discernment of Bullying and Joking in Peer Interactions in Schools. 2024

[20] Plc P. 2024 AI Education Report (Higher Education). London, United Kingdom: Pearson; 2024

[21] Zhao H, Wang Y, Bashford-Rogers T, Donzella V, Debattista K. Exploring generative AI for sim2real in driving data synthesis. In: 2024 IEEE Intelligent Vehicles Symposium (IV); Jeju Island, Korea. 2024. pp. 3071-3077

Decentralized Energy Management Using Multi-Agent Systems for Remote Hybrid Power Systems

*Puteri Nor Aznie Fahsyar, Nohaidda Sariff
and Zool Hilmi Ismail*

Abstract

Hybrid Renewable Energy Systems (HRES) are increasingly adopted to provide reliable electricity in rural and remote areas lacking access to centralized grids. This chapter explores the development and validation of a Multi-Agent System (MAS) framework designed to enhance the control and coordination of off-grid HRES integrating solar photovoltaic, wind turbines, battery storage, a diesel generator, and an inverter. The decentralized architecture assigns autonomous agents to each energy component, overseen by a central Control Agent that ensures optimal energy dispatch, fuel minimization, and system stability. The framework was evaluated over a 92-day period using real meteorological and load data. The MAS-based control strategy demonstrated significant improvements in operational efficiency, achieving over 90% reduction in fuel consumption and operating costs compared to conventional diesel-only systems. Solar photovoltaic emerged as the dominant energy source, with battery storage playing a key role in mitigating renewable intermittency and maintaining reliability. Economic assessment showed a 25% reduction in total system cost and a levelized cost of energy of \$0.276/kWh, with an investment payback period of less than 2 years. These findings highlight the potential of MAS as an intelligent, scalable, and cost-effective solution for managing decentralized energy systems in underserved regions. The approach supports broader goals of enhancing energy access, promoting renewable integration, and advancing sustainable development in off-grid contexts.

Keywords: multi-agent system, hybrid renewable energy system, off-grid electrification, decentralized energy management, techno-economic analysis

1. Introduction

Providing reliable, sustainable, and affordable electricity to remote and rural communities remains a major global challenge. These areas are often not connected to the main electricity grid due to high extension costs, difficult terrain, and low population density. As a result, many rural communities experience energy poverty, which

limits access to education, healthcare, and economic opportunities widening the gap between rural and urban areas. With limited grid infrastructure available, off-grid electrification solutions have become essential. However, most of these communities still rely heavily on diesel generators, which pose serious issues such as high and unstable fuel prices, costly fuel transportation to remote areas, and overall expensive electricity for users. The impact of energy shortages in rural areas is a serious barrier to overall human development. Without affordable and reliable electricity, daily life is affected, economic activities are limited, and social progress is slowed. For instance, lack of electricity reduces study time for children, weakens healthcare services that rely on powered equipment or refrigeration, and holds back the growth of small businesses. These challenges directly hinder progress on several United Nations Sustainable Development Goals (SDGs), not just SDG7 (Affordable and Clean Energy), but also goals related to poverty (SDG1), education (SDG4), health (SDG3), and economic growth (SDG8) [1].

Traditional energy solutions, especially standalone diesel generators, have been widely used for off-grid power in rural areas. However, this method is neither economically nor environmentally sustainable. High and unpredictable fuel prices, along with the challenges of delivering diesel to remote locations, make electricity costly for rural families and small businesses. In addition, burning diesel fuels local air pollution and increases greenhouse gas emissions, which goes against global efforts to promote sustainability. Even when Hybrid Renewable Energy Systems (HRES) which combine renewable sources like solar and wind with conventional backups, traditional design and control methods come with their own challenges [2]. One major issue is the risk of oversizing or undersizing the system. Oversized systems waste money on equipment that is not fully used, while undersized systems fail to meet the energy needs of the community. In addition, conventional centralized control systems often lack the flexibility to manage the changing nature of renewable energy and the fluctuating energy demands in real time, especially in remote off-grid settings.

2. Integrating multi-agent systems into HRES for decentralized management

2.1 Overview of HRES

HRES have become a promising solution for providing electricity to off-grid and remote areas, offering a practical mix of reliability and sustainability. These systems typically combine two or more energy sources such as solar photovoltaic (PV) panels and wind turbines along with conventional backup sources like diesel generators. Batteries are also included to store excess energy. To manage the flow of electricity and maintain power quality, power converters and advanced controllers play an essential role.

The common HRES setup that includes solar PV, wind turbines, a diesel generator, and a battery energy storage system (BESS) is illustrated in **Figure 1**. Compared to diesel-only systems, HRES offer many advantages. They reduce reliance on imported fossil fuels, lower greenhouse gas emissions, and have a smaller environmental impact. By using local renewable resources, these systems can improve the reliability of electricity supply and reduce long-term costs due to lower fuel use and less maintenance for renewable components. HRES offer several advantages over traditional

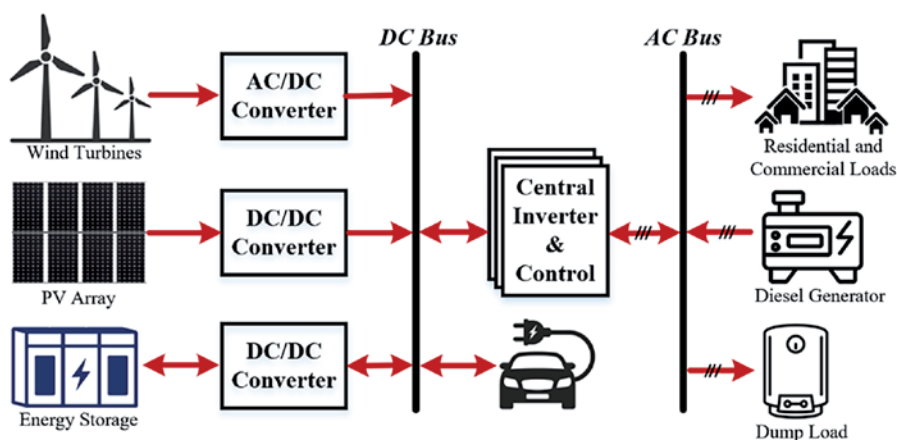


Figure 1.
 Hybrid renewable energy system [3].

diesel-only power systems, particularly in remote and off-grid regions. One of the most prominent benefits is the reduced reliance on imported fossil fuels, which can be costly and subject to geopolitical price fluctuations. By incorporating renewable energy sources, HRES reduce the operational dependency on diesel fuel, leading to more stable and predictable long-term energy costs. This transition not only enhances national energy security but also aligns with global goals of reducing greenhouse gas emissions and mitigating climate change impacts [4, 5]. Although the initial capital investment for HRES can be significantly higher due to the integration of renewable generation technologies and energy storage systems, the overall lifecycle cost is typically lower than that of diesel-only systems. This is because renewable energy incurs minimal ongoing fuel costs and demands less frequent maintenance, especially when compared to diesel generators with many moving parts and short service intervals.

2.2 Challenges in operating HRES

Despite their numerous advantages, HRES face technical and operational challenges. One of them is the intermittent and variable nature of renewable energy sources such as solar irradiance and wind speed. These fluctuations make it difficult to maintain a continuous and stable power supply without effective control and forecasting mechanisms. Real-time energy balancing becomes critical, particularly in systems where renewable penetration is high and dispatchable backup sources are limited [6, 7]. Energy storage systems are essential for absorbing surplus energy and supplying power during periods of low generation. However, the optimal sizing of these storage units is a complex task. Oversized batteries may lead to excessive capital expenditures and inefficient usage, while undersized systems can suffer from accelerated degradation and high cycling frequencies. Additionally, the integration of heterogeneous components such as solar PV panels, wind turbines, diesel generators, and batteries requires sophisticated control systems capable of managing their interactions dynamically and reliably. These control systems must ensure operational efficiency and system stability, particularly during peak demand or low generation intervals [8].

2.3 Trends and optimization in HRES

The global interest in HRES has been steadily increasing, particularly in developing regions such as Southeast Asia, Sub-Saharan Africa, and North Africa, where access to reliable electricity remains a challenge. Governments, NGOs, and private stakeholders have shown growing interest in hybrid systems as a viable solution to bridge the energy access gap. Technological developments have further accelerated the deployment of HRES in these regions. Advances in microcontrollers, IoT-enabled sensors, and smart metering technologies have significantly enhanced system monitoring, control, and maintenance capabilities, reducing both operational risk and costs [9]. Moreover, there has been a growing interest in intelligent control strategies such as MAS, which enable autonomous coordination of energy generation, storage, and consumption. MAS frameworks distribute decision-making across system components allowing agents like solar, battery, or load units to act independently yet collaboratively, thus improving responsiveness and scalability in dynamic environments.

Given the nonlinear, dynamic, and multi-objective nature of HRES, optimization plays a crucial role in their design and operation. The integration of renewable energy introduces complexity in balancing cost, reliability, and environmental impact. Traditional deterministic and gradient-based optimization techniques often fall short when dealing with the uncertainties and complexities of renewable energy availability and load profiles. Consequently, metaheuristic algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Simulated Annealing (SA) have become widely adopted for HRES applications. These techniques are capable of navigating large, multidimensional search spaces while simultaneously handling multiple conflicting objectives, making them suitable for optimizing system configurations under various environmental and economic constraints. Optimization in HRES primarily aims to minimize system costs while maintaining a desired level of performance and reliability. Typical decision variables in the optimization process include the installed capacity of photovoltaic modules, wind turbines, battery banks, and diesel generators. These variables are subject to constraints such as power balance, battery state-of-charge limits, and reliability indices like the Loss of Power Supply Probability (LPSP). The objective functions most commonly targeted are the minimization of Total Net Present Cost (TNPC) and Cost of Energy (COE), as well as maximization of renewable energy penetration and system autonomy [10–12].

3. System design and architecture for off-grid energy management

3.1 Overview of the proposed off-grid HRES

The system architecture, as illustrated in the **Figure 2**, integrates several key energy components: photovoltaic (PV) panels, a wind turbine, a diesel generator for backup power, a battery storage system to manage energy surpluses and deficits, and an inverter for necessary power conversions.

In this study, the modeling and simulation of this HRES were conducted using MATLAB, which served as the platform for implementing a MAS framework. Within this framework, each physical component of the HRES is represented as an autonomous agent with its own defined characteristics and behaviors. Furthermore, PSO, a metaheuristic optimization technique, was employed to determine the most

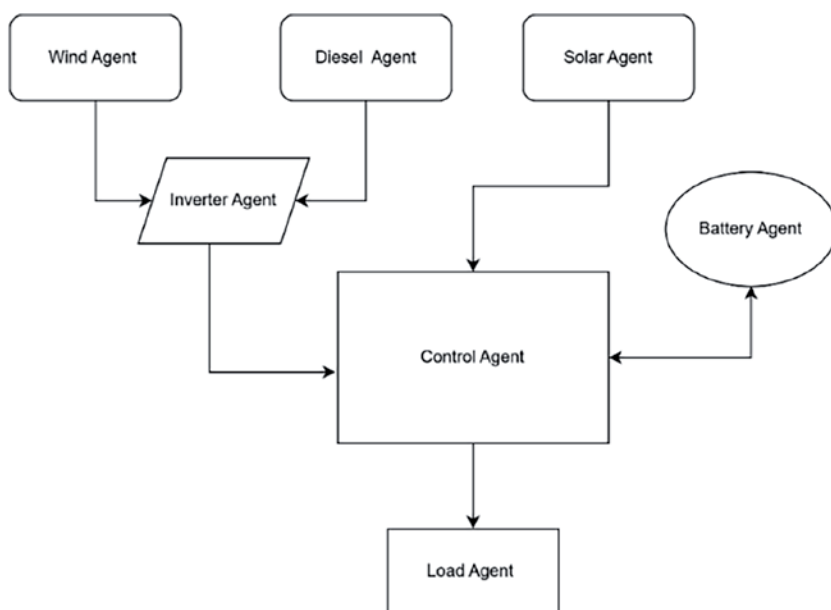


Figure 2.
 System agent diagram.

cost-effective sizing for the renewable energy components (PV panels and wind turbine) and the capacity of the battery storage system, ensuring an economically optimized configuration.

3.2 Component agents and their specifications

The selection of the components and their specifications is based on commercially available products to ensure a degree of realism in the simulation.

- *Solar Agent:* This agent represents the photovoltaic array, which is the primary renewable energy source. It comprises 12 Panasonic HIT N340 solar panels, each with a rated power output of 340 W and a module efficiency of 20.3%. The total installed PV capacity is therefore $12 \times 340 \text{ W} = 4.08 \text{ kW}$. The cost per panel is approximately \$359.72. The agent calculates power output based on solar irradiance, panel area, efficiency, and the number of panels.
- *Wind Agent:* This agent models a Pikasola 200 W Vertical Axis Wind Turbine (VAWT). It has a rated power output of 200 W, with a start-up wind speed of 1.5 m/s, a rated wind speed of 11.5 m/s, and a cut-out wind speed of 45 m/s. The cost of the turbine is approximately \$202.50. The agent determines power generation based on wind speed and the turbine's power curve.
- *Battery (Storage) Agent:* An EG Solar 10kWh LiFePO4 (Lithium Iron Phosphate) battery is represented by this agent. It has a nominal capacity of 10.24 kWh, with specified charge and discharge efficiencies of 95%. The operational State of Charge (SoC) is managed between a minimum of 10% and a maximum of 100% to protect battery health and prolong its lifespan. The approximate cost of this battery unit is \$990.00.

- *Diesel Agent:* A Generac XD5000E diesel generator, with a rated power output of 5 kW, serves as the backup power source. Its fuel consumption is modeled using specific coefficients ($a = 0.246$ L/kWh for load-dependent consumption and $b = 0.084$ L/h for baseline consumption). The fuel price was assumed to be \$0.20 per liter for the simulation. The agent calculates power output, fuel used, and operational cost when activated.
- *Inverter Agent:* An SMA Sunny Island 6.0H inverter, with a rated power of 6 kW and an efficiency of 95%, is modeled by this agent. Its primary function is to convert DC power from the solar panels and battery into AC power for the load. It also handles AC power from the diesel and wind turbine, converting it to DC for system compatibility before delivery to the load agent, as indicated by the system diagram and control logic.
- *Load Agent:* This agent represents the electricity demand of the off-grid household or facility. It utilizes a fixed hourly load profile, with a total daily energy consumption of 33.5 kWh. This profile includes typical loads such as lighting, appliances, and air conditioning, providing a realistic demand pattern for the simulation.

3.3 Environmental context and load profile

The performance and realism of the HRES simulation are heavily dependent on the environmental data and load characteristics used. The simulations were specifically contextualized to the local weather conditions of one of the area in Selangor, Malaysia. Real hourly meteorological data, encompassing solar irradiance and wind speed, were sourced for a 92-day period. This use of actual, site-specific environmental data is crucial because the output of renewable energy components, particularly solar PV and wind turbines, is directly governed by these local weather patterns. Employing generic or averaged data could lead to significant inaccuracies in performance prediction and potentially result in suboptimal system sizing either oversizing components, leading to wasted investment, or undersizing them, resulting in an unreliable power supply [13]. The demand side of the system was represented by a fixed daily load profile, averaging 33.5 kWh per day, characteristic of a typical rural household or small facility in the region. This hourly profile accounts for typical variations in energy consumption throughout a 24-hour cycle. While a fixed load profile is a simplification (as real-world demand can vary daily and seasonally), it provides a consistent and repeatable baseline against which the performance of the HRES and the effectiveness of the MAS control strategy can be rigorously evaluated. This approach ensures that observed variations in system performance are primarily attributable to changes in renewable resource availability and the control system's response, rather than fluctuating demand patterns.

4. The MAS control framework

4.1 The central role of the control agent

The main part of the MAS framework is the Control Agent, which functions as the central intelligence of the entire off-grid HRES. Unlike the other agents that

represent physical hardware components, the Control Agent is a software-based entity, meticulously modeled and implemented within the MATLAB environment. Its primary responsibility is to continuously monitor the operational status of all connected agents. By gathering real-time data from these agents, such as current generation levels, load demand, and the battery's SoC, the Control Agent makes informed, dynamic decisions to ensure a consistent balance between energy supply and demand within the system. Its central coordinating position is visually depicted in the System presented in the source thesis.

4.2 Decision-making logic and energy source prioritization

The Control Agent employs a clearly defined, priority-based control strategy to manage the flow of energy within the HRES. This strategy is designed to maximize the utilization of renewable energy sources, minimize reliance on the diesel generator, and ensure the longevity of the battery storage system. The decision-making hierarchy is as follows:

1. *Maximize renewable energy utilization:* The system prioritizes the use of energy generated by the Solar Agent first, supplemented by any available energy from the Wind Agent. The combined output from these renewable sources is the first to be dispatched to meet the load demand.
2. *Battery charging with surplus energy:* If the total renewable energy generation exceeds the current load demand, the surplus power is directed to the Battery Agent for charging. This ensures that excess clean energy is stored for later use rather than being curtailed or wasted.
3. *Battery discharging to meet deficits:* If the load demand is greater than the available renewable energy generation, the Control Agent instructs the Battery Agent to discharge and supply the deficit. The battery thus acts as a crucial buffer, smoothing out the intermittency of renewables.
4. *Diesel generator as last resort:* The Diesel Agent is activated only when both renewable energy sources and the battery are insufficient to meet the load demand. Specifically, the diesel generator engages when the battery's SoC reaches its pre-defined minimum threshold (10% in this study) and a power deficit still exists.

To implement this logic, the Control Agent continuously evaluates the total available power, meticulously accounting for any necessary AC/DC or DC/AC conversions managed by the Inverter Agent and compares this available power against the current load demand reported by the Load Agent. This ensures an accurate assessment of the energy balance at each time step. A detailed flowchart illustrating the MATLAB-based decision logic of the Control Agent is provided in the Appendix (**Figure A1**) to support algorithm transparency and reproducibility.

4.3 Real-time data collection and action dispatching

The control process orchestrated by the Control Agent is cyclical, typically operating on an hourly basis corresponding to the resolution of the input environmental data and load profiles. During each simulation step, the Control Agent gathers fresh

data from all other agents. Based on this instantaneous snapshot of the system's state, including current solar irradiance, wind speed, load demand, and battery SoC, it executes the decision-making logic outlined above. Once a decision has been made, the Control Agent dispatches the appropriate commands to the respective agents. It then updates the internal state of each agent to reflect the actions taken and the resulting energy flows. This continuous loop of data collection, decision-making, and action dispatching ensures that the HRES operates in a synchronized and responsive manner, capable of adapting to the highly variable nature of renewable energy generation and load consumption in real-time. This dynamic adaptation is a fundamental advantage over simpler, rule-based control systems that might rely on fixed operational thresholds or predetermined schedules. Such static systems often struggle to optimize resource utilization or respond effectively to unexpected changes, such as a sudden drop in solar irradiance due to cloud cover or an unpredicted spike in load. The MAS framework, by contrast, makes decisions based on the actual, immediate state of all components and environmental inputs, leading to more efficient energy management and improved system resilience [14].

5. Performance evaluation

5.1 Energy contribution by source

The 92-day simulation provided detailed insights into the energy dynamics of the HRES under MAS control. The contributions of each energy source to meeting the total load demand of 3082.00 kWh were as follows:

- *Solar dominance:* Solar PV was unequivocally the primary energy contributor. The solar array produced 1518.44 kWh of energy over the simulation period. This underscores the significant potential of solar energy in the Malaysian context.
- *Negligible wind contribution:* The wind turbine's contribution was minimal, producing only 3.93 kWh of energy. This low output was attributed to generally insufficient wind speeds in the chosen dataset for the Subang location or potentially a suboptimal choice of wind turbine for the prevailing conditions. The combined renewable fraction (solar and wind) was 49.4% of the total load served.
- *Critical battery role:* The 10.24 kWh LiFePO₄ battery played an essential role as an energy buffer. It absorbed 222.85 kWh of surplus renewable energy during periods of high generation and discharged 215.50 kWh to meet load demand when renewable output was insufficient. This buffering capability was crucial in minimizing the operational hours of the diesel generator and enhancing overall system efficiency.
- *Reduced diesel usage:* The diesel generator was activated by the MAS control logic only as a last resort. Over the 92 days, it supplied 1344.13 kWh of energy, consuming 531.33 liters of fuel at a total cost of \$106.27. This represents a substantial reduction compared to a hypothetical diesel-only scenario, which was simulated to require 10273.33 liters of fuel at a cost of \$2054.67 to meet the same load demand over the same period. This highlights a fuel cost saving of over 90% due to the hybrid system and MAS control.

