Integration of Robot Swarm Intelligence for Distributed Manufacturing Systems

Anca-Gabriela PETRESCU[1]*, Ionica ONCIOIU[2,3], Mihai PETRESCU[1], Florentina-Raluca BÎLCAN[1], Marius PETRESCU[3], Marius Florin GHIBERDIC[4]

[1] Valahia University of Targoviste, 2 Carol I Avenue, Targoviste, 130024, Romania
anki.p2007@gmail.com (*Corresponding author*), bilcan.florentina.raluca@gmail.com, mihai_tina@yahoo.com

[2] Titu Maiorescu University, 189 Calea Vacaresti Street, Bucharest, 040051, Romania
ionica.oncioiu@prof.utm.ro

[3]Academy of Romanian Scientists, 3 Ilfov Street, Bucharest, 050044, Romania
profdrmpetrescu@yahoo.com

[4] Doctoral School of Management, Bucharest University of Economic Studies,
6 Piata Romana Street, 010374 Bucharest, Romania
mfghiberdic@yahoo.com

Abstract: The integration of Robot Swarm Intelligence (RSI) into Industry 4.0 represents a significant step towards fully autonomous and distributed manufacturing systems. Inspired by the biological swarm behaviour, RSI enables the adaptive collaboration of groups of robots with the purpose of optimising industrial processes. This paper presents a novel architecture for integrating RSI with Industry 4.0 technologies like IoT, Cyber-Physical Systems (CPS), and Big Data Analytics for distributed manufacturing systems. Advanced simulation tools are employed for evaluating the performance of the proposed architecture in three manufacturing scenarios, focusing on indicators such as the resource utilisation rate, the task completion time, and fault tolerance. The obtained results demonstrate the potential of RSI to significantly optimise industrial processes, and reduce waste in distributed manufacturing environments, and highlight its transformative role in promoting innovation and competitiveness in smart manufacturing.

Keywords: Robotic swarm, Multi-agent systems, Optimise processes, Automatic algorithms.

1. Introduction

Industry 4.0 will enable no less than a revolution in business processes and business models, and the interaction between people and IT will be redefined in the already clearly emerging scenarios of Industry 4.0, corresponding to new internal and external processes (Silva et al., 2016). More responsibility will lie in leading and orchestrating processes that take place in real time and hand in hand with machines and intelligent systems. The entire process landscapes are currently changing due to digitalisation and networking and will continue to do so in the future.

Lean processes optimise business processes by reducing process steps or by automation, on the one hand to meet customer requirements for individual, high-quality products and services and fast response to orders, on the other hand to avoid bottlenecks or delays in supply (Gerkey & Matarić, 2004). Among other things, the aim is to continuously reduce direct and indirect costs by identifying relevant cost drivers and waste in existing processes.

The Internet of Things and cyber-physical systems (CPS) are examples of these technologies. Industry 4.0 focuses on digitisation, automation, and connectivity, whereas Industry 5.0 introduces additional dimensions by stressing collaboration, sustainability, and customisation in the relationship between people and technology. Despite the advancements in this field of technology, these technologies are still not being utilised broadly or sustainably, and they continue to struggle to integrate into complicated industrial environments (Alitappeh & Jeddisaravi, 2022). In this respect, robotic swarm intelligence (RSI) is a new technology that provides a decentralised way to improve dispersed production.

Within a RSI system, robots operate autonomously but collaborate through local interactions to achieve common goals (Mendiburu et al., 2022). This capability enables RSI-based systems to swiftly adapt to evolving circumstances and effectively manage intricate tasks, rendering them suitable for dynamic industrial settings. Research indicates that RSI can decrease processing times by as much as 40% in production optimisation scenarios (Birattari et al., 2019).

Equally, the RSI represents a promising approach, allowing robots to make autonomous decisions based on local information and thus eliminating

https://doi.org/10.24846/v34i1y202503

the need for a centralised command structure. For example, in complex supply chains, RSI can optimise resource allocation, reduce operational costs, and increase system resilience (Kuckling et al., 2022). These capabilities are particularly relevant in the context of digitalisation and fluctuating market demands.

Even if certain professions might be eliminated due to automation, RSI opens up new chances in fields like data analysis, programming, and robot maintenance. The use of RSI technologies hastened the recent 25% surge in the need for experts in the domain of automation and artificial intelligence, according to recent figures (Farivarnejad & Berman, 2021; Trelea, 2003). Therefore, this technological transition requires strategic planning and adapted public policies.