Source	Energy supplied/ managed (kWh)	Percentage of total load served (3082.00 kWh)	Remarks
Solar energy	1518.44	49.27%	Primary renewable source
Wind energy	3.93	0.13%	Negligible contribution
Battery discharged	215.50	6.99%	Energy supplied from storage
Diesel energy	1344.13	43.61%	Backup generation
Battery charged	222.85	N/A	Surplus renewable energy stored
Unserved load	78.93	2.56%	Load not met by the system

Table 1.
 Energy contribution breakdown.

The energy contributions are summarized in **Table 1**.

5.2 MAS control effectiveness and system reliability

The system demonstrated a generally high level of reliability in meeting the load demand. Over the 92-day period, 3082.00 kWh of load was served. However, an unserved load (deficit) of 78.93 kWh was recorded, representing approximately 2.56% of the total demand. While the system was largely effective, this unserved energy indicates periods where the combined capacity of the HRES components, even under MAS control, could not fully meet the instantaneous demand. This could be attributed to factors such as prolonged periods of low renewable resource availability leading to battery depletion before the diesel generator could cover the entire shortfall, or momentary peak loads exceeding the combined available generation [15]. This observation is valuable, as it highlights a potential area for further system optimization, such as refining component sizing incorporating demand-side management strategies, a point identified for future work. Nevertheless, the presence of the diesel generator, even with its minimized usage, proved crucial as a backup to limit the extent of unserved load. Although the unserved load remains relatively low, its occurrence reflects the inherent challenges associated with renewable energy intermittency in off-grid systems. To further mitigate unserved energy, several strategies can be considered, including increasing battery storage capacity, optimizing renewable energy sizing, and implementing demand-side management measures such as load prioritization or load shifting. In addition, enhancing the MAS control framework with predictive or adaptive decision-making based on short-term renewable and load forecasts could enable earlier intervention during critical operating conditions, thereby further improving overall system reliability.

6. Techno-economic viability

6.1 Comparative economic metrics

The HOMER Pro simulations yielded comprehensive economic metrics, allowing for a robust assessment of the hybrid system's financial viability, especially when compared against a diesel-only base case scenario.

- *Net Present Cost (NPC)*: The optimized hybrid system exhibited an NPC of \$43,092 over its 25-year lifetime. This was significantly lower by 24.6% than the NPC of the diesel-only base case, which was \$57,094. The NPC considers all costs incurred throughout the project lifetime, discounted to their present values.
- *Levelized Cost of Energy (LCOE)*: The LCOE for the hybrid system was calculated to be \$0.276/kWh. This is substantially more favorable than the LCOE of \$0.377/kWh for the diesel-only system, indicating that energy produced by the hybrid system is cheaper on a per-unit basis over the project's life.
- *Initial Capital Cost*: The hybrid system required a higher initial capital investment of \$14,978, compared to \$4284 for the diesel-only base case (which primarily involves the generator cost). This higher upfront cost is typical for renewable-based systems.
- *Operating & Maintenance (O&M) Cost*: The annual O&M cost for the hybrid system was \$2199/year, which is lower than the O&M costs associated with a continuously running diesel-only system (primarily driven by fuel costs).
- *Payback Period*: A simple payback period of 5.4 years for the hybrid system when compared against the diesel-only base case. This is the time taken for the cumulative cost savings from the hybrid system to offset its higher initial investment. A separate manual calculation, based on the fuel savings observed in the 3-month simulation (annualized fuel cost saving of \$9059.56 and an upfront MAS component cost of \$14,798), yielded a shorter payback period of approximately 1.63 years. The discrepancy arises because the calculation is more comprehensive, incorporating full lifecycle costs including O&M, component replacements and the time value of money, whereas the manual calculation focused primarily on direct fuel savings. Both figures, however, suggest a reasonable timeframe for recouping the investment.
- *Internal Rate of Return (IRR)*: The IRR is 18%.
- *Return on Investment (ROI)*: The ROI was calculated at 14%.

These metrics collectively demonstrate that despite the higher initial capital outlay, the MAS-controlled hybrid system is economically superior to a diesel-only system over the long term, driven by significantly lower fuel and operational costs [16].

6.2 Long-term energy contribution

Over the simulated 25-year project lifetime, projected the following energy contributions to meet the annual load of 12,212 kWh:

- *Solar PV*: 8791.82 kWh/year, accounting for 72.0% of the total energy supplied.
- *Diesel Generator*: 3159.85 kWh/year, contributing 25.9%.

Metric	MAS-HRES (25 years)	MAS-HRES (92 days)	Diesel-only (25 yrs)
Net present cost (NPC)	\$43,092	\$14,792 (Initial Cost +3-month fuel)	\$57,094
Levelized cost of energy	\$0.276/kWh	N/A	\$0.377/kWh
Initial capital cost	\$14,978	\$14,798 (Components for MAS)	\$4284
Annual O&M cost	\$2199/year	N/A (Fuel cost \$106.27 for 92 days)	\$4131/year (Primarily Fuel)
Payback period	5.4 years	1.63 years (vs. MATLAB Diesel-only)	N/A
Renewable fraction	72%	49.4%	0%
Diesel fuel cost (92 days)	N/A (Part of O&M)	\$106.27	\$2054.67 (MATLAB Diesel-only)

Table 2.
Comparative techno-economic performance: MAS-HRES vs. Diesel-only.

- *Battery Discharge:* 260.75 kWh/year, providing 2.1% of the energy.
- *Wind Turbine:* Contribution remained negligible (effectively 0.0%).

The comparative techno-economic performance is summarized in **Table 2**.

It should be noted that variations in diesel fuel price and battery degradation can affect long-term system cost and performance. Higher fuel prices would increase diesel operating costs, further favoring the MAS-HRES configuration, while battery aging would reduce effective storage capacity and may increase diesel usage. Nevertheless, the proposed MAS control strategy inherently minimizes diesel dependency and adaptively manages battery usage, indicating robust techno-economic performance under such parameter variations.

7. Conclusion and future recommendation

The development and evaluation of a MAS-based control framework for managing off-grid HRES revealed promising outcomes in terms of both operational performance and economic feasibility. Through agent-based simulations using real environmental and load data, the system demonstrated effective energy balancing and autonomous coordination among energy components. A notable reduction in diesel consumption over 90% approximately 53.6% over a 92-day period highlights the capacity of the MAS to prioritize renewable sources and minimize reliance on fossil fuels. Complementary techno-economic analysis conducted projected long-term benefits over a 25-year project lifetime, including a 25% reduction in NPC and a favorable LCOE. The estimated payback period, ranging from 1.63 years based on short-term fuel savings to 5.4 years via lifecycle modeling, further confirms the economic attractiveness of the proposed system.

Building upon the findings and limitations of this study, several recommended to further advance the field of intelligent off-grid HRES:

- *Dynamic Load Modeling and DSM:* Integrate dynamic load profiles that reflect seasonal, daily, and behavioral variations in energy consumption. Incorporate Demand-Side Management algorithms within the MAS framework in MATLAB to actively manage loads, shift deferrable consumption, and improve overall system efficiency and battery health.
- *Advanced AI and Predictive Control:* Enhance the MAS with more sophisticated artificial intelligence techniques, such as reinforcement learning for adaptive control strategies or predictive optimization algorithms that utilize weather and load forecasts to make proactive energy management decisions.
- *Wind Energy Optimization:* Conduct a thorough reassessment of wind turbine sizing and selection based on detailed, location-specific wind resource assessments to more accurately evaluate and potentially enhance the contribution of wind power to the energy mix.
- *Enhanced Simulation Coupling:* For more robust validation, explore methods for closer coupling between dynamic simulation tools like MATLAB and techno-economic tools like HOMER Pro. This could involve purchasing the full version of HOMER to enable the import of hourly dispatch schedules from the MAS simulation, allowing for a more accurate economic analysis of the specific control strategy.
- *Hardware Implementation and Pilot Projects:* Progress from simulation to physical implementation through hardware-in-the-loop (HIL) testing to validate control algorithms with real hardware components. Subsequently, develop pilot deployment projects in target rural communities to test the system in a real-world operational environment and gather practical field data. The inherent scalability of MAS, potentially using microcontrollers like Arduino or Raspberry Pi for agent implementation, supports this transition.
- *Socio-Economic Impact Assessment:* Following potential pilot deployments, conduct comprehensive socio-economic impact studies to quantify the benefits of the MAS-HRES on the quality of life, economic activities, and overall development of the electrified communities.

Acknowledgements

The authors acknowledge the use of artificial intelligence-based language tools, specifically ChatGPT (OpenAI), to assist with language polishing, grammar checking, and clarity improvement during the preparation of this chapter. The use of these tools was limited to language refinement only. All technical content, analysis, interpretations, and conclusions were entirely developed by the authors, who take full responsibility for the accuracy and integrity of the work.

Appendix

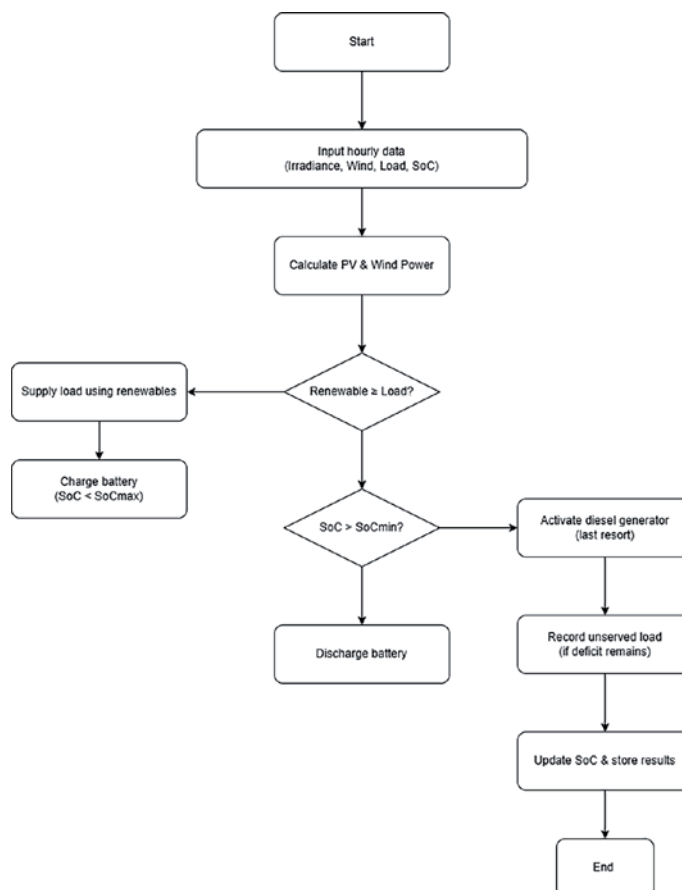


Figure A1.
 Flowchart of the MATLAB-Based MAS Control Algorithm for the Off-Grid HRES.

Author details

Puteri Nor Aznie Fahsyar^{1*†}, Nohaidda Sariff^{2†} and Zool Hilmi Ismail^{3†}

1 Faculty of Engineering, Higher College of Technology, Abu Dhabi,
United Arab Emirates

2 School of Engineering, Taylor's University, Subang Jaya, Selangor, Malaysia

3 Center for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia,
Jalan Sultan Yahya Petra, Kuala Lumpur, Malaysia

*Address all correspondence to: pfahsyar@hct.ac.ae

† These authors contributed equally.

IntechOpen

© 2026 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] International Energy Agency (IEA). Energy Access Outlook 2022: Securing Access to Affordable, Reliable, Sustainable and Modern Energy for all. Paris, France: International Energy Agency; 2022
- [2] Mandelli S, Barbieri J, Mereu R, Colombo E. Off-grid systems for rural electrification in developing countries: Definitions, classification and a comprehensive literature review. *Renewable and Sustainable Energy Reviews*. 2016;**58**:1621-1646. DOI: 10.1016/j.rser.2015.12.338
- [3] Gajewski P, Pieńkowski K. Control of the hybrid renewable energy system with wind turbine, photovoltaic panels and battery energy storage. *Energies*. 2021;**14**(6):1595. DOI: 10.3390/en14061595
- [4] Rakib MW, Munna AH, Farooq T, Agili B. Enhancing grid stability and sustainability: Energy-storage-based hybrid systems for seamless renewable integration. *European Journal of Electrical Engineering and Computer Science*. 2024;**8**(3):1-8. DOI: 10.24018/ejece.2024.8.3.618
- [5] Come Zebra EI, van der Windt H, Nhumaio G, Faaij A. A review of hybrid renewable energy systems in mini-grids for off-grid electrification in developing countries. *Renewable and Sustainable Energy Reviews*. 2021;**144**:111036. DOI: 10.1016/j.rser.2021.111036
- [6] Sahoo S, Timmann P. Energy storage technologies for modern power systems: A detailed analysis of functionalities, potentials, and impacts. *IEEE Access*. 2023;**11**:49689-49729. DOI: 10.1109/ACCESS.2023.3274504
- [7] Wu Y, Chen X, Khan M. Hybrid renewable energy systems-a review of optimization approaches and future challenges. *Applied Sciences*. 2023;**15**(4):1744. DOI: 10.3390/app15041744
- [8] Sharma R, Khare V, Jaiswal AK. Modelling and optimal energy management for battery energy storage systems in renewable energy systems: A review. *Renewable and Sustainable Energy Reviews*. 2022;**160**:112271. DOI: 10.1016/j.rser.2022.112271
- [9] Eltamaly AM, Alotaibi MA, Alolah AI, Ahmed MA. IoT-based hybrid renewable energy system for smart campus. *Sustainability*. 2021;**13**(15):8555. DOI: 10.3390/su13158555
- [10] Rahmani N, Mostefai M. Multi-objective MPSO/GA optimization of an autonomous PV-wind hybrid energy system. *Engineering, Technology & Applied Science Research*. 2022;**12**(4): 8817-8824. DOI: 10.48084/etasr.4877
- [11] Ma X, Abuelrub A, Khamees M, Al-Sumaiti A, Alhajeri H. Hybrid energy system design using greedy particle swarm and biogeography-based optimisation. *IET Renewable Power Generation*. 2020;**14**(10):2734-2744. DOI: 10.1049/iet-rpg.2019.0858
- [12] Beyene TT, Misganu TM, Gutema MW. Multi-criteria optimization of a hybrid renewable energy system using particle swarm optimization for optimal sizing and performance evaluation. *Clean Technologies*. 2025;**7**(1):23. DOI: 10.3390/cleantechnol7010023
- [13] Elobaid D, Dey A, Islam M, Vadera S. Techno-economic and environmental

analysis of a fully renewable hybrid energy system for sustainable power infrastructure advancement. *Sustainable Energy Technologies and Assessments*. 2022;**52**:102102. DOI: 10.1016/j.seta.2022.102102

[14] Dou D, Yahyaouy A, Boumhidi J. Layered management and hybrid control strategy based on multi-agent system theory for microgrids. *IET Renewable Power Generation*. 2019;**13**(10):1754-1765. DOI: 10.1049/iet-rpg.2019.0664

[15] Trevizan RD, Headley AJ, Geer R, Atcitty S, Gyuk I. Integration of energy storage with diesel generation in remote communities. *Journal of Energy Storage*. 2021;**8**(2):57-74. DOI: 10.1557/s43581-021-00013-9

[16] El Din AHM, Abbas MA, Al Hamed AS, et al. Techno-economic and environmental analysis of renewable mix hybrid energy system for sustainable electrification of Al Dhafra rural area in Oman. *Energies*. 2023;**16**(1):288. DOI: 10.3390/en16010288

Cutting-Edge Technologies in Agent Architecture for Use at the Microgrid Protection Level

Andres Mauricio Diaz Caicedo

Abstract

This chapter discusses cutting-edge technologies in agent architecture for microgrid protection, focusing on agent-based hierarchical control. It introduces adaptive multi-agent systems (MAS) that enhance protection schemes by assigning agents to intelligent electronic devices (IEDs) for tailored solutions in dynamic environments. The proposed methodologies aim to improve operational efficiency and fault tolerance through structured communication, validation mechanisms, and real-time control, ultimately redefining energy management in microgrids.

Keywords: microgrid, multi-agent system, wavelet indicators, edge computing, hierarchical protection

1. Introduction

In the evolving landscape of microgrid management, the need for robust and adaptive protection schemes has become increasingly critical. Traditional methodologies often fall short in addressing the complexities of dynamic topologies and power fluctuations, leading to vulnerabilities in system resilience. This section introduces agent-based hierarchical control as a transformative approach, leveraging adaptive multi-agent systems (MAS) to enhance protection strategies by assigning dedicated agents to intelligent electronic devices (IEDs), thereby facilitating efficient task execution and localized decision-making in diverse application zones.

2. Agent-based hierarchical control: From white-box dynamics to management strategies

In the evolving landscape of microgrid management, the need for robust and adaptive protection schemes has become increasingly critical. Traditional methodologies often fall short in addressing the complexities of dynamic topologies and power fluctuations, leading to vulnerabilities in system resilience [1]. This section introduces agent-based hierarchical control as a transformative approach, leveraging adaptive multi-agent systems (MAS) to enhance protection strategies by assigning dedicated

agents to intelligent electronic devices (IEDs), thereby facilitating efficient task execution and localized decision-making in diverse application zones.

The purpose of structured protection schemes is to formulate a multistaged solution tailored for diverse application zones. Typically found in networks with dynamic topologies and significant power fluctuations, the methodologies existing in the literature often lack adaptive protection features. The proposed method introduces adaptive multi-agent systems (MAS), assigning each agent to an IED (zones). These agents function as executive units, conducting specific tasks by means of logical selective mapping in local environments. The suggested method unveils two distinctive environments: serialized branches (localities) [2] and parallelized acquisition, arranging different DER according to their internal dynamics and their connection to the bus bar for state extraction [3].

The system defines two types of agents: the parametric-selective agent and the validation agent. The main goal is to create overlapping zones based on short-circuit grouping indices, enabling the parametric agent to gather feedback from a local dynamic detector. This detector separates logical outputs based on the fault type and conveys these outputs to network equipment, including DER, relays, and loads, to classify protective functions across multiple levels. The secondary agent verifies fault clearance, adjusts settings, and maintains protection coordination. This process is executed through a two-tier hierarchical process. The first tier establishes an updatable quick-reference table for operational scenarios to load adjustments chosen within the power system transient analysis program [4, 5].

Figure 1 illustrates a hierarchical approach to the integration of communication interfaces in power systems, highlighting three key levels: local, parallel, and higher. At the local level, emulations/simulations of distributed energy systems (DERs) are performed using environments such as MATLAB and IED test devices. The parallel level handles concurrent tasks related to substation signal data extraction and integration into frameworks such as ETAP/Power Factory [7]. Finally, the top level optimizes operational management using Python, applying methods such as Numba for the validation

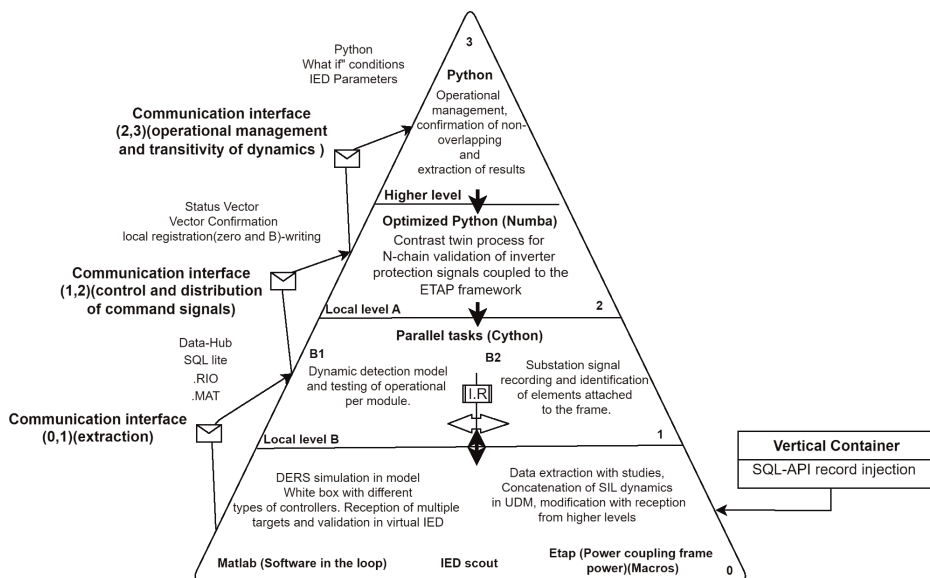


Figure 1.
Index of structured programming (start-up simulation multi-stage) [6].

and analysis of protection signals. This framework ensures the transitivity of the dynamics and facilitates efficient command distribution and real-time data mining.

Incorporating edge computing into these schemes introduces transformative capabilities, allowing real-time decision-making to occur directly at the periphery of the network. By enabling intelligent electronic devices (IEDs) and distributed controllers to process data locally, edge computing minimizes latency and improves the responsiveness of the system under fault conditions. Architecturally, the use of edge computing leverages highly distributed processing nodes equipped with advanced hardware accelerators, such as ARM-based SoCs, GPUs, and FPGAs, to handle computationally intensive tasks, including fault classification and adaptive relay coordination. These devices employ microservice-oriented software designs, where containerized modules encapsulate protection algorithms. Technologies such as Kubernetes and lightweight orchestrators like K3s are employed to manage these services dynamically across the edge nodes, ensuring high availability and scalability even under dynamic system conditions.

The proposed scheme employs distributed intelligence, where each agent embedded in the edge infrastructure processes local signals using pre-trained neural networks optimized for on-device inference. These models are compressed via techniques like quantization and pruning to minimize computational overhead, enabling their deployment on devices with constrained resources. Communication between agents is facilitated using high-efficiency protocols like MQTT and CoAP, which allow low-latency data exchange across the protection zones. The distributed nature of this architecture ensures resilience against single-point failures, as localized decision-making remains functional even if remote communication links are disrupted.

Furthermore, edge computing strengthens the integration of parallel processing capabilities into structured protection schemes. By segmenting data acquisition and decision-making tasks across multiple edge devices, the system achieves substantial improvements in fault clearance times. For instance, edge clusters responsible for coordinating distributed energy resources (DERs) employ frameworks such as Dask to parallelize data processing workloads. These clusters aggregate measurements from serially connected DERs, mapping fault impacts dynamically across interconnected zones. Such architectures also utilize decentralized ledger technologies to securely record protection actions, ensuring traceability and compliance with operational requirements.

A critical aspect of this approach is the synchronization of edge processing nodes with centralized SCADA systems. Edge devices periodically validate their operational states by querying simulation datasets hosted in SCADA. This synchronization is achieved through the establishment of shared memory structures and reference models that adapt to changing system conditions. Fault scenarios are simulated in near real-time at the edge, allowing for rapid adjustments to relay settings and protection logic. These simulations utilize transient analysis libraries encoded in high-performance languages, with JIT compilers like Numba optimizing the execution of computational kernels for dynamic system modeling.

Despite these advancements, structured protection schemes must align with rigorous regulatory frameworks to ensure their safety, interoperability, and cybersecurity compliance [8]. Standards such as IEC 61850 play a pivotal role in defining the communication protocols and data models required for seamless integration of edge-based agents with traditional protection infrastructure. Additionally, adherence to IEEE 55 ensures that distributed energy resources operate within permissible voltage and frequency limits during fault conditions. These standards provide the baseline for designing interoperable systems that can adapt to both centralized and decentralized operational paradigms.

2.1 Hierarchical communication and data storage

This protection scheme uses distributed communication for adjacent agents and remote links for distant ones. Local devices update via memory, confirming faults through mapping tests and relaying event data to the central device for state transitions. All events are logged in the control center for offline analysis [9].

The fault detection framework using Kolmogorov–Smirnov chains ensures completeness and safety for nine-cycle records at up to 50 kHz sampling, focusing on impedance transitions in bus couplers for rapid fault classification. Time-sensitive networking (TSN) enables deterministic communication with ultralow latency and bounded jitter, ensuring timely transmission of critical fault data and precise synchronization across local agents, essential for high-speed decision-making in dynamic networks.

The protection system is designed to address the increasing complexity arising from the expansion of sub-areas at a national level, requiring advanced tools for distributed data management. Fault logs and operational metrics are distributed across multiple nodes to ensure high availability and reliability, even during hardware failures. The system employs block-level redundancy to efficiently process high-frequency sampled measured values (SMV), supporting both offline analytics and real-time historical queries. By integrating parallel computing frameworks, it enables the analysis of large protection event datasets while maintaining scalability to meet the demands of a growing network of sub-areas.

The proposed relay agents also encompass prioritization recognition functions, operating based on weighted event ordinal functions. In this scheme, overcurrent protections take precedence, followed by voltage, frequency, and, finally, internal islanding prevention protections. These agents relay information in the form of voltage and current block data across various domains, contrasting with the local coordination map. The hierarchical design allows each agent to act as an independent decision-making node while remaining synchronized with the system's overarching coordination protocols. This multilevel synchronization ensures that local protections remain effective even during partial network outages, achieving a balance between resilience and flexibility.

The communication protocol adheres to IEC-61850, with enhancements to support dynamic data streaming and advanced configuration of relays through a centralized controller. For faster and more reliable coordination, sampled measured values (SMV) are transmitted directly to relay agents, which utilize real-time transient data to adjust their protection settings dynamically. This configuration significantly reduces the reaction time to transient faults and enhances the system's capacity to adapt to changing conditions.

Local agents handle the observability and classification of dynamics for event selectivity, subsequently forming a logical chain of triggers. These triggers are passed to a decision layer, where the digital twin platform simulates various scenarios to validate and optimize the chosen protection strategies. The integration of a digital twin with transient simulation capabilities not only improves operational reliability but also minimizes the risk of unnecessary tripping in complex grid scenarios. The hierarchical nature of the system ensures that decisions validated at the digital twin level are implemented seamlessly, with adjustments communicated back to local agents and controllers via TSN-enabled channels.

Resilience is further enhanced through the implementation of distributed edge computing capabilities. Each agent is equipped with sufficient computational power

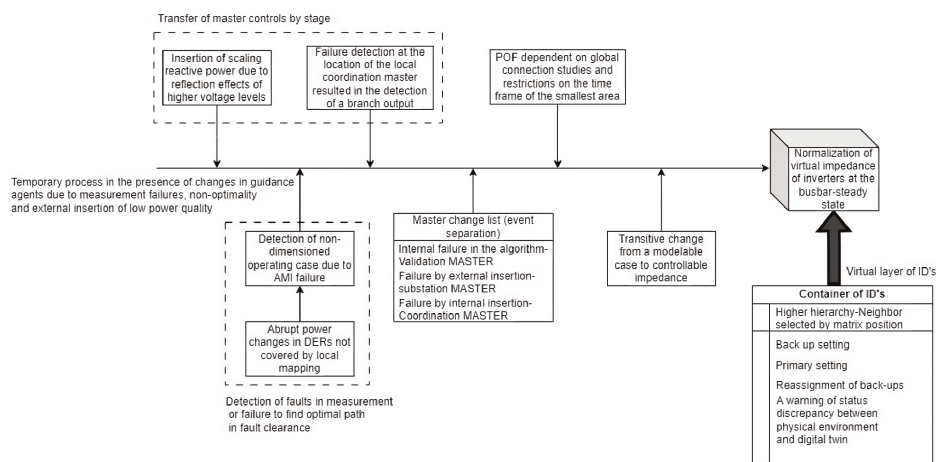


Figure 2. Temporary process in the presence of changes in guidance agents due to measurement failures, non-optimality, and external insertion of low power quality [6].

to execute critical protection algorithms locally, ensuring that decisions are made even in the absence of central controller connectivity. The memory system embedded in local devices is designed with modularity in mind, allowing for easy updates and capacity expansions as the network evolves. These edge nodes also incorporate AI-based anomaly detection algorithms to proactively identify potential system instabilities and initiate pre-emptive protective actions.

Flexibility is achieved by enabling agents to dynamically redefine their operational zones based on real-time network conditions. For instance, an agent detecting a significant voltage sag in its area of influence can autonomously expand its protection zone to cover adjacent areas, coordinating with nearby agents to ensure comprehensive fault coverage. This capability is critical for modern grids with high penetration of distributed energy resources (DER), where traditional fixed-zone protections are insufficient.

As shown in **Figure 2**, the temporary process outlines the transfer of master control across various management stages in response to measurement failures, non-optimal conditions, and external disturbances such as low power quality. This operational framework emphasizes the dynamic adaptability of the system, where events like abrupt changes in distributed energy resources (DERs) or undimensioned failures trigger protective responses.

A key component of this process involves the validation framework between the server and simulation layers. This step ensures that the optimization function is recalibrated to accommodate the specific requirements of local stages. Moreover, as the system transitions from an impaired state to a stabilized one, mechanisms such as transitive impedance changes and master coordination are implemented to maintain system resilience. Such coordinated actions underscore the importance of hierarchical control and the role of agents in sustaining operational continuity across interconnected grid zones.

2.2 Hardware and software protection-management agents

Protection agents manifest in two forms: hardware and software. The local aspect employs a distributed architecture anchored in single board computers (SBCs), where

dynamic detection algorithms are hosted. These algorithms facilitate the activation of a master switch mechanism during communication or acquisition agent failures. Such agents are designed to operate autonomously, leveraging network data and their direct coupling with the microgrid to make informed decisions. Communication protocols enable robust data exchange through feedback bimodal systems, which utilize probabilistic operational functions to validate state transitions between hierarchical levels. Effective communication among agents remains a cornerstone, enabling precise evaluation of the network's status and adaptive decision-making under rapidly changing conditions.

In microgrids dominated by inverter-based configurations, the intrinsic observational framework tasked with state assessment faces challenges in processing complex waveforms. Traditional sensitivity sweeps often fail to extract actionable insights, particularly when addressing high-frequency disturbances or irregularities in power quality. To overcome these limitations, cost-effective embedded cards are deployed on a per-board basis, acting as dynamic sensing nodes (**Figure 3**). These embedded devices are instrumental in enhancing the granularity of local measurements, enabling precise fault detection and state estimation. At the bay level, retaining and integrating adaptive schemes is crucial for system validation, especially in topologies where diverse command structures and fluctuating operational scenarios demand resilient and redundant architectures. This redundancy is achieved through hybrid communication standards that combine staggered feedback mechanisms with comparative assessments across distinct strategies, thus bolstering system reliability and responsiveness during contingencies.

The resolution of coordination phases necessitates rigorous validation processes. This begins with analyzing the physical controller's primary layer via a real-time complementary detection stage, ensuring rapid response times in transient simulations. Persistent branch disconnections, for instance, are categorized as operational deviations rather than systemic failures. This distinction ensures that the system maintains operability even amidst communication link disruptions, effectively preventing cascading failures. All events, including transient anomalies and operational adjustments, are meticulously logged for post-event analysis, providing operators with actionable insights to refine system performance further.

Flexibility in managing distributed energy resource (DER) states is another critical attribute of the proposed system. By dynamically moderating converter impacts on overcurrents and high-frequency voltage fluctuations, the system ensures seamless coordination among protection devices. This adaptability mitigates risks associated with fluctuating power quality while maintaining the integrity of interconnected network components. The operational dynamics are complemented by a layered approach to decision-making, where local agents execute first-level classifications, while higher-order algorithms validate and refine these decisions based on comprehensive system-wide metrics [10].

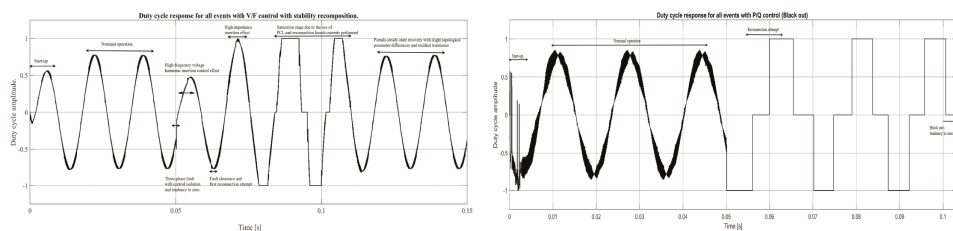


Figure 3.
Control responses in inverters due to controlled injection during faults.

A pivotal aspect of the proposed framework is the validation loop established between the physical server and simulation environments. This loop not only ensures real-time readjustment of optimization functions tailored to local stages but also facilitates impedance recalibrations during system recovery from a downed state. For example, changes in transitive impedance are executed to accommodate dynamic power flow adjustments and maintain operational stability. Concurrently, ID containers play an essential role in recalibrating command and control parameters, ensuring continuity and robustness under varying operational conditions.

The integration of redundancy within the system architecture is underscored by the dual-layered protection strategy, wherein hardware agents operate as the primary safeguards, and virtual layers provide backup mechanisms. This dual approach significantly enhances system resilience against unforeseen events, such as hardware failures or cyber intrusions. Furthermore, the optimization of resource allocation and the continuous adaptation of control algorithms directly contribute to mitigating economic losses by minimizing downtime and ensuring consistent energy delivery.

In conclusion, the orchestration of hardware and software protection-management agents within this multi-layered framework exemplifies the forefront of modern grid protection technologies. By leveraging real-time data processing, hierarchical communication, and adaptive optimization techniques, the system achieves unparalleled levels of reliability, flexibility, and economic efficiency in microgrid and DER management.

2.3 System identification for insertion (DERs by branch and coupling by converter)

Certainly, this process holds a pivotal role in the hierarchical protective structure, as it is responsible for establishing a sequence for each agent incorporated into the system in a modular manner [3]. This step is critical in laying down the foundational framework for analysis, ensuring adjustments validation and protection coordination. Notably, the topological data should be based on a pre-event structure. Synchronizing it would not only be resource-intensive but also lead to characterization problems due to dynamic overlap. Therefore, a global mapping of the microgrid must be organized through a concatenated digital mapping of three main structures. The first delineates the insertion ID, specifying the order in which the element enters the microgrid bus. The second structure focuses on the type of white box block used for the internal dynamics of serialized branches of DER. Lastly, the last structure details the black box of the parallelism groupings, highlighting the type of control for coupling to the bus. These dynamics are loaded onto a shared SIL coupling frame, entered as INT_DYN and IMP_Models, extracted from the program's SQL database through the Python connection interface [9, 11].

Given the ongoing changes in the system's topology, the operational mapping remains variable and requires consistent reordering. To accommodate this, a matrix of sub-indexes and states (**Figure 4**) is created, automatically adjusting with cycle variations. It's crucial not to concatenate registration information in the agent; thus, the system merely stores the size of an event frame. A hash table is established, assigning a management value to the grouping function based on the type of failure, its location, and the respective response in each case. This method facilitates organized searches by characteristics within the simulation program. To transmit information to a higher tier, a compartmentalized chain of the phenomenon is dispatched, comprising a 3-level identification ID (Branch, serial stage, and type of control). This information is

loaded into the physical environment, forming the basis for real-time verification of power system variations.

Although this architecture is primarily tailored for decentralized networks, it also holds potential for systemic-level formulation. Globally, distributed system operator (DSO) models with energy distribution grid edge (EDGE) boundaries are under development, aiming to integrate decentralized networks into a coordinated infrastructure capable of dynamically managing distributed resources with high flexibility [12]. However, in regions like Colombia, regulatory and technical constraints have hindered significant progress in DSO implementation. A critical challenge is the lack of modern technologies in older inverters, which prevents effective controllability during events. This limitation is especially significant in mitigating high-frequency oscillations or transient overcurrents that could compromise network stability. As a temporary solution, some functionalities in initial preset configurations are disabled to prevent high-frequency oscillations. While this approach mitigates short-term risks, it restricts the dynamic response capability of the network.

Upgrading existing plants and retrofitting legacy inverters with advanced controls are essential to meet modern technical requirements. Countries like Australia and Germany have led this transition by enforcing stricter inverter standards and implementing retrofit programs, providing valuable references for evolving frameworks in regions like Colombia.

After ensuring stability with transient safety measures, the system is partitioned by load significance and power quality needs. Local agents connect only within the same tier, requiring distinct classifications. Each DER is managed by a dedicated master agent that handles requests and regulates power for DERs with the same matrix code [13]. This reduces communication links and costs while enabling autonomous zone management through decentralized master agents.

Agents operate in a matrix array, communicating via rows (agent type), columns (location and insertion order), and depth (controller-driven variability). Column-based interactions require pre-validation to ensure compliance with coordination IDs and power quality under national standards. Optimization focuses on low-power operations, minimizing control wear through rapid impedance transitions in bus bar coupling. Cross-zone communication occurs via the master agent of the requesting zone.

In the feedback stage, mutual ID vectors couple with the ordering matrix to track transitions per electrical cycle [14]. This creates a hash-based assurance search engine

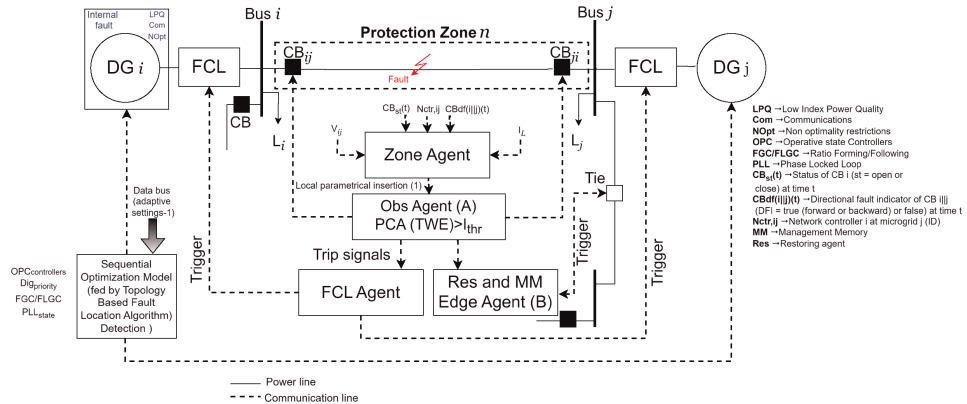


Figure 4. Operation of proposed protection scheme for power restoration (Resilient Int/Ext Failure) (ID) [6].

for operator traceability. Neighboring master agents share ID tables for DER coupling, ensuring updates. New zones announce their connection and share ID tables with neighboring master agents, maintaining a cohesive system.

2.4 Fault-tolerant network adaptation

The algorithm alternative route selection in distributed networks is designed to dynamically adapt the topology of a distributed network in the event of node failures. It identifies the direct neighbors of a failed node, removes the failed node from the graph, and evaluates potential replacement nodes based on their load and sensitivity. Sensitivity is calculated as an inverse function of node degree, ensuring that less congested and structurally optimal nodes are prioritized for re-routing. Parallelized evaluations by local agents reduce computation time and enable real-time responsiveness. The selected node is then integrated into the network by creating a new link with the neighbors of the failed node, ensuring minimal disruption to the system's overall operation. This approach is particularly relevant for power grids and communication networks, where fault tolerance and dynamic adaptability are critical.

The impact of topology based fault location detection-CV multistage and faulted section isolation on fault-tolerant network adaptation is enhanced through their implementation in agents within the power system. The first algorithm, topology based fault location detection-CV multistage, functions as an accumulator at the bus level, collecting and processing data from various field devices, protection relays, and system stability monitors. These agents are responsible for detecting faults by analyzing the network's topology and its transition states. When a fault is detected, the agent identifies the location of the faulted section by applying topological methods and homology persistence, considering the statuses of protection devices, switches, and other critical components. The agent then adapts the network by following the demand of the next microgrid in the serial cluster, ensuring that fault detection and isolation are carried out efficiently, maintaining system stability.