As well, the implementation of RSI in real industrial environments poses significant challenges. Problems such as latency in communication, conflicts between robots, and technological interoperability need to be solved. In the specialist literature, only 15% of studies on RSI focus on addressing these practical difficulties (Akopov, 2024; Wang et al., 2020). With the help of advanced simulations of industrial distributed manufacturing scenarios, using platforms like MATLAB and ROS, this paper addresses these challenges by creating and testing an integrated architecture for RSI.

The originality of this work lies in its interdisciplinary and practical approach to distributed manufacturing systems. This article aims to evaluate RSI in an applied context, using realistic distributed manufacturing scenarios, unlike most studies that limit themselves to theoretical simulations (Chakraa et al., 2023; Larabi-Marie-Sainte & Ghouzali, 2020).

This study takes a closer look at how RSI fits into the larger framework of Industry 4.0, focusing not just on its technological growth but also on its potential to reshape modern manufacturing systems. Its findings offer a starting point for future work in distributed and autonomous production, pointing to new opportunities for innovation. It also highlights the broader benefits of RSI, including its economic, social, and environmental impacts, and suggests a practical path for its successful adoption in smart manufacturing.

The structure of this paper is as follows. Section 2 explores the relevant literature, highlighting the recent developments in Industry 4.0 and its intersection with robotic swarm intelligence. Section 3 outlines the methodology, describing how IoT, Big Data, and cyber-physical systems are combined to form an integrated and adaptive model. Section 4 discusses the results obtained from simulations, focusing on the system's performance, practical advantages, and areas for improvement. Finally, Section 5 sets forth the main conclusions and suggests future directions for enhancing adaptive production systems in the face of evolving industrial challenges.

2. Theoretical Framework

The advanced integration of digital technologies is characteristic of Industry 4.0 - a profound transformation in the design and operation of industrial processes. The real-time monitoring and control of industrial processes through these technologies ensures a high operational efficiency and an increased adaptability to market requirements. In particular, the IoT creates a network of interconnected devices capable of collecting and transmitting data, which allows organisations to optimise resource usage and respond quickly to variations in the operational environment (Han & Song, 2024).

In general terms, optimisation may be thought of as a strategy that needs two fundamental elements: adaptability and purpose. To be precise, optimisation can be described as: (1) a systematic alteration, modification, or adaption of a process that tries to (2) fulfil a pre-specified objective. This purpose might be the maximum or lowest value of a numerical function that the user defines.

In other words, optimisation must have a teleological nature, which means that one must know what should be accomplished (maximise a function), but not how this can be achieved, which is the general path that leads to the intended objective. The essence of the entire field of study known as optimisation theory is to provide a realistic solution to this perplexing predicament.

Distributed production systems, integrated in the context of Industry 4.0, represent an operational model that optimises resource use and improves the coordination among autonomous units (Xie et al., 2019). Distributed systems provide redundancy and diminish susceptibility to failure, in contrast to centralised systems. This organisational structure facilitates the distribution of tasks across multiple nodes within the network, resulting in an enhanced flexibility and resilience to variations in market demands.

Furthermore, the implementation of complex predictive maintenance strategies is facilitated by distributed systems. Additionally, this method enables the integration of personalised production into industrial flows, thus ensuring process adaptability without sacrificing execution speed or quality for that. Companies can optimise equipment availability and reduce maintenance costs by proactively identifying potential malfunctions. This is achieved by utilising data collected by IoT and analysed through Big Data (Ligot & Birattari, 2020).

Robotic swarm intelligence (RSI) is based on the concept of collective behaviour inspired by nature, especially through the observation of colonies of bees, ants, or other social insects (Abd Elfattah et al., 2019). This bio-inspiration has led to the development of distributed robotic systems that simulate this behaviour to address problems such as the exploration, navigation, and optimisation of industrial processes (Trianni & Nolfi, 2011). For these algorithms, the main focus is on the cooperation between individual solutions, which results in a broader perspective of the algorithm to reach always better solutions.