Algorithm 1. Alternative route selection in distributed networks.

Require: Graph $G = (V, E)$, Failed node S_f

Ensure: Updated graph G

- 1: Identify the direct neighbors N_f of S_f in the graph
 - 2: Remove S_f from the graph: $V \leftarrow V \setminus \{S_f\}$, $E \leftarrow E \setminus \{(S_f, v) | v \in N_f\}$
 - 3: Find candidate nodes C : nodes in V that do not belong to N_f and are not S_f
 - 4: Evaluate nodes in C based on load and sensitivity
 - (1) Compute load $\text{load}(v) = \sum_{u \in \text{neighbors}(v)} w(u, v)$
 - (2) Compute sensitivity $\text{sensitivity}(v) = \frac{1}{\text{deg}(v)+1}$
 - 5: Parallelize evaluation for all nodes in C using local agents
 - 6: Select the node $S_{\text{best}} \in C$ with the minimum load and maximum sensitivity
 - 7: Create a new link $(S_{\text{best}}, N_{f1})$, where N_{f1} is the first neighbor of S_f
 - 8: **return** G
-

Faulted section isolation is handled locally by agents that manage the isolation of faulted sections within the microgrid. They monitor circuit breakers and switches,

issuing open commands to isolate affected areas during faults. These agents adjust the power output of distributed energy resources (DERs), ensure proper switch operation, and prevent fault propagation to maintain system stability. Additionally, they validate the operability of isolation systems, addressing failures promptly to stabilize the network. These monitoring nodes (signal trains) are integrated through a Pareto front within the resource optimization framework, considering the number of nodes (orchestrator and agents) and coordinating the transition during transient periods (ensuring correct control mode and compliance with inertia zones).

Ultimately, the previous algorithms for transient detection, when integrated with system observability, necessitate an architecture grounded in edge computing. This approach enhances the responsiveness and efficiency of protection schemes by enabling localized processing and decision-making. The subsequent section will illustrate the deployment of this architecture within a service-based computing (SBC) framework, further emphasizing its pivotal role in advancing microgrid management.

The integration of renewable energy sources and their diverse control strategies introduces dynamic shifts in the static security regions of power systems, significantly impacting protection schemes and reserve configurations. To address these challenges, advanced algorithms are proposed to ensure optimal PMU transitions during faults under a hierarchical communication standard. This approach enhances information reliability for control centers, aligning with Northern Europe's sectionalized system methodology, which emphasizes distributed agent-based management.

Algorithm 2. Topology based fault location detection-CV multistage.

Input: Measurement and statuses of field devices, information from protective IEDs, DFIs from switches, state of latent space of energetic Wavelet and system stability monitoring (Resilient Self-Healing)(SQL)

Output: Location of the faulted section in microgrid (Modular internal fault in the converter, power filter coupling or external fault in the feeder)

STATUS: Field devices ($TWE_{(j,l)}$, $TWE_{(j,l)}^{Pmax}$, TWE_{thr} , $SWc(j,t)$, $SWv(j,t)$, $SW_{st}(j,t)$, $CBc(j,t)$, $CBv(j,t)$, $CB_{st}(j,t)$, $CBdfi(j,t)$) $\rightarrow N_{ctr,jm}$; Switching conditions manager ($p, o \in \text{list}(\text{DMS}, \text{FRT}, \text{FCM})$) $\rightarrow Fcon/com_{prev_state}$; HIL conditions ($Dig_{priority}$, $Bus_{Local_parametric_insertion}$, $OPC_{controllers}$) $\rightarrow \text{Changes_process_container}()$.

$N = \text{len}(\text{M_ID.row})$

$m \rightarrow$ Number of agents

$fc \rightarrow$ Forecast of generative NN (follow up $\chi(\text{FCM register, MATLAB event framework, test (Mutli-CE), optimizer(ADAM), hyperparameters validation (RandomSearchCV)}$)

Previous_state = M_ID(Class) $\rightarrow$ Zone(STATUS(p, o)), $t^- \in \text{Map of fault transitions}$

while $Dig_{priority} = \text{true}$ **Projection_latent_classifier**($TWE_{(j,l)}^{Pmax}$, TWE_{thr}) = Fault **do**

if IEDtr(j, t) = true **then**

 Calculate topology-based fault location detection, $N_{ctr,j} \rightarrow$

 HomologyPersistent(Victoris.rips, n.ring.follow

 = $N_{transitories_training}$, SW_{set} , CB_{set})

 Calculate information failure trip in HIL, FRT_Zone(STATUS-UDM models)

end if

if $CB_j = r'' \#j_False_External$ in Previous_state **then** $\triangleright$ Contrast (Manager.list, field)

```

For 1: N do
     $D_{pre} = \text{get}(\text{before the event, STATUS}(t^-))$ 
     $D_{post} = \text{get}(\text{after the event, STATUS}(t^+))$ 
EndFor
SQL_Ext = SQL manager extraction ( $F_{con}/com_{prev\_state}$ )
Check DFIs ( $D_{pre}, D_{post}$ )
if [DFIs.STATUS(Direct) ( $D_{pre} = D_{post}$ ) Stability( $Bus_{Local\_param\_insertion} = true$ )
OPC.Dll= $f_c^{gen}$ ] then
    Fault at the next section of the last DFI
     $OPC_{controllers}(t^+).list = \text{Digital map}(m, \text{Controlled external harmonic}$ 
insertion)
    else if DFIs.STATUS(Direct) ( $D_{pre} \neq D_{post}$ ) OPC.Dll= $f_c^{gen}$  then
        Fault at the pointing section
        execute (SIPS conceptual design)-Class Python
    else if OPC.Dll $\neq f_c^{gen}$  Stability( $Bus_{Local\_param\_insertion} = false$ ) then
        Failure in (IEC 61850 metrics, Tuning-dynamic detector, Internal board
failure)  $\rightarrow$  DMS virtualized
        Execute ( $Changes\_process\_container(guidance\_agents())$ )
    end if
end if
end while.

```

Algorithm 3. Faulted section isolation.

Input: Measurement and statuses of field devices, and status feedback from the switches and DERs

Output: Isolation of faulted section within microgrid

STATUS: $SW_c(j, t), SW_{st}(j, t), CB_c(j, t), CB_{st}(j, t), TM_{st}(j, t) \rightarrow N_{ctr,jm}, M_j$

Check the status of the target CB, switches $SW_{st}(j, t), CB_{st}(j, t)$

if $SW_{st}(j, t)=\text{close} \ \& \ CB_{st}(j, t)=\text{close}$ **then**

 Send open command to SW_j and CB_j

 Change $TM_{st}(j, t) = true$

For 1:j **do**

$M_j = \alpha * M_j + (1 - \alpha) * M_{j-1}, \alpha \rightarrow$

$sign\left(\sum_j (TM_{st}(j+1) * Framework(DER_c, Load_{pq}(j)))\right)$

$TM_{st}(j+1) = TM_{st}^m(j) * e^{|V_{highF-injection}|}$

EndFor

 Validation of switchover operability in manager.

end if

if DER exists within the faulted section $\vee SW_{st}(j, t) = \text{close}$ **then**

 Send open command to SW_j

else

 Reduce power order of DER j to min. Set $DER_{power}(j, t) = \text{minimum}$

Check API (physical/virtual) (register_movements.sdf = $SW_c(j, t)$)

 Switch.oti = $CB_c(j, t)$

end if

3. Kubernetes architecture for energy management: Agent-based flexibility systems

As the demand for efficient energy management systems grows, the integration of advanced technologies becomes essential for optimizing microgrid operations. This section delves into the architecture of Kubernetes for energy management, highlighting its role in facilitating agent-based flexibility systems. By leveraging Kubernetes, the proposed framework aims to enhance scalability, resilience, and real-time responsiveness in managing complex energy systems, ultimately driving improvements in operational efficiency and system stability.

The increasing complexity of modern energy systems necessitates robust and scalable architectures to manage flexibility effectively. Kubernetes, as a container orchestration platform, provides a powerful foundation for deploying and managing agent-based systems within energy management environments. By leveraging a cluster-based architecture, energy systems can achieve heightened levels of automation, adaptability, and fault tolerance. This capability enables the efficient deployment of agents tasked with critical operations such as demand response, resource optimization, and real-time grid balancing, ensuring the system remains flexible and responsive to dynamic conditions.

In the evolving paradigm of flexible power system operations, three core aspects demand attention: observability, response speed, and failure or downtime rates. These factors are pivotal in designing an optimal architecture tailored to the incremental rate of the Independent System Operator (ISO). This incremental rate, alongside the corresponding data pools and databases, drives the need for increasingly granular and scalable architectures. With the progressive proliferation of low-voltage systems, such as microgrids, the necessity for hybrid managers based on distributed system operators (DSOs) has become evident. These developments underscore the growing volumes of information and the escalating functionality requirements of Energy Management Systems (EMS).

Traditional monolithic architectures can no longer meet the low coupling and high-reliability demands of modern EMSs. Gartner's proposed architecture from 1996 introduced a new operational paradigm by integrating functionalities such as five-minute forecasting and short-term hourly offers, emphasizing grid security, reliability, and resilience. Current control centers predominantly utilize web-based service applications to manage EMS functionalities. Service-Oriented Architectures (SOA) offer ease of replacement and high reliability, making them a mainstream choice for software development. SOA has thus become a standard approach in the implementation of modern EMSs.

However, SOA-based energy management systems (S-EMS) face several limitations. First, the coarse-grained decoupling in SOA results in a lack of fine-grained fault isolation. Service failures within an application can propagate, causing the application to fail entirely. This issue complicates fault localization, requiring prolonged system downtimes and jeopardizing the safety and stability of the power system. Second, the absence of robust application isolation increases the risk of cascading failures triggered by individual application malfunctions. Third, a sudden surge in system access can lead to performance degradation due to the enterprise service bus (ESB). Failures or performance bottlenecks in the ESB compromise the robustness of the entire system.

In response to these limitations, Kubernetes offers a microservices-based architecture that addresses the drawbacks of traditional SOA. By employing fine-grained

decoupling, Kubernetes-based EMS architectures ensure better fault isolation and resilience. Each microservice operates independently, allowing the system to remain operational even during localized failures. Furthermore, Kubernetes' ability to dynamically scale resources enhances system robustness during high-demand scenarios, ensuring uninterrupted service. This shift towards a Kubernetes-based architecture represents a transformative advancement in the design and deployment of flexible and resilient EMSs, enabling them to meet the evolving demands of modern energy systems effectively.

Complementing Kubernetes, a physical deployment based on intelligent electronic devices (IEDs) and single board computers (SBCs) is proposed to establish a foundational layer at the local level. The IEC 61850 standard provides the necessary framework for the physical layer and the implementation of protection and control functionalities. This standard ensures interoperability and reliability in the communication and coordination of devices within the power system. On the other hand, a RESTful API serves as a lightweight communication method for transmitting global indicators, enabling efficient data exchange between local and central systems. This dual-layered approach enhances both localized control and global coordination, offering a robust and scalable solution for modern EMS architectures.

- Lighter communication method: RESTful realizes the management of and access to resources through a URI, which has stronger scalability and a clearer system structure. Communication between services is carried out by accessing the specific API of the target service, which breaks the performance bottleneck of the ESB and has high reliability.
- Excellent application isolation: The K-EMS decouples EMS applications into services and deploys them in containers. Services are effectively isolated, and cascading failure is avoided.
- Upgraded expansion method: The system expansion method is refined from the system to the application, and the system can achieve higher reliability under limited resources.

To validate the communication system's reliability and synchronization under real-world conditions, physical tests were conducted using a Raspberry Pi-based setup. These tests focused on offset calibration between local agents and higher-level controllers, ensuring precise timing and seamless coordination in distributed tasks. The results confirmed the system's ability to dynamically adjust offsets and maintain synchronization across hierarchical layers, even under variable operating conditions. **Figure 5** illustrates the physical test setup and the offset calibration process, emphasizing the accuracy achieved during these trials.

Figure 6 illustrates the information flow dynamics between hierarchical services implementing master transition through zonification, with particular emphasis on scenarios involving high memory consumption and critical importance cases as zone masters. The diagram presents three distinct states of the distributed network: the original configuration with full connectivity (**Figure 6a**), followed by two post-failure scenarios showcasing the implementation of Algorithm 1 Alternative Route Selection in Distributed Networks. This algorithm facilitates dynamic master reassignment based on local network metrics and zonification parameters, ensuring service continuity through intelligent path recalculation.

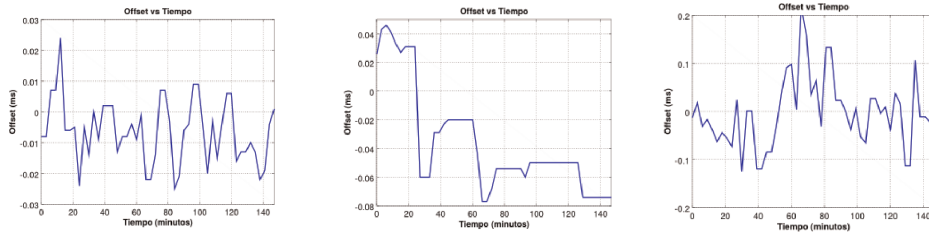


Figure 5.
Offset detection for minimum trigger level adjustment to local sequences.

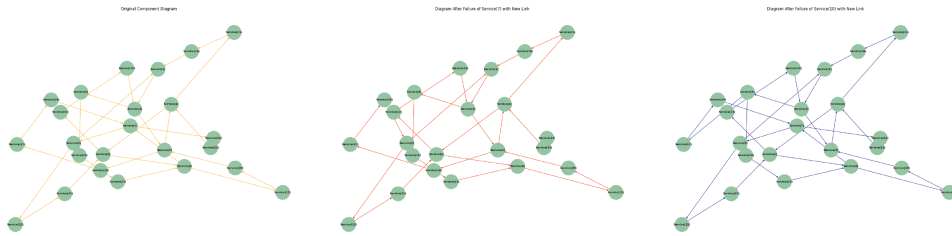


Figure 6.
Information Flow Between Hierarchical Services with Master Transition Through Zonification, focusing on high memory consumption and critical importance cases as zone masters: (a) original state of distributed agent processes, (b) decision-making path after service failure in 7, and (c) decision-making path after service failure in 10.

When analyzing the transition states after service failures in nodes 7 and 10 (**Figure 6b** and **6c** respectively), we observe how the algorithm orchestrates master transitions by evaluating local performance metrics within each zone. The network's adaptive response is evident in the modified connection patterns - represented by red paths in the post-node 7 failure state and blue paths in the post-node 10 failure state - demonstrating how the algorithm optimizes the network topology while maintaining the hierarchical service structure. This implementation shows particular efficiency in handling high memory consumption scenarios, where master selection decisions are critical for maintaining optimal service distribution across the zonified architecture.

3.1 Resource allocation in distributed process management systems

This structure is disruptive compared to conventional methods for identifying natural constraints. By integrating the communication topology with the power system, uncertainties related to demand variations and topological changes are significantly reduced. This approach narrows the analysis framework to short-term events, shifting the maintenance perspective towards dynamic disconnection management. Such a shift not only requires regulatory improvements but also facilitates flexibility markets, effectively eliminating the need to explore incremental pessimistic scenarios.

This technological variation necessitates the subdivision of operational areas into smaller compact elements, where DSOs (distribution system operators) are tasked with managing system outputs and making critical control decisions. This subdivision aims to address challenges in grid robustness and stability, ensuring that localized issues can be managed efficiently while maintaining overall system performance.

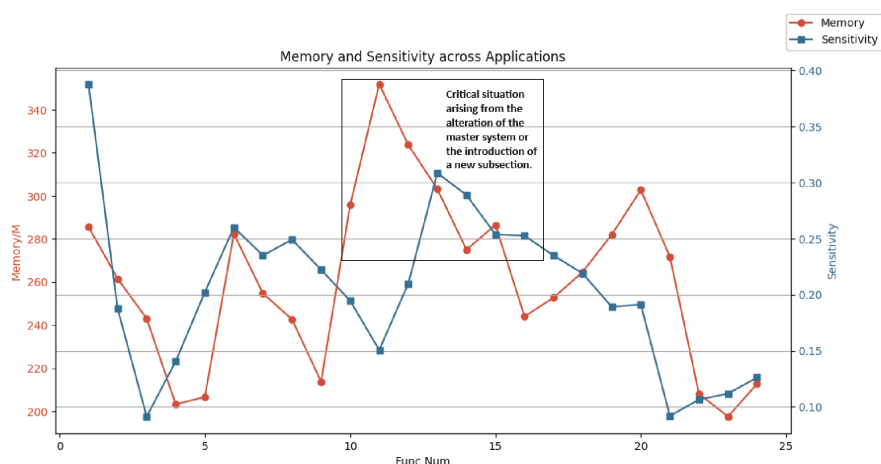


Figure 7.
 Evaluation of resource depletion sensitivity in relation to function scalability, considering the involvement of resilient agents (12–24).

This technological variation necessitates the subdivision of operational areas into smaller, compact elements, where DSOs (distribution system operators) are tasked with managing system outputs and making critical control decisions. This subdivision aims to address challenges in grid robustness and stability, ensuring that localized issues can be managed efficiently while maintaining overall system performance. As demonstrated in **Figure 7**, the evaluation of resource depletion sensitivity in relation to function scalability reveals a significant pattern: every twelve processes correspond to a microgrid process that exhibits decremental behavior. This observation is particularly evident in the memory and sensitivity curves (12–24), where the resource utilization pattern shows distinct cyclical variations, suggesting an inherent relationship between process allocation and system efficiency in managing distributed resources.

Furthermore, under a system-based control (SBC) framework, physical constraints inevitably emerge. These include limitations related to memory capacity, response times, and communication distances between agents. If this data structure were integrated into an optimization problem involving power flow analysis, an incremental variant of constraints at the n-gram level would appear. Therefore, it is recommended that the communication structure be expanded significantly. Such scaling would allow tasks managed by agents to be progressively subdivided with greater precision, rendering the Pareto front unnecessary, as system supportability would be overestimated.

3.2 Structural hierarchy for service functions in intelligent microgrids

Figures 8 and 9 illustrate a hierarchical multi-agent system using hybrid control, organized into upper, middle, and lower levels. The upper level handles global decision-making with optimization strategies and energy management, supported by Kubernetes containers for scalable agent deployment and cloud databases for centralized knowledge management.

The process bus links the physical layer—comprising sensors, actuators, and field devices—to agents deployed in Kubernetes containers. These agents, especially at the

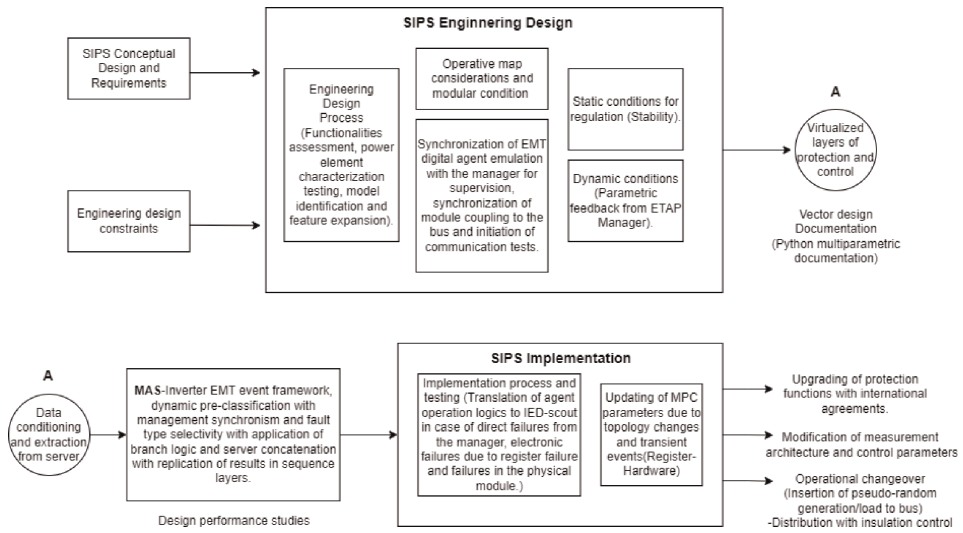


Figure 8.
SIPS diagram (information system for planning and supervision).

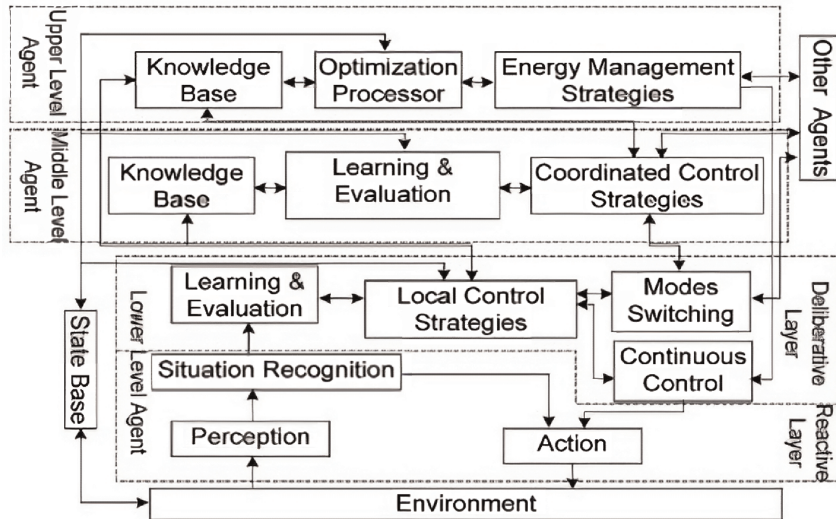


Figure 9.
Architecture of a hierarchical multi-agent system based on hybrid control [15].

lower level, perform local control, event recognition, and perception, leveraging Kubernetes for low-latency processing and rapid response to environmental changes.

At the middle level, agents coordinate strategies between the upper and lower levels, ensuring optimized models are validated and adjusted for grid dynamics. Kubernetes enables scalable container orchestration for efficient evaluation, learning, and communication across levels, while cloud databases manage information flow with redundancy and failover capabilities.

Knowledge management integrates with system infrastructure through a cloud-hosted repository for control strategies and system integrity protection scheme (SIPS)

results. Kubernetes facilitates modular agent implementation for protection logic and fast grid response, redefining traditional approaches by virtualizing protection and control layers.

Integration studies ensure grid stability during transitions using pseudo-random generation and operational strategies. Kubernetes dynamically scales agents based on computational needs, optimizing resources for system demands.

Finally, the process bus ensures robust communication between the physical layer and agents, enabling synchronized operation and dynamic real-time control. This architecture, managed by Kubernetes, provides a scalable, resilient platform for systematic security studies and efficient grid operation.

4. Conclusions

The integration of cutting-edge technologies, such as adaptive multi-agent systems (MAS), edge computing, and hierarchical control, has significantly advanced microgrid protection. These methodologies enhance fault tolerance, operational efficiency, and real-time decision-making by enabling localized control through dedicated agents assigned to intelligent electronic devices (IEDs). Edge computing, in particular, minimizes latency, improves fault clearance times, and ensures system resilience by allowing autonomous operation even during communication disruptions.

4.1 Future recommendations

Future work should focus on finding the Pareto optimal point between the electrical proximity model with state estimation and the unit-based resource allocation model. This approach will balance the trade-offs between accuracy, computational efficiency, and resource utilization, enabling more precise fault detection, load balancing, and grid management. By integrating this optimization into edge computing frameworks, microgrids can achieve enhanced responsiveness and reliability, reducing dependency on traditional state estimators and improving real-world operational performance.

Acknowledgements

This research was funded by Minciencias and Universidad del Valle under the project “Estrategias para el desarrollo de sistemas energéticos sostenibles, confiables, eficientes y accesibles para el futuro de Colombia” (Minciencias Code: 1150-852-70378, Hermes Code: 46771).

Notes

Note from author: The structural basis of the architecture can be found in article [6] which was developed with the Colombian Ministry of Science and tested through a laboratory, in addition to international consultancy for the implementation at the microgrid level. In case of any doubts regarding the types of tests, safety conditions, cascade simulations, and master reconnection, please refer to the article. Additionally,

for inquiries about real-time setup, wavelet mother criteria definition, or creation of virtualized IEDs, contact the author at andres.diaz.caicedo@correounivalle.edu.co.

Thanks

I would like to express my profound gratitude to my family for their unwavering support throughout my academic journey, which has been a constant source of strength and motivation. I am also deeply thankful to the directors of my research group for their trust and mentorship, enabling me to lead the development of new algorithms in microgrids. Additionally, I acknowledge the University of Valle for providing an exceptional intellectual environment and essential resources that have been instrumental in advancing my research.

Abbreviations

ADAM	algorithm for optimization used in neural networks.
AN	agent node.
API	application programming interface.
ARM	advanced RISC machine (processor architecture).
BAT	battery.
CE	control event or efficiency condition.
CIP	critical infrastructure protection.
CLUSTER	group of interconnected computing nodes.
CV	cross-validation.
DE	data event.
DER	distributed energy resources.
DFI	distributed fault indicator.
DMS	distribution management system.
DSO	distribution system operator.
DYN	dynamic processes or states.
EMS	energy management system.
ESB	enterprise service bus.
ETAP	electrical transient analyzer program.
FCM	fault current management.
FRT	fault ride through (grid fault tolerance).
GOOSE	generic object-oriented substation event (IEC 61850).
HIL	hardware-in-the-loop.
ICDE	international conference on data engineering.
IED	intelligent electronic device.
INT	interaction model.
ISO	independent system operator.
JIT	just-in-time compiler.
MAS	multi-agent systems.
MCC	microgrid central controller.
MQTT	message queuing telemetry transport (protocol).
NERC	North American electric reliability corporation.
OPC	open platform communications (industrial protocol).
PEDG	power electronics for distributed generation.

PV	photovoltaic system.
SBC	single board computer.
SCADA	supervisory control and data acquisition.
SCL	substation configuration language.
SIGPL	special interest group on programming languages.
SIL	Software-in-the-loop.
SIPS	system integrity protection scheme.
SMV	sampled measured values (IEC 61850).
SNR	signal-to-noise ratio.
SOA	service-oriented architecture.
TSN	time-sensitive networking.
UDM	universal dynamic models.
URI	uniform resource identifier.
VLDB	very large data bases.

Author details

Andres Mauricio Diaz Caicedo^{1,2}

1 International Group for Research in Control (GICI), Electrical and Electronic Engineering Department, Universidad del Valle, Cali, Colombia

2 Group for Research in Advanced High-Voltage Technologies and Applications (GRALTA), Electrical and Electronic Engineering Department, Universidad del Valle, Cali, Colombia

*Address all correspondence to: andres.diaz.caicedo@correounivalle.edu.co

IntechOpen

© 71 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Cuadrado N, Gutierrez R, Zhu Y, Takac M. Mahtm: A Multi-Agent Framework for Hierarchical Transactive Microgrids. arXiv; 2023
- [2] Zhang X, Zhong Q-C, Ming W-L. Adaptive series-virtual-impedance control strategy for load converters to improve the stability of the cascaded system. In: 2016 IEEE 7th International Symposium on Power Electronics for Distributed Generation Systems (PEDG). Piscataway, NJ, USA: IEEE; 2016. pp. 1-5
- [3] Liu C, Jiang B, Patton RJ, Zhang K. Hierarchical-structure-based fault estimation and fault-tolerant control for multiagent systems. IEEE Transactions on Control of Network Systems. 2019; **6**(2):586-597
- [4] Fkaier S, Khargui M, Frey G. Meta-model for control applications of microgrids. In: 2020 6th IEEE International Energy Conference (ENERGYCon). Piscataway, NJ, USA: IEEE; 2020. pp. 945-950
- [5] Shobole AA, Wadi M. Multiagent systems application for the smart grid protection. Renewable and Sustainable Energy Reviews. 2021;**149**:111352
- [6] Diaz Caicedo AM, Mejía ÉF, Gómez-Luna E. Revolutionizing protection dynamics in microgrids: Local validation environment and a novel global management control through multi-agent systems. Computers and Electrical Engineering. 2024;**120**:109748
- [7] Alasali F, Hayajneh AM, Ghalyon SA, El-Naily N, AlMajali A, Itradat A, et al. Enhancing resilience of advanced power protection systems in smart grids against cyber-physical threats. IET Renewable Power Generation. 2024;**18**(5):837-862
- [8] Sahu A, Wlazlo P, Mao Z, Huang H, Goulart A, Davis K, Zonouz S. Design and Evaluation of a Cyber-Physical Resilient Power System Testbed. arXiv; 2020
- [9] Abbaspour E, Fani B, Sadeghkhani I, Alhelou HH. Multi-agent system-based hierarchical protection scheme for distribution networks with high penetration of electronically-coupled dgs. IEEE Access. 2021;**9**:102998-103018
- [10] Taveras-Cruz AJ, Mariano-Hernández D, Jiménez-Matos E, Aybar-Mejia M, Mendoza-Araya PA, Molina-Garca A. Adaptive protection based on multi-agent systems for ac microgrids: A review. Applied Energy. 2025;**377**:124673
- [11] Waraphok P, Saengsuwan T. Database development for power quality in pea's distribution system. In: 2007 9th International Conference on Electrical Power Quality and Utilisation (EPQU). Piscataway, NJ, USA: IEEE; 2007. pp. 1-6
- [12] Galantino S, Risso F, Cazzaniga A, Garrone F, Terruggia R, Lazzari R. An edge-based architecture for phasor measurements in smart grids. In: 2022 AEIT International Annual Conference. Piscataway, NJ, USA: IEEE; 2022. pp. 1-6
- [13] Mahmoudian A, Garmabdari R, Bai F. Adaptive power-sharing strategy in hybrid ac/dc microgrid for enhancing voltage and frequency regulation. International Journal of Electrical Power and Energy Systems. 2024;**156**:109696
- [14] Duan J, Wang C, Hao X, Liu W, Xue Y, Peng J-c, et al. Distributed control of inverter-interfaced microgrids based on consensus algorithm with improved transient performance. IEEE Transactions on Smart Grid. 2019;**10**(2):1303-1312
- [15] Dou C-X, Liu B. Multi-agent based hierarchical hybrid control for smart microgrid. IEEE Transactions on Smart Grid. 2013;**4**(2):771-778

A Multi-Agent Infrastructure Monitoring IoT System for Urban Geo-Resilience Framework

Hasan Tariq, Farid Touati, Mohammed Abdulla E. Al-Hitmi, Damiano Crescini and Adel Ben Mnaouer

Abstract

Multi-agent systems (MAS) have emerged as a cornerstone technology in developing agile and effective resilience frameworks for geo-distributed ecosystems. These systems play a pivotal role in addressing real-time challenges in Structural Health Monitoring (SHM), where millions of distributed data-processing tasks must be executed swiftly/parallel to ensure system resilience. Core functionalities in this context include resolving complex sustainability conflicts, detecting and characterizing anomalies in chaotic datasets, and formulating adaptive and unique resilience strategies. This work presents a multi-agent SHM system through the development of a Geo-Resilience Support System (GRSS) based on an enhanced synthetic DADO production machine architecture. The proposed multi-agent system architecture optimized the distributed data handling, rule-based decision-making, and conflict resolution by combining the advantages of GATOR, LEAPS, TREAT and RETE networks. The architecture greatly improved SHM capabilities for regional-scale resilience by utilizing agents to effectively manage working memory (WM) and apply conflict resolution methods (CRS) in alpha and beta rule networks. The system can now dynamically align left-hand-side (LHS) conditions with right-hand-side (RHS) solution sets thanks to important developments including the enrichment of condition-action rule sets and the optimization of solution strategies. With a packet throughput of 21 packets/second across geographic structures and a round-trip response time of 76.2 seconds for the initial event, the system's usefulness is demonstrated. This work underscores the transformative potential of multi-agent SHM systems in delivering robust, scalable, and distributed solutions for geo-distributed ecosystems, enabling real-time resilience and sustainability in diverse infrastructural contexts.

Keywords: multi-agent systems, decision support systems, structural health monitoring, IoT, resilience

1. Introduction

In the context of multi-agent systems (MAS), global sustainability and regional resilience can be approached through a distributed network of intelligent agents working collaboratively to address complex challenges [1]. These agents, representing various urban actors and systems, can coordinate to optimize trade-offs between development and risk, forming a dynamic Urban Resilience System (URS) [2]. Multi-agent technologies (MAT) and ontologies can be leveraged to create a framework for resilient cities, allowing for adaptive responses to extreme events and changing conditions [3]. By integrating artificial intelligence, standardized geospatial sensing capabilities, and real-time data analysis, multi-agent systems can enhance urban-scale resilience systems (URS) and support the development of resilience maturity models (RMM) [4]. These systems can simulate and manage various aspects of urban resilience, including emergency response, infrastructure safety, and environmental sustainability, providing a comprehensive approach to urban resilience against natural and human-induced calamities. Thus, MASs offer a promising approach to structural health monitoring (SHM) by enabling distributed, collaborative, and adaptive monitoring of complex structures and can be implemented to coordinate various sensors, process data in real-time, and make decisions based on collective intelligence [5]. This approach allows for more efficient data gathering, interpretation, and analysis compared to traditional centralized methods [6]. Multi-agent SHM systems can integrate diverse technologies such as wireless sensor networks, building information modeling (BIM), finite element analysis (FEA), and artificial intelligence to create a comprehensive monitoring solution [7]. These systems can perform tasks such as damage detection, data compression, alarm generation, and long-term data storage while adapting to changing structural conditions [8]. By leveraging the distributed nature of MASs, SHM can achieve improved scalability, robustness, and real-time performance in monitoring large and complex structures.

In the realm of multi-agent systems, recent years have witnessed a diverse array of approaches aimed at enhancing sustainability and resilience automation [9]. These approaches encompass stochastic methods for risk assessment, vulnerability, robustness, and resilience from a decision-theoretic perspective, which require improvements in practical implementation constraints. Furthermore, existing characterization and dispute resolution techniques and procedures must complement resilience methodologies and sustainability trends and visions in key infrastructures [10]. Utilizing the SVIDT technique, the robustness of connected human-digital environment systems necessitates intelligent architecture and the inclusion of data at the urban scale, which is what this work addresses.

The proposed framework emphasizes the need for matching algorithms for event classification and handling. According to existing research, TREAT just requires an α network, whereas AI matching algorithms like RETE require a hierarchical α and β network for solution space and greater working memory [11]. Although it is more appropriate, the LEAPS method [12], which combines the advantages of TREAT and RETE, introduces gaps at the solution side (RHS) and left-hand-side state-level sample spaces (LHS), resulting in sizable conflict sets inside the rules space [13]. Moreover, the semantic data processing approach using approximate string matching requires enhancements to support resilience models with minimal variables, necessitating a robust parallel computing architecture like DADO [14, 15].

The multi-agent SHM framework scalable infrastructure developed in previous works [16, 17] needs integration with heterogeneous sensing mechanisms for safety

assessments over urban-scale distances along with optimized data flow from sensors to cloud systems. Furthermore, the Resilience Support System (RSS) created using dynamic programming (DP) in DADO machines must evolve into a software-defined network (SDN) [18, 19] to resolve conflict sets (CS) effectively using the GATOR algorithm as the most cost-efficient option [20]. Network function virtualization (NFV) has also been incorporated into the proposed RSS for urban-scale sustainability. This work addresses the observed gap in physical and real application-centered Urban Resilience System (URS) implementations with multi-cluster model optimization over sensor data found in earlier studies [21, 22], proposing specific improvements within the limitations of the Delta project. Ultimately, this work presents a Resilience Support System Architecture (RSSA) implemented using a novel multi-agent SHM DADO machine (MA-SDM) system prototype by considering key recommendations from the review work [23] and capitalizing the geospatial filtering algorithm from our work [24].

The reviewed studies on IoT architectures and urban infrastructure monitoring revealed several research gaps that have limited the effectiveness of existing frameworks. These gaps underscored the need for a Multi-Agent Infrastructure Monitoring IoT System for Urban Geo-Resilience Framework as a comprehensive solution. One of the key limitations observed was the lack of a unified and scalable IoT architecture tailored to urban geo-resilience. While Ray [25] and Sethi and Sarangi [26] provided extensive surveys on IoT architectures, their works primarily focused on broad classifications without addressing the specific requirements of resilient urban infrastructure. Similarly, Domínguez-Bolaño et al. [27] reviewed IoT technologies and open-source projects but did not propose an integrative framework suited for multi-agent collaboration in geo-resilience. Another significant gap was identified in real-time environmental and infrastructure health monitoring. Tariq et al. [28] developed an IoT-based air quality mapping system, but it primarily focused on air pollutants without considering the structural and environmental resilience of urban infrastructures. Likewise, Tariq et al. [29] proposed a cadastral geospatial IoT network but lacked a multi-agent framework capable of adaptive decision-making based on real-time data. A critical challenge in standardization was also noted. Al-Qaseemi et al. [30] and Yaqoob et al. [31] highlighted the lack of interoperability among IoT systems, which hindered seamless data exchange between heterogeneous urban infrastructure monitoring devices. Tariq et al. [32] attempted to address this issue through a multi-protocol data network interface, but their work did not implement an agent-based approach to enhance autonomous system coordination. Moreover, while Ganchev et al. [33] proposed a generic IoT architecture for smart cities, their work did not incorporate resilience metrics essential for adaptive urban planning. The proposed architecture lacked a dynamic response mechanism for environmental hazards and structural vulnerabilities.

Given these gaps, the Multi-Agent Infrastructure Monitoring IoT System for Urban Geo-Resilience Framework is positioned as the optimal solution. This framework integrates real-time data acquisition, multi-agent collaboration, and adaptive analytics to enhance urban resilience. It addresses the limitations of previous studies by introducing a comprehensive monitoring system that ensures interoperability, real-time infrastructure health assessment, and geo-resilience adaptation. Through this approach, urban areas can achieve enhanced sustainability, proactive disaster response, and long-term infrastructural integrity.

2. Methodology

The proposed multi-agent infrastructure health monitoring IoT system integrates a distributed expert system employing belief-rule (BR) deductive logic to enhance urban geo-resilience. The methodology consists of two structured components and a synergic workflow of step-wise hierarchical integration of these key components:

2.1 Multi-agent decision support system architecture

The system comprises three blocks:

- *SHM-DSS interaction block (SD-IB)*: Interfaces with remote Structural Health Monitoring (SHM) systems using MySQL and SQLite for data management. SHM gateways process sensor data at various scales (structure, zone, city, country, and global levels). ThingSpeak supports public cloud analytics.
- *MA-DSS expert system block (MD-ESB)*: Includes a Query Block (data processing) and an Advice Block (decision display). It employs rule-based inference engines with RETE, TREAT, LEAPS, and GATOR algorithms for event analysis, matching conditions, and optimizing decision-making.
- *MA-DSS response block (MD-RB)*: Manages real-time responses *via* IoT GATOR blocks. These function as dynamic buffers, routing event-based decisions for population protection.

2.2 Multi-agent SHM synthetic DADO machine (MA-SDM)

MA-SDM extends MA-DSS with a parallel hardware-software system integrating AI and machine learning algorithms. It consists of:

2.2.1 Hardware components

- *Remote block*: Uses RPi3 OSB, CANopen sensors (MMA8451Q accelerometer, STM32F10RBT6 microcontroller) in grid topologies for seismic monitoring.
- *Site block*: A mesh network (RPi3, Wi-Fi/Bluetooth, Banana Pi R1 gateway) handling local and remote configurations.
- *Cloud block*: Hybrid personal and public cloud architecture using MySQL, MATLAB, LabView, Kaa, and EmonCMS for advanced analytics.

2.2.2 Software components

- *IoT blocks*: Multi-client, multi-gateway, and multi-server layers processing event data with algorithms ensuring real-time anomaly detection.
- *Processing stack*: LabView, MATLAB, SQLite3, and MySQL handle analytics, while Python-based RTL blocks manage decision workflows.