The main algorithms used in RSI are particle swarm optimisation (PSO) and genetic algorithms. PSO simulates the behaviour of a swarm of particles moving in the solution space to find optimal values (Berman et al., 2009). It is successfully applied for nonlinear and multi-objective optimisation problems. Genetic algorithms, which are based on evolutionary ideas like mutation, selection, and recombination, can quickly find answers in a very large and complicated search space. These methods contribute to improving the performance of robotic systems by implementing adaptive and robust strategies.

In recent literature, the integration of Industry 4.0 technologies, such as the Internet of Things (IoT), big data analytics, and cyber-physical systems (CPS) with optimisation algorithms used in robotic swarm intelligence (RSI), has been recognised as having a fundamental role in improving the efficiency of industrial processes and increasing the ability of production systems to adapt to changing operational requirements. For example, a study by Dorigo, Theraulaz and Trianni (2020) demonstrated the application of the particle swarm optimisation (PSO) algorithm in the optimal placement of sensors in industrial processes, highlighting significant improvements in monitoring accuracy and operational efficiency.

In a comparable direction, Kuckling (2023) looked into smart factory energy management and optimised energy use by employing PSO. The obtained results demonstrated a considerable improvement in energy efficiency and a dramatic decrease in operating expenses. As a result, RSI algorithms may be valuable for resource management in intricate industrial contexts. A more sustainable use of energy resources and substantial cost reductions were the results of this integration.

Using PSO to optimise industrial mobile robot routes was also the focus of the work of Schranz et al. (2020). According to this study, particle swarm optimisation greatly improved the production system efficiency by decreasing the robot walking time and energy consumption. This method exemplifies the potential of RSI algorithms for improving the manufacturing facilities' internal logistics and material flows.

On the other hand, Bao & Zelinka (2019) studied the integration of PSO with cyber-physical systems (CPS) in the context of industrial process control. Their study showed that this combination was able to adapt to process fluctuations in real time and to continuously optimise operating parameters, thereby improving productivity and product quality. This study highlights the importance of integrating RSI algorithms with CPS to develop intelligent and adaptive production systems.

3. Data and Methodology

The employed methodology is based on the development of an advanced architectural model that integrates Industry 4.0 technologies such as the Internet of Things (IoT), centralized data hubs, big data analytics, and cloud computing with autonomous robots equipped with RSI algorithms. These technologies work together to create an interconnected and adaptable manufacturing ecosystem. Thus, manufacturing systems become more flexible and adaptable, capable of managing the complexity and uncertainty of the modern industrial environment. Figure 1 shows how components like IoT, sensors, robots, and cloud integration interact to support a decentralised, autonomous robotic swarm system. Therefore, each component of the proposed architecture fundamentally contributes to the consolidation of an intelligent and interconnected ecosystem, indispensable for the efficient and adaptable operation of distributed production systems as follows:

- the IoT platform represents the foundation of the digital infrastructure, with the role of ensuring the connectivity between sensors, robots, and other devices integrated into the system, facilitating the collection and distribution of data in real time, creating an ideal environment for the collaboration of autonomous robots. Its central functionality consists in a two-way communication, allowing both the transmission of commands to robots and the continuous monitoring of the system status;

- the autonomous robots operate according to the RSI principle and coordinate with each other in real time to optimise the production process. Each robot can make local decisions using information received from the IoT platform and feedback from other robots;

- the centralised data hub functions as a node for processing and distributing information, integrating the data collected from the IoT platform and distributing it to robots and analysis modules, being responsible for preprocessing and aggregating information, reducing redundancy, and improving data relevance;

- the sensors continuously monitor key variables such as temperature, pressure, vibration, and component position, and transmit the collected data to the IoT platform for further processing and analysis;

- the production line functions as a practical interface between digital technologies and the tangible result of the industrial process. In this field, the robots coordinated by RSI algorithms perform tasks in a synchronised manner, ensuring the efficient use of resources and the high quality of the final product;

- Big Data analytics leverages the vast amounts of information collected by sensors and robots to generate predictive insights and optimise the production processes. The advanced algorithms implemented in this module identify anomalies, prevent errors, and improve the overall performance of the system;

- cloud integration ensures the fast and secure access to information, facilitating real-time analysis and integration with other systems, providing the computing power necessary to support the complex algorithms used in RSI and the Big Data analysis, and contributing to the scalability and flexibility of the proposed architecture.

Figure 1 also provides a detailed description of the data flows between the components of this architecture. These flows not only show how the component pieces interact with each other, but they also help the system work by linking the activities of collecting, analysing, and executing data.