2.3 Integrated decision support and resilience framework

MA-SDM extends MA-DSS and integrates MA-ESB at multiple operational levels (site, gateway, and cloud). A Dell Xeon Hex-Core Server coordinates dashboards for the Geo-Resilience Support System (GRSS) with an integration chain as:

1. Data collection: Smart sensors gather real-time geotechnical, and structural data.
2. Multi-agent framework: Distributed agents analyze and communicate sensor data for real-time decision-making.
3. BR deductive logic: A knowledge-driven system processes uncertain data, improving risk assessment.
4. Geo-resilience support: Automated alerts and mitigation strategies enhance urban resilience.
5. Cloud processing: Data visualization and reporting enable proactive infrastructure management.

The final architecture enables scalable, efficient, and autonomous decision-making across multi-agent networks, ensuring robust disaster response and structural health monitoring.

3. Multi-agent decision support system architecture

A Multi-Agent Decision Support System (MA-DSS) can be conceptualized as a distributed expert system employing Belief-Rule (BR) deductive logic, specifically utilizing IF-THEN rules. Within this framework, the degrees of belief are determined by the intersection of confidence intervals for multiple events. Consider a multi-agent monitoring system with time vector t , a set V of n time-series variables, and a set A of anomalies. For each relationship between V and A , within a confidence interval I , there is a conflict support C and a resolution support R at time t . This rule model serves as the foundation for an isometric monitoring system's real-time BR expert system. Where multiple agents collaboratively process and interpret data, make decisions, and respond to anomalies in a coordinated manner.

$$Rn(t) = \begin{cases} \text{IF } V \text{ has } A \text{ in } n(t) = (0, 1, 2, \dots, t) \text{ and } a = (0, 1, 2, \dots, t) \\ \text{THEN } C \text{ has } R \text{ for } I \text{ in } n(t) = (0, 1, 2, \dots, t) \end{cases} \quad (1)$$

The equation provides the belief update for R and the degree of confidence in accurate event management β_k with a referential value i determines the value of I for R .

$$\beta n(t) = \beta n(t-1) \frac{I(t)}{I(t-1)} \text{ for } i = (0, 1, 2, \dots, t-1) \quad (2)$$

Each β_k has five severity levels, which are a single condition membership: LL stands for Low-Low, HH for High-High, HL for High-Low, M for Medium, and LH for Low-High. $I(t) > I(t-1)$ for the new $\beta_k(t)$ will replace the previous degree of trust or a greater confidence interval than the prior one. We must create a precise architecture based on our mathematical model that can handle large amounts of rule and belief data. The main goals of the RSS architecture are to maintain the utility computing system operating at peak efficiency while reducing the cost and duration of disaster recovery, as stated in Ref. [14]. Three levels make up the suggested RSS, and they cooperate to enable it to address event management issues. These layers are as follows:

1. SHM-DSS Interaction Block (SD-IB)
2. MA-DSS Expert System Block (MD-ESB)
3. MA-DSS Response Block (MD-RB)

3.1 SHM-DSS interaction block (SD-IB)

Data sources feed the expert system *via* SHM systems at remote locations, using MySQL clients in gateways and SQLite in nodes. The MA-DSS server processes this data using MySQL Server 5.5, while SHM gateways use MySQL Client 5.5. Results are stored in MySQL databases as remote site solution files presented in **Figure 1**. The system architecture in **Figure 1** has a minimum data rate requirement of 512 kbps and an initial input buffer with 36 attributes spread across four entities (Projects, SHM, Structures, and Sensors-Grid). This configuration represents a single site or structure within a city and forms the Left-Hand Side (LHS) of the system. The database tables

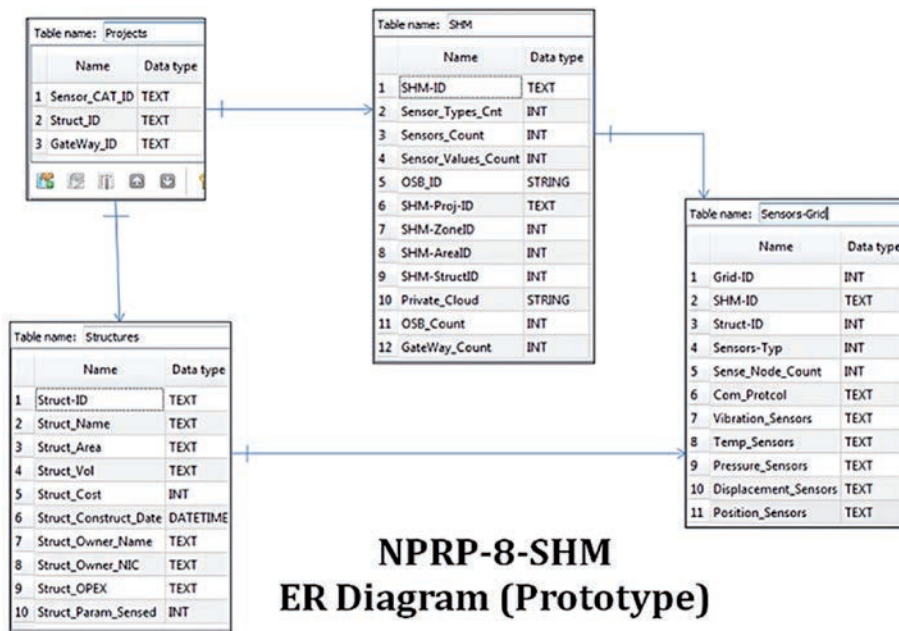


Figure 1.
ER diagram of SD-IB.

are designed to optimize the SHM-DSS operations, allowing clients to perform their tasks while reserving the server for critical functions.

Computational tasks are distributed across different levels: structure-level calculations are performed by the site block (**Figure 2**), zone-level by the gateway (**Figure 3**), city-level by the RSS private cloud level 1 (PRCL1), and country-level by private cloud level 2 (PRCL2). Global-scale information processing and forecasting are handled by external organizations like NASA and NOAA. The system also incorporates ThingSpeak, Mathworks' IoT analytics platform, as a public cloud (PBCL) component.

3.2 MA-DSS expert system block (MD-ESB)

The MD-ESB architecture, as illustrated in **Figure 4**, comprises two key components: the "Query Block" and the "Advice Block".

The Query Block manages sensor data manipulation and storage requests, while the Advice Block is responsible for displaying outputs, such as alarms and structural health status probabilities. The SHM inference engine operates on a rule-based program, containing a set of rules (R_0 to R_n) that correspond to various possible relationships within the problem database. These rules are linked to a set of actions or commands (S_0 to S_m) that update the existing database for a given $\beta_k(t)$, allowing the multi-agent system to process and respond to incoming data collaboratively. Each R is a rule equation that represents a unit relationship, which may include variables and constants. The SELECT command evaluates constants directly, while JOIN is used to evaluate variable relationships.

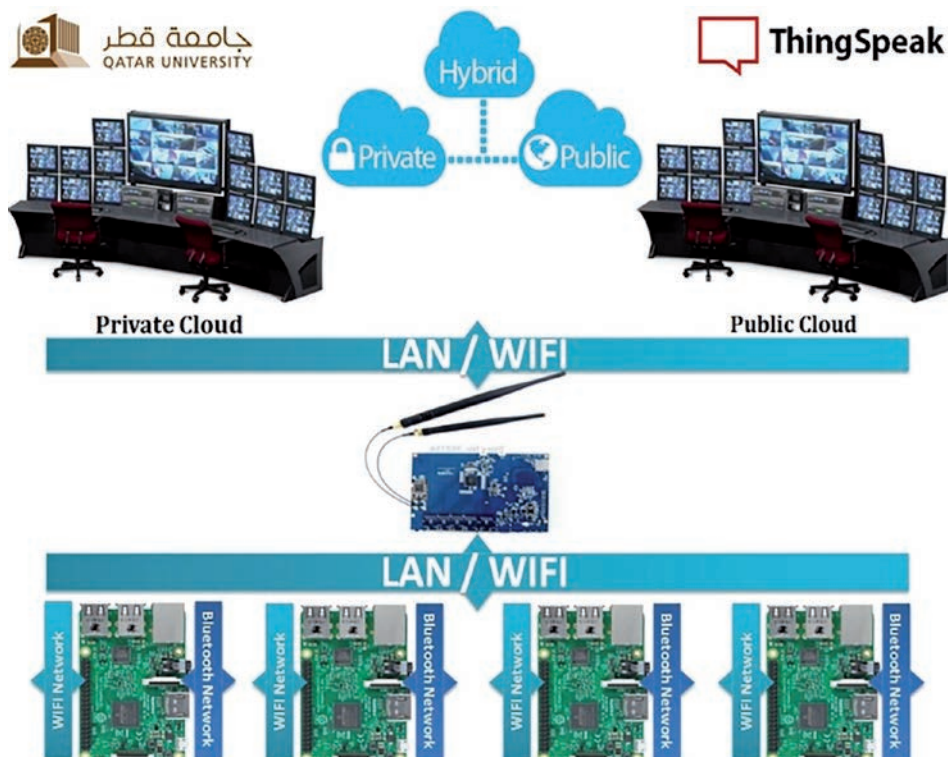


Figure 2.
 Overall clouds and network block for MA-SDM-CB.

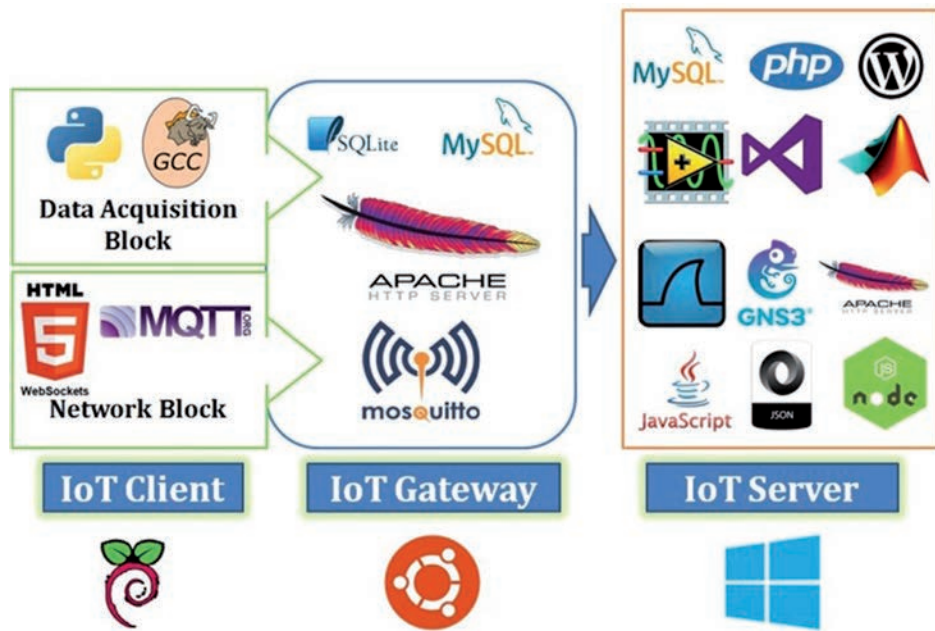


Figure 3.
MA-SDM software block architecture.

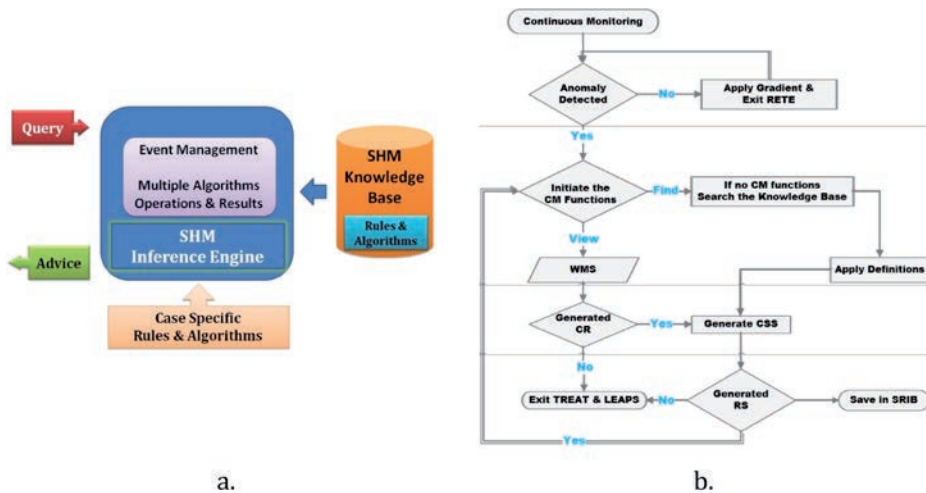


Figure 4.
MD-ESB integrated with Local RTL block. (a) MD-ESB block diagram. (b) MD-ESB local RTL block diagram.

One R is evaluated at a time by the inference engine (IE), which then performs cyclic workflow. Each R is a rule equation that represents a unit relationship, which may include variables and constants. The SELECT command evaluates constants directly, while JOIN is used to evaluate variable relationships. One R is evaluated at a time by the inference engine (IE), which then performs cyclic workflow. **Figure 4 (b)** introduces a gradient function based on the Newton Forward Difference method to track changes in sensor values over time. A significant incremental difference suggests a higher probability of

anomaly. The Real-Time Logic (RTL) block's first segment activates only when an anomaly is detected, or the gradient is increasing. If the Condition Monitoring (CM) functions lack appropriate rules, the system searches the knowledge base. The Working Memory Set (WMS) is dynamically accessed by TREAT and LEAPS algorithms using concurrent-read operations. The SHM inference engine, a rule-based system implemented in a high-level programming language, executes rules and performs data operations. For event management, data analysis, and decision-making, the system employs four AI algorithms: RETE, TREAT, LEAPS, and GATOR, all utilizing a rule-based approach. A matching algorithm's state information is evaluated using conditions membership (CM), working memory support (WMS), condition relationship (CR), conflict set support (CSS), and resolution support (RS). Please refer to the **Table 1** from Ref. [12] below for more information.

Table 1 makes it clear that in event management situations, TREAT requires less memory, LEAPS is an excellent finisher, and RETE is a strong initiator. Better outcomes may be obtained by cascading these three algorithms into a single RETE-TREAT-LEAPS (RTL) block. The CM capabilities of RETE and the joint WMS block can be combined to create an effective RTL block that extracts the best rule and CR intersection to determine the optimal CSS set for RS.

3.3 MA-DSS response block (MD-RB)

The MA-RB consists of a shared WMS excess block, IoT GATOR block, and event response block all running in parallel as shown in **Figure 5**. In the multi-agent system

Algorithm	CM	WMS	CR	CSS	RS
LEAPS					
TREAT					
RETE					

Table 1.
LHS matching algorithm status information.

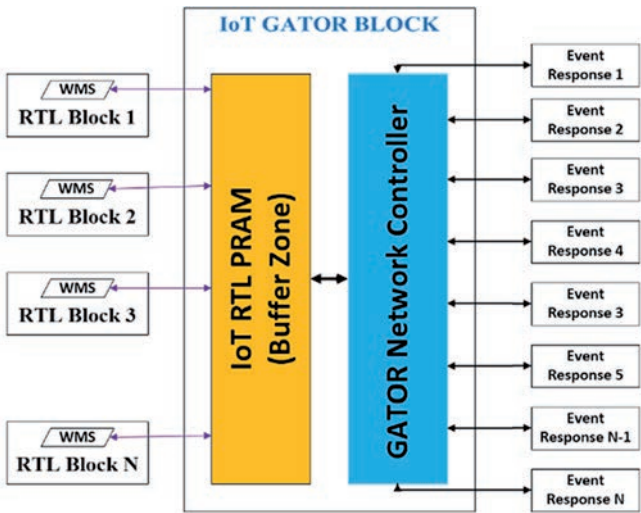


Figure 5.
Multi-agent MD-RB inference engine (IoT GATOR block) diagram.

context, **Figure 5**'s IoT GATOR block diagram illustrates that each city's Real-Time Logic (RTL) block has its Working Memory Set (WMS) mapped to an IoT RTL programmable RAM. This RAM acts as a buffer zone for event management, facilitating the routing of instructions and countermeasures to protect the population.

4. Multi-agent SHM synthetic DADO machine (MA-SDM) architecture

DADO is a large-scale parallel machine designed for the efficient execution of expert systems, including multiple independent ones. The proposed multi-agent SHM DADO machine (MA-SDM) implemented in a Multi-Agent Decision Support System (MA-DSS), incorporates both probabilistic and deterministic models. This system processes data using AI and machine learning algorithms within its inference engine. Real-time rules are managed through a working memory and network hierarchy, with rules and definitions sourced from an SHM knowledge base. MA-SDM was structured as a system with a hardware block and a software block.

4.1 MA-SDM hardware block

The MA-SDM system hardware has three blocks:

- MA-SDM Remote Block (MA-SDM-RB)
- MA-SDM Site Block (MA-SDM-SB)
- MA-SDM Cloud Block (MA-SDM-CB)

4.1.1 MA-SDM remote block (MD-SDM-RB)

It is composed of two layers of RPi3 OSB and CANOpen sensor nodes, which are comprised of a 14-bit accelerometer (MMA8451Q) and a 32-bit microcontroller with a CANOpen transceiver (STM32F10RBT6) (**Figure 6**).

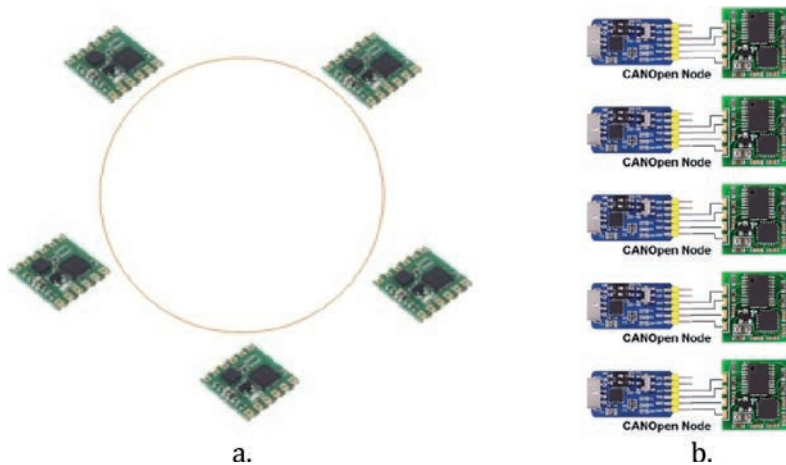


Figure 6. SHM remote block. (a) Circular grid topology, (b) Linear grid topology.

Along with configuration options, **Figure 2** displays both local and remote displays. The nodes were placed in a circular and rectilinear arrangement to track primary (P), secondary (S), and Rayleigh (R) seismic waves.

4.1.2 MA-SDM site block (MA-SDM-SB)

It consists of three layers: the application gateway (Banana Pi R1), the mesh network layer (built-in RPi3) with Wi-Fi and Bluetooth, and the bridge layer (wired and wireless). As illustrated in **Figure 7**, there are two choices for both local as well as remote display and configuration: as Bananian OS on the SHM gateway, and as SSH for Raspbian OS on the OSB using VNC.

4.1.3 MA-SDM cloud block (MA-SDM-CB)

The first is a personalized cloud-based on a local workstation serving as a server and enhanced with Apache, MySQL, LabView and MATLAB. The second is a public cloud built on open-source IoTs that will support Kaa and EmonCMS services. At present, it is simply connected to ThingSpeak IoT. **Figure 2** illustrates this with two clouds as three layers of the MA-SDM-CB system (IoT Edge, IoT Gateway, Servers).

4.2 MA-SDM software block

The software block has three parts as displayed in **Figure 3**:

- Multi-Client IoT Block
- Multi-Gateway IoT Block
- Multi-Server IoT Block

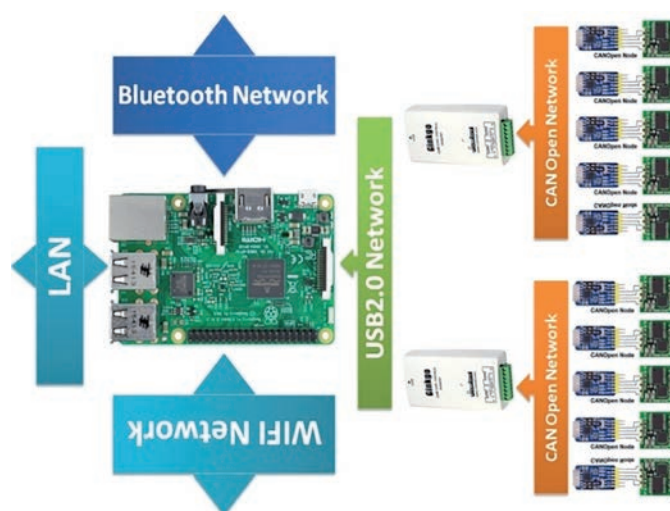


Figure 7.
 SHM-RSS Site block.

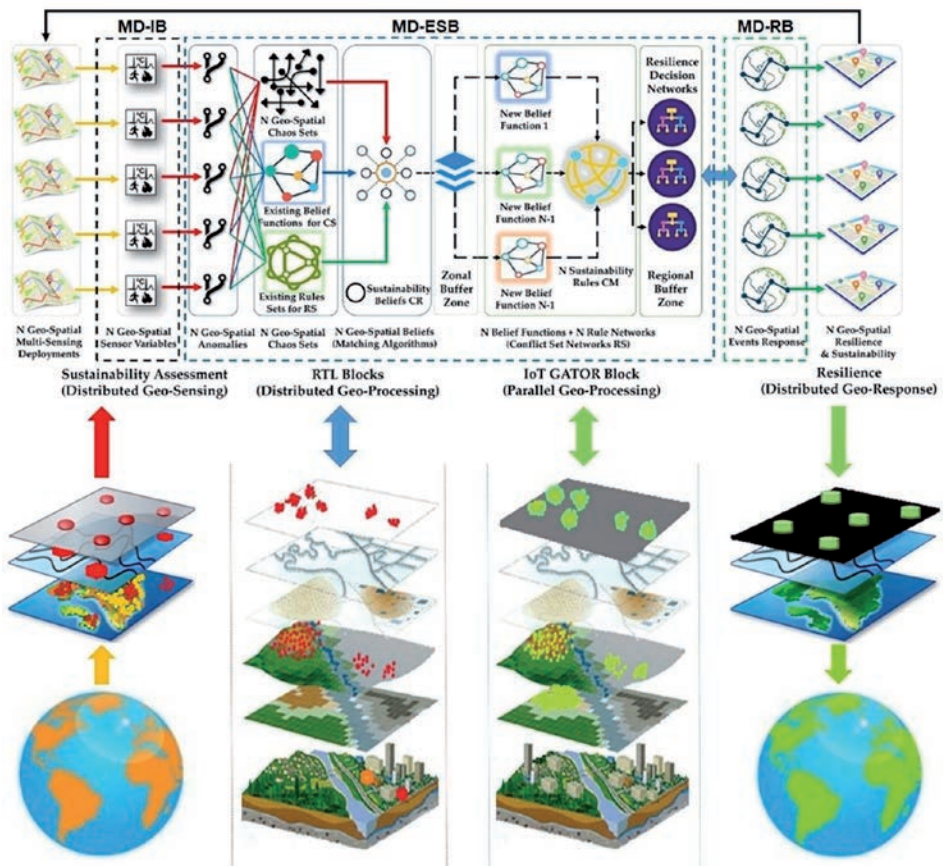


Figure 8.
Complete interaction diagram for geo-resilience support system (GRSS).

The IoT system is structured with a client, gateway, and server architecture, leveraging various software technologies for event management using relevant algorithms, while also ensuring self-hardware monitoring. MATLAB and LabView coordinate with Visual Studio through wrapper functions, and SQLite3 and MySQL clients operate at the IoT client level to collect and store data for analytics processors.

Finally, integrating Section 2 and Section 3, the result of multi-agent SHM IoT is presented in **Figure 8**.

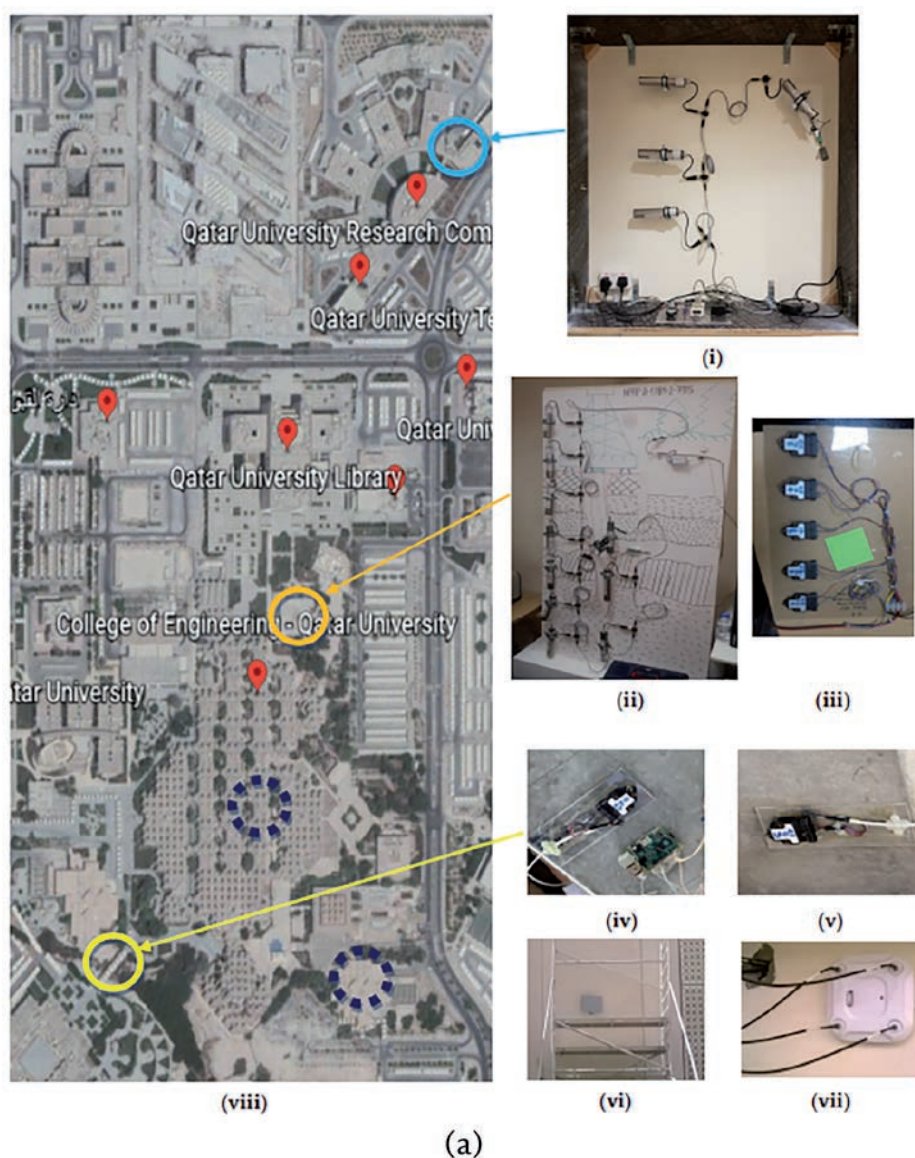
The MA-ESB, designed in Python 2.7, utilizes DSS and integrates an RTL block at the OSB or SHM site level on Raspberry Pi 3, as well as IoT RTL Level 1 PRAM at OSB and RTL Level 2 PRAM at SHM or RSS gateway levels. Meanwhile, a Dell Xeon Hex-Core Server Workstation in PRCL0 handles GNC and RSS dashboard server functions.

5. Case study: SHM systems deployed in Qatar university buildings

The SHM system prediction for Qatar University (QU), a possible RSS research site, was the selected case study. Three distinct specification and configuration systems—MA-SDM-QU-B09 Lab, MA-SDM-QU-CO5-Bridge,

and MA-SDM-QU-H10-Research Complex—were physically placed in QU at three distinct sites, each 1 kilometer away. The MA-SDM-QU-B03 Corridor and MA-SDM-QU-B01 Higher Administrative Building are two virtual installations that are running concurrently. Details of the SHM system for these sites are presented in **Figure 9 (a)** and **(b)**.

Figure 9 shows how the two distinct remote blocks are linked to the SHM-OSB, also known as the out-surface board (OSB), *via* a CANopen network and a Ginkgo CAN-USB adapter. The CANopen addresses on both sides are 1, 2, 3, 4, and 5. Nodes at RSS-SHM site A and site B are based on biaxial inclinometers and accelerometers, respectively. The two SHM site nodes are linked to the RSS gateway's I/O multiplexer block, which includes a power supply unit (PSU), communication ports, and a keyboard-video-multimedia (KVM) switch for multiple displays.



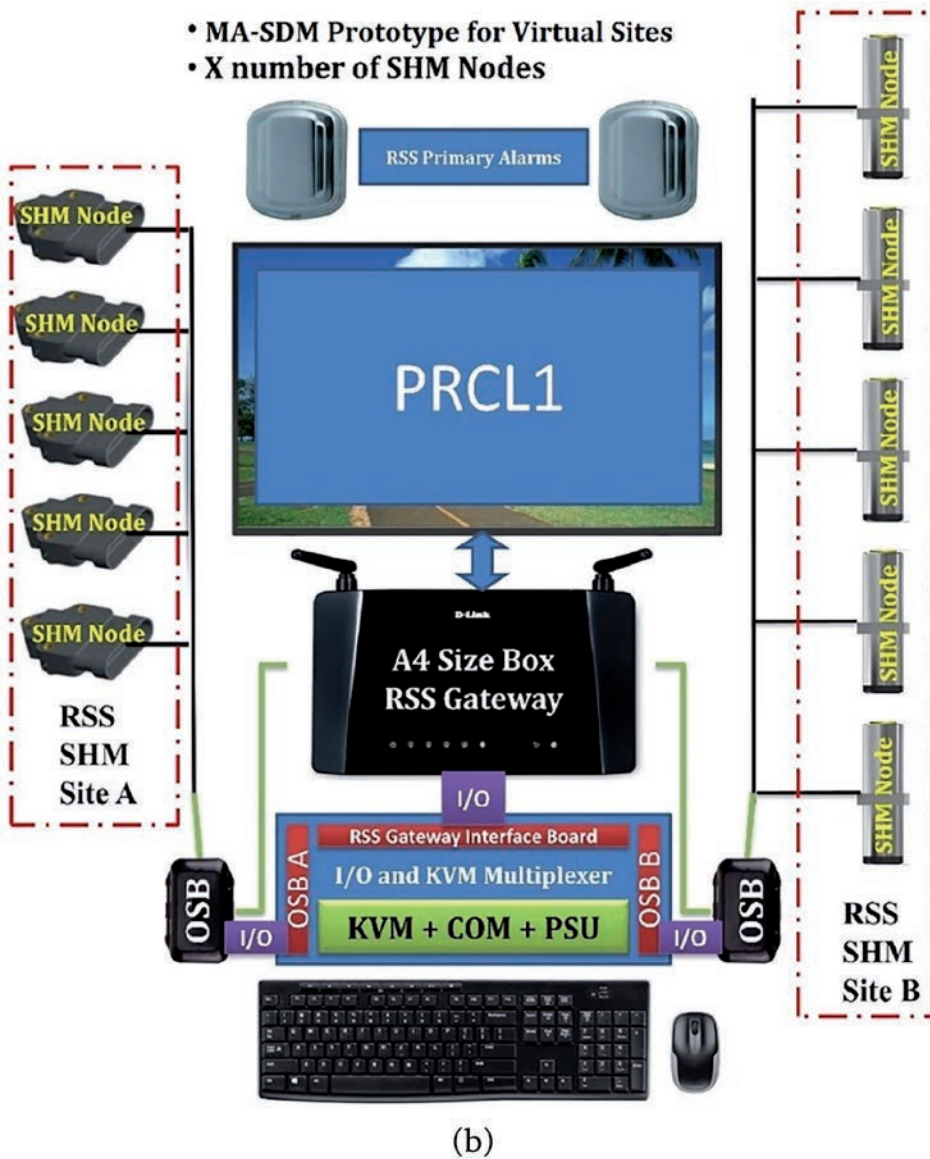


Figure 9.

Complete MA-SDM IoT agents deployments at QU, Doha, Qatar. (a) Case study: Deployment details for QU SHM Sites: (i) H10 SHM Site with 4 CSNs and 1 SSN; (ii) B09 Lab with 10 CSNs on SHM stand; (iii) B09 Lab with 5 SSNs on the table; (iv) OSB and an SSN QU Bridge Site; (v) SSN QU Bridge Site; (vi) ODU for QU Bridge site mounted on the outside of Co5's boundary wall; (vii) IDU for QU Bridge site mounted on the inside of Co5's boundary wall; (viii) Google Maps overview for QU SHM sites. (b) MA-SDM prototype for virtual sites (dotted dark blue circles) on map.

6. Results and discussion

The data payload in terms of packets defines the performance of cloud-based MA-DSS. The entire MA-DSS for QU was designed and deployed with the system configuration given in **Table 2**.

Systems	Nodes quantity	Data-rate	Packet size	Data-rate	Protocols
PBCL	1	128 Mbps	1500 Bytes	128 Mbps	IPv4
PRCL-N	2+	1 Gbps	1500–9000 Bytes	1 Gbps	IPv4
MA-SDM-Gateway	2+	1 Gbps	2312 Bytes	1 Gbps	IPv4
SHM Site	3+	100 Mbps	32-Bit to 251 Bytes	100 Mbps	IPv4 & BT4.2
Remote Block	15+	250 kbps	8 Bytes	250 kbps	CANopen

Table 2.
MA-SDM IoT network configuration.

Data processors and data busses, by any definition, are the foundation of any data processing system. As data processors, databases’ size and storage technology—which includes read/write and access times—have a significant impact on a system’s capacity to handle data. **Table 3** provides the MA-SDM-ESB data setup for the synthetic DADO machine.

Systems	Role	Size on disk	Database	Technology
PBCL	Server	3 TB	MySQL 5.5	PCI-E SSD
PRCL-N	Server	1 TB	MySQL 5.5	m.2 SSD
MA-SDM-Gateway	Client	200GB	MySQL 5.5	SATA SSD
SHM Site	Client	8GB	SQLite 3.0	Micro-SD-XC
Remote Block	Node	32 KB	NIL	SPIFFS

Table 3.
MA-SDM-ESB data configuration.

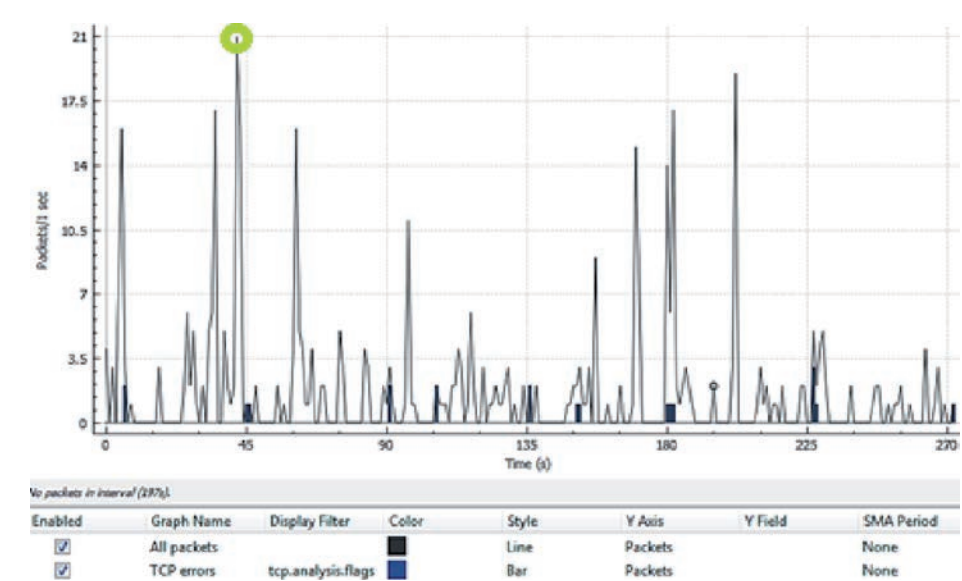


Figure 10.
SHM-RSS site to gateway I/O graph.

Additionally, the data architecture was created so that the number of data processing clusters or the breadth of data blocks increased as the hierarchy rose. Wireshark 3.0's packet capture statistics trace utility provides the first CANopen to Wi-Fi performance I/O plot.

Figure 10 shows that, according to **Table 2**, a maximum of less than 21 packets per second, or 47,225 bytes per second, or 377.8 kbps, were transmitted from SHM Sites to RSS Gateway. The only anomaly data A is passed up in the hierarchy to avoid data fusion and collusion instances. With 15 x-axis tilt-angles (θ_x), 15 y-axis tilt-angles (θ_y), 15 x-axis accelerations (A_x), 15 y-axis accelerations (A_y), and 15

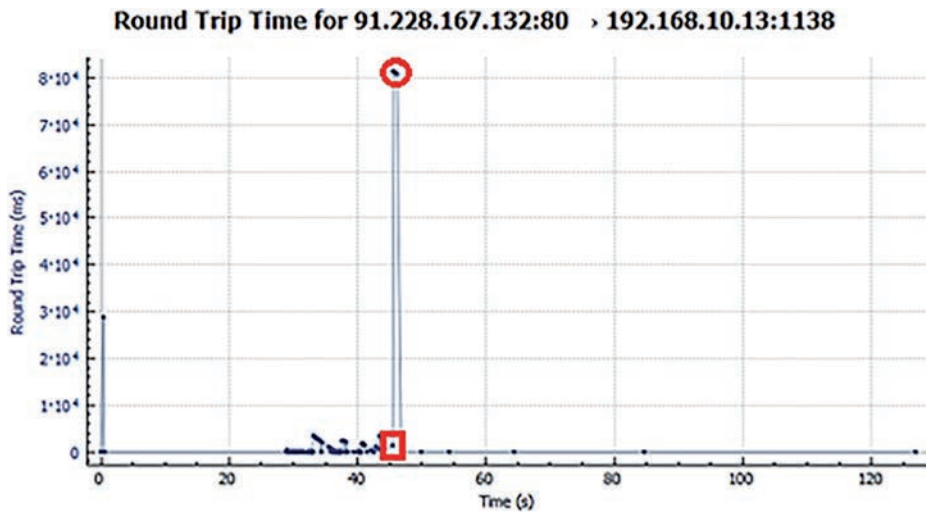


Figure 11.
SHM-MA-DSS belief function update round trip.

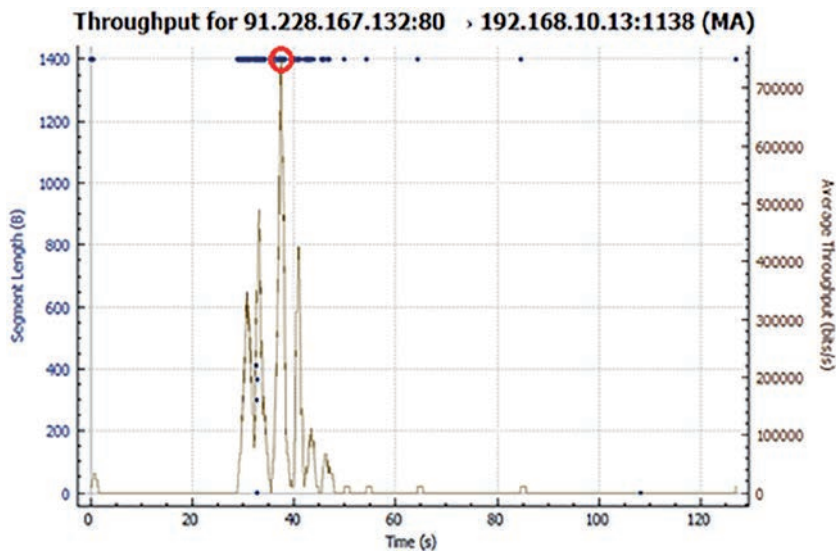


Figure 12.
SHM-MA-DSS throughput for first 2 minutes for 75 SHM variables.

temperatures, this data rate is suitable for 15 CANopen nodes with a total of 75 variables. The optimal speed is 3550 kbps for CANopen three clusters alone, or 8193.75 CANopen packets/second at 250 kbps for 15 nodes. However, CANopen data nodes are transported as tables that can be queried as queries.

In the worst scenario, **Figure 11** illustrates that the maximum round trip time for updating the belief function in this MA-DSS-SHM is 46.1 seconds in 81,005 ms. **Figure 9** shows that the first belief function was updated at a timeslot of 622 ms in 22.7 seconds. Even a single solar day may be needed to update a single belief function.

A maximum throughput of 742,500 bits/second was achieved in 38.2 seconds using an MA-DSS gateway with MTU = 1500B and a segment size of 1482 bytes, as illustrated in **Figure 12**. The very competitive benchmark of obtaining event response below MTU at such a low data rate has enabled further studies on nationwide cloud-based MA-DSSs.

The MD-SDM-RB 120,000+ sequence numbers produced for the first event response at 29.2 seconds and belief function update for a certain SHM variable are shown in **Figure 13**. The blue dots show that response generation remained constant with each subsequent sequence, whereas the horizontal green line shows the moments when one belief function data remained constant. The belief function reached maturity after more than 365,000 sequence numbers, as seen by the horizontal line. At 48 seconds, the 372,000th token was stopped during the creation of the event. Only convex problems can cause it.

A thorough system overview is given by the MD-SDM-ESB dashboard, which is depicted in **Figure 14**. According to the green speedometer charts, resilience is not necessary. Indicating that the system meets the safety integrity level (SIL) criteria and regulations is the dashboard's lack of color speedometer charts. Only 40 belief functional values were modified in a 24-hour period, as shown in the line graph at the bottom.

The actual view obtained from Google Earth without the use of MA-DSS-ES is shown in **Figure 15(a)**. In **Figure 15(b)**, the applied algorithm for testing was the machine learning algorithm MOSPA for urban scale geo-informatics and SHM analytics, which was first presented in Ref. [24]. The MOSPA assessment metric is

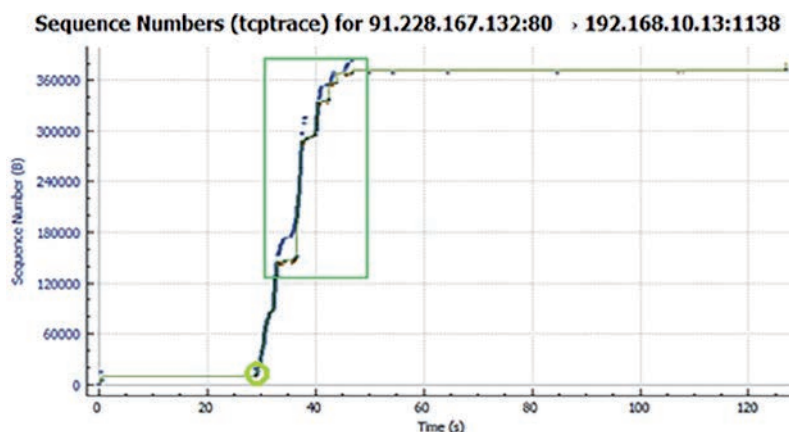


Figure 13.
 Sequence numbers generated in the first 2 minutes.

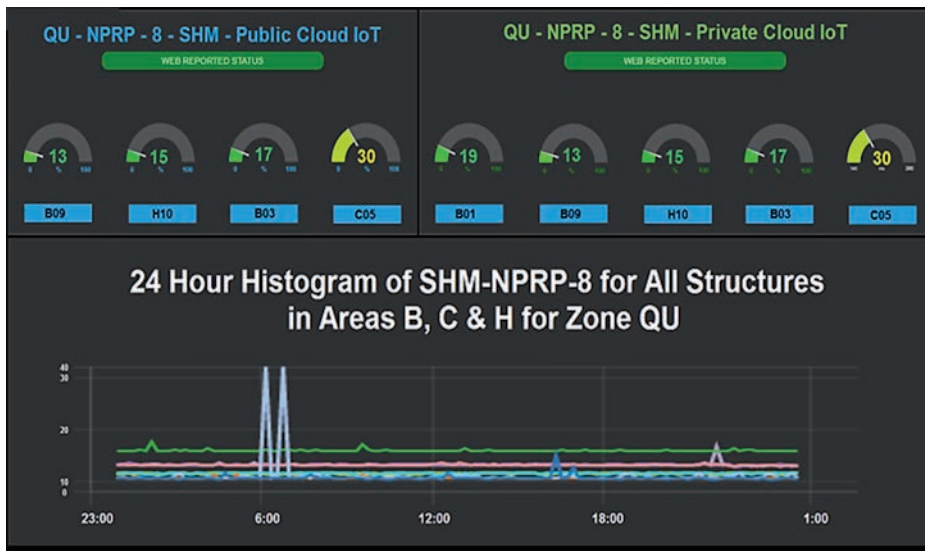


Figure 14.
QU-MA-DSS dashboard for hybrid cloud.

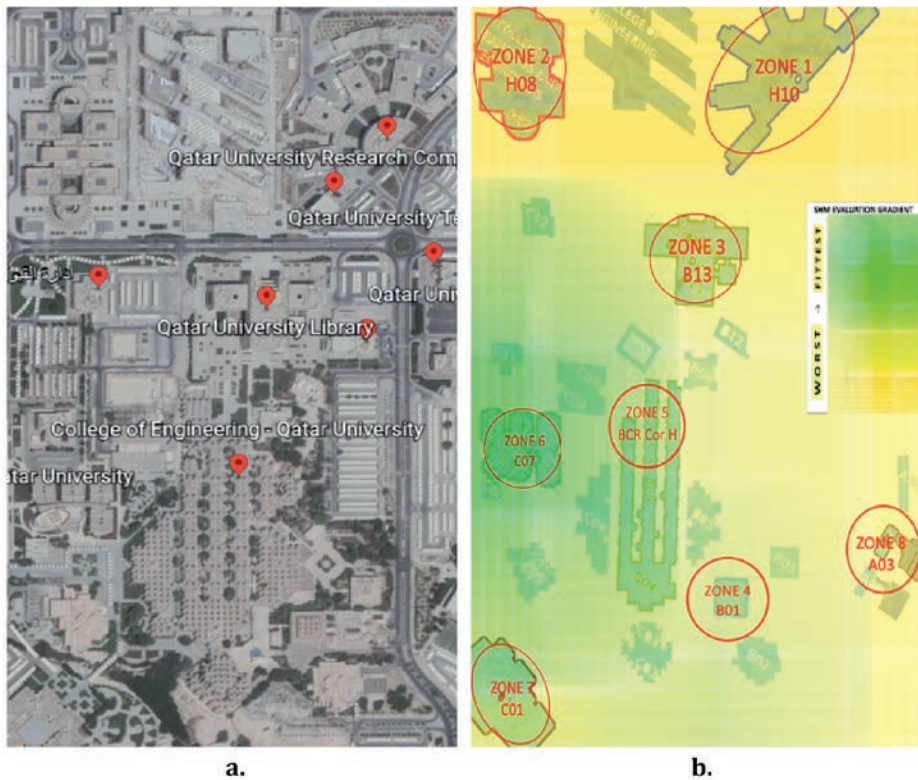


Figure 15.
MA-SDM-ESB events response layer using MOSPA in MA-SDM-ESB. (a) Google Earth view. (b) MA-SDM-ESB layer.

the cost of damage. Maximum robustness is required at maximum cost. The term “cost” describes the projected and anticipated lifetime harm to human resources and structures in real-time.

7. Conclusions

This study used a structured RSS framework to enable creative real-time artificial intelligence by investigating the design and implementation possibilities of expert systems within multi-agent systems (MAS) for Internet of Things scenarios. A synthetic DADO machine with a thorough hierarchy serves as the basis for the extremely symmetric design of the suggested RSS hardware, which guarantees optimal performance in a variety of scenarios. The SRIB architecture, built on SQL RDBMS, improves the adaptability of RSS-based expert systems by enabling smooth agent coordination. Using RETE, TREAT, LEAPS, and GATOR to sequentially optimize matching, conflict resolution, and response network building further improves system efficiency. A standardized WMS supported the remarkable resilience of cloud-based apps. Further demonstrating the architecture’s ability to promote MAS-driven IoT, the GNC-enabled RSS gateway helped to produce more efficient and rapid response sets.

Acknowledgements

This publication was made possible by NPRP grant # 8-1781-2-725 from the Qatar National Research Fund (a member of Qatar Foundation). The publication of this article was funded by the Qatar National Library. The statements made herein are solely the responsibility of the authors. In addition, the authors would like to acknowledge the support of the Osamah Al-Sayyed research center fund award at the University of Prince Mugrin, and of the Research Development and Innovation Authority (RDIA), Saudi Arabia, under grant number: 13465-UPM-2023-UPM-R-3-1-EF.

Conflict of interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Author details

Hasan Tariq^{1*}, Farid Touati¹, Mohammed Abdulla E. Al-Hitmi¹, Damiano Crescini² and Adel Ben Mnaouer³

1 Department of Electrical Engineering, College of Engineering, Qatar University, Doha, Qatar

2 Brescia University, Brescia, Italy

3 Dr. Hussein AlSayyed Center for Scientific Research, Department of Scientific Research and Graduate Studies, University of Prince Mugrin, Medina, Saudi Arabia

*Address all correspondence to: hasan.tariq@qu.edu.qa

IntechOpen

© 2025 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Al-Jumaily A, Al-Jaafreh M. Multi-Agent System Concepts Theory and Application Phases. Mobile Robotics, Moving Intelligence. Germany/ARS, Austria: Pro Literatur Verlag; 2006. DOI: 10.5772/4730
- [2] Huerta Barrientos A, Nila Luevano A. A state-of-the-art survey on various domains of multi-agent systems and machine learning. In: Artificial Intelligence. London, UK: IntechOpen; 2023. DOI: 10.5772/intechopen.107109
- [3] Tariq H, Touati F, Crescini D, Ben Mnaouer A. State-of-the-art low-cost air quality sensors, assemblies, calibration and evaluation for respiration-associated diseases: A systematic review. *Atmosphere*. 2024;**15**(4):471. DOI: 10.3390/atmos15040471
- [4] Kozhevnikov S, Svitek M, Skobelev P. Multi-agent approach for smart Resilient City. In: Borangiu T, Trentesaux D, Leitão P, Cardin O, Lamouri S, editors. Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2020. Studies in Computational Intelligence. Vol. 952. Cham: Springer; 2021. DOI: 10.1007/978-3-030-69373-2_15
- [5] Tan E, Ma X, Du Y, Zhang Z. Enhancing urban metro system resilience under disruptive events through multi-agent reinforcement learning. *Journal of Transportation Safety and Security*. 2024;**16**(10):1184-1211. DOI: 10.1080/19439962.2023.2300272
- [6] Tariq H, Touati F, Al-Hitmi MAE, Crescini D, Mnaouer AB. Design and implementation of information centered protocol for long haul SHM monitoring. In: IEEE International Conference on Design and Test of Integrated Micro and Nano-Systems (DTS). Gammarth, Tunisia; 2019. DOI: 10.1109/DTSS.2019.8915286
- [7] Mustapha K, Mcheick H, Mellouli S. Smart cities and resilience plans: A multi-agent based simulation for extreme event rescuing. In: Gil-Garcia J, Pardo T, Nam T, editors. Smarter as the New Urban Agenda. Public Administration and Information Technology. Vol. 11. Cham: Springer; 2016. DOI: 10.1007/978-3-319-17620-8_8
- [8] Tariq H, Tahir A, Touati F, Al-Hitmi MA, Crescini D, Mnaouer AB. Structural health monitoring and installation scheme deployment using utility computing model. In: 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS). 2018. DOI: 10.1109/EECS.2018.00019
- [9] Zhao X, Yuan S, Zhou H, Sun H, Qiu L. An evaluation on the multi-agent system based structural health monitoring for large scale structures. *Expert Systems with Applications*. 2009;**36**(3, Part 1):4900-4914. ISSN 0957-4174,. DOI: 10.1016/j.eswa.2008.05.056
- [10] Tariq H, Touati F, Al-Hitmi MA, Crescini D, Mnaouer AB. A real-time early warning seismic event detection algorithm using smart geo-spatial Bi-axial inclinometer nodes for industry 4.0 applications. *Applied Sciences*. 2019;**9**:3650. DOI: 10.3390/app9183650
- [11] Forgy CL. RETE: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*. 1982;**19**(1):17-37. ISSN 0004-3702. DOI: 10.1016/0004-3702(82)90020-0

- [12] Daniel PM. TREAT: A better match algorithm for AI production systems. In: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1 (AAAI'87)*. Seattle, Washington, USA: AAAI Press; 1987. pp. 42-47. DOI: 10.5555/1856670.1856678
- [13] Brant DA, Miranker DP. Index support for rule activation. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (SIGMOD '93)*. New York, NY, USA: Association for Computing Machinery. pp. 42-48. DOI: 10.1145/170035.170047
- [14] Stolfo SJ, Miranker DP. The DADO production system machine. *Journal of Parallel and Distributed Computing*. 1986;**3**(2):269-296. DOI: 10.1016/0743-7315(86)90007-9
- [15] Salvatore JS, Shaw DE. DADO: A tree-structured machine architecture for production systems. In: *Proceedings of the Second AAAI Conference on Artificial Intelligence (AAAI'82)*. Pittsburgh, Pennsylvania, USA: AAAI Press; 1982. pp. 242-246
- [16] Smarsly K, Law K, Hartmann D. A multi-agent-based collaborative framework for a self-managing structural health monitoring system. *Journal of Computing in Civil Engineering*. 2012;**26**:76-89. DOI: 10.1061/(ASCE)CP.1943-5487.0000107
- [17] Tariq H, Abdaoui A, Touati F, Al-Hitmi MA, Crescini D, Mnaouer AB. A real-time gradient aware multi-variable handheld urban scale air quality mapping IoT system. In: *IEEE International Conference on Design and Test of Integrated Micro and Nano-Systems (DTS)*. Hammamet, Tunisia; 2020. pp. 1-5. DOI: 10.1109/DTS48731.2020.9196131
- [18] Sun H, Yuan S, Zhao X, Zhou H, Lianq D. Technology of structure damage monitoring based on multi-agent. *Journal of Systems Engineering and Electronics*. 2010;**21**(4):616-622. DOI: 10.3969/j.issn.1004-4132.2010.04.014
- [19] Santos A, da Silva P, Smith AM, Schaeffer-Filho A. Resilience support in software-defined networking: A survey. *Computer Networks*. 2015;**92**(Part 1): 189-207. ISSN 1389-1286. DOI: 10.1016/j.comnet.2015.09.012
- [20] Hanson EN, Hasan MS. GATOR: An Optimized Discrimination Network for Active Database Rule Condition Testing. Technical Report TR-93-36. USA: University of Florida; 1993
- [21] Xiao Y, Yang H, Chen L, Huang H, Chang M. Urban resilience assessment and multi-scenario simulation: A case study of three major urban agglomerations in China. *Environmental Impact Assessment Review*. 2025;**111**:107734. ISSN 0195-9255. DOI: 10.1016/j.eiar.2024.107734
- [22] Tariq H, Touati F, Crescini D, Mnaouer AB. IoT-based Bi-cluster forecasting using automated ML-model optimization for COVID-19. *Atmosphere*. 2023;**14**:534. DOI: 10.3390/atmos14030534
- [23] Bhuiyan MZA, Wu J, Wang G, Cao J, Jiang W, Atiquzzaman M. Towards cyber-physical Systems Design for Structural Health Monitoring: Hurdles and opportunities. *ACM Transactions on Cyber-Physical Systems*. 2017. 26 pages;**1**(4, Article 19). DOI: 10.1145/3086508
- [24] Tariq H, Tahir A, Touati F, Al-Hitmi MA, Crescini D, Mnaouer AB. Geographical area network—Structural health monitoring utility computing model. *ISPRS International Journal*

of Geo-Information. 2019;**8**:154.
 DOI: 10.3390/ijgi8030154

[25] Ray PP. A survey on internet of things architectures. Journal of King Saud University - Computer and Information Sciences. 2018;**30**(3):291-319. ISSN 1319-1578. DOI: 10.1016/j.jksuci.2016.10.003

[26] Sethi P, Sarangi Smruti R. Internet of things: Architectures, protocols, and applications. Journal of Electrical and Computer Engineering. 2017;**9324035**. 25 pages. DOI: 10.1155/2017/9324035

[27] Domínguez-Bolaño T, Campos O, Barral V, Escudero CJ, García-Naya JA. An overview of IoT architectures, technologies, and existing open-source projects. Internet of Things. 2022;**20**:100626. ISSN 2542-6605. DOI: 10.1016/j.iot.2022.100626

[28] Tariq H, Abdaoui A, Touati F, Al-Hitmi MAE, Crescini D, Manouer AB. A real-time gradient aware multi-variable handheld urban scale air quality mapping IoT system. In: 2020 IEEE International Conference on Design & Test of Integrated Micro and Nano-Systems (DTS), Hammamet, Tunisia. 2020. pp. 1-5. DOI: 10.1109/DTS48731.2020.9196131

[29] Tariq H, Abdaoui A, Touati F, Al-Hitmi MAE, Crescini D, Mnaouer AB. Design and implementation of cadastral geo-spatial IoT network gateway analyzer for urban scale infrastructure health monitoring. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA. 2020. pp. 0858-0863. DOI: 10.1109/CCWC47524.2020.9031188

[30] Al-Qaseemi SA, Almulhim HA, Almulhim MF, Chaudhry SR. IoT architecture challenges and issues: Lack of standardization. In: 2016 Future

technologies conference (FTC). San Francisco, CA, USA; 2016. pp. 731-738. DOI: 10.1109/FTC.2016.7821686

[31] Yaqoob I et al. Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges. IEEE Wireless Communications. 2017;**24**(3):10-16. DOI: 10.1109/MWC.2017.1600421

[32] Tariq H, Abdaoui A, Touati F, Al-Hitmi MAE, Crescini D, Ben Manouer A. Design and implementation of multi-protocol data networks interface detector in heterogeneous IoTs. In: ICloT. Doha, Qatar; 2020. pp. 563-569. DOI: 10.1109/ICloT48696.2020.9089435

[33] Ganchev I, Ji Z, O'Droma M. A generic IoT architecture for smart cities. In: 25th IET Irish Signals and Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CICT 2014), Limerick. 2014. pp. 196-199. DOI: 10.1049/cp.2014.0684

*Edited by Nohaidda Sariff, Zool Hilmi Ismail,
Denesh Sooriamoorthy and Puteri Nor Aznie Fahsyar*

Multi-Agent Systems - From Basic Concepts to Cutting-Edge Technologies presents a rigorous and integrated perspective on the theory, design, and deployment of intelligent agent-based systems in complex and distributed environments. The book examines core concepts of multi-agent systems, including autonomy, cooperation, coordination, negotiation, and decentralized decision-making, while highlighting recent advances in agent architectures, learning-enabled agents, and agent-based control frameworks. Emphasis is placed on the application of artificial intelligence, deep reinforcement learning, distributed optimization, and IoT-integrated agent systems to address real-world challenges in robotics, smart energy management, microgrid protection, and resilient urban infrastructure. By focusing on scalability, robustness, fault tolerance, and adaptability, the book demonstrates the technical advantages of multi-agent approaches over centralized solutions in dynamic, uncertain, and large-scale systems. This volume serves as a technical reference for researchers, postgraduate students, and professionals seeking a systematic understanding of state-of-the-art methodologies, implementation considerations, and emerging research directions in intelligent distributed systems. Through its balanced coverage of foundational principles and advanced technological developments, the book supports interdisciplinary research and provides a strong knowledge base for the development of next-generation autonomous and agent-oriented systems.

Andries Engelbrecht, Artificial Intelligence Series Editor

Published in London, UK

© 2026 IntechOpen
© your_photo / iStock

IntechOpen

ISSN 2633-1403

ISBN 978-1-83635-148-1