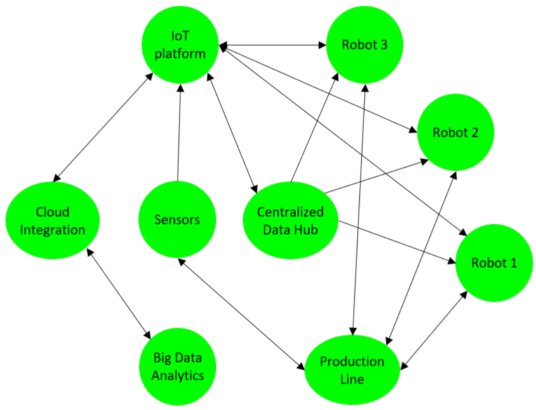

Figure 1. The Proposed Architecture for RSI Integration in Industry 4.0 and Data Flow

To test the proposed architecture and evaluate the performance of the distributed manufacturing system, both practical experiments and simulations were conducted. Tools such as MATLAB and Robot Operating System (ROS) were used due to their ability to model complex scenarios and replicate the collaborative actions of autonomous robots controlled through RSI. Additionally, the Gazebo simulator allowed the creation of realistic virtual environments, useful for testing and validating robot interactions under close-to-real conditions.

Three simulation scenarios were designed to evaluate the impact of RSI (Robot Swarm Intelligence) in various operational contexts, with increasing levels of complexity:

- Scenario 1 – Stable production line with sequential tasks: it was developed to represent a stable industrial environment, where each robot performs a fixed task, without dynamic shifts of responsibilities. The number of robots used is 3, each of them programmed to complete tasks in a sequential manner. This framework allows the analysis of individual robot performance and provides a reference for more complex scenarios;

- Scenario 2 – Distributed manufacturing with dynamic task allocation: the number of robots is 4, and tasks are no longer predefined, but they are dynamically distributed using RSI algorithms. The robots must collaborate to manage the production flows and optimise resource utilisation. This configuration tests the ability of RSI to intelligently allocate tasks and minimise robot downtime;

- Scenario 3 – Complex environment with limited resources and simulated faults: introduces a high level of complexity, including limited resources and a 5% probability of robot failure. Four robots are involved in this scenario, and the system must adapt by automatically redistributing tasks in case a robot becomes inoperable. The main goal is to evaluate fault tolerance and the system's ability to maintain operational efficiency under dynamic conditions.

These scenarios were chosen to reflect the challenges encountered in smart manufacturing, providing a logical progression from static and well-defined processes (Scenario 1), to collaborative optimisation (Scenario 2) and the management of defects and limited resources (Scenario 3).

In order to maintain the methodological consistency between the simulation scenarios and to allow for a precise comparison of the obtained results, a set of standardised experimental parameters was established, necessary for evaluating the performance of the proposed architecture. The communication rate between the robots was set at 10 Hz, ensuring a continuous flow of information and the efficient coordination of autonomous units during task execution. Regarding fault tolerance, Scenario 3 was designed to analyse the system's ability to adapt to faults, by introducing a 5% probability of failure for each robot, which requires dynamic task redistribution and the real-time adjustment of the manufacturing process.

The energy consumption for the robots varies between 40 W and 65 W, depending on the complexity and duration of the tasks performed, which highlights the importance of a balanced use of resources. Also, the time required to complete a task was set between 8 and 18 minutes, this variation being influenced by the specifics of the operational process and the requirements of each test scenario. The initial positions of the robots were defined for each simulation, optimising the distribution of tasks, reducing the risk of collision and improving the collaboration between units within the analysed system.

This testing method provides a clear perspective on how RSI influences the performance of the analysed system and the initial data used for testing the robots in each scenario is presented in Table 1.

Table 1. Typesetting specifications

Scenario	Robot	Task Execution Time [min]	Power Consumption [W]	Efficiency [%]	Initial Position (x, y)
Scenario 1 – Stable production line with sequential tasks	R1	10	50	80	(0,0)
	R2	12	45	78	(1,1)
	R3	15	55	75	(2,0)
Scenario 2 – Distributed manufacturing with dynamic task allocation	R1	9	48	82	(0,0)
	R2	14	52	76	(1,1)
	R3	16	58	73	(2,0)
	R4	8	50	85	(3,3)
Scenario 3 – Complex environment with limited resources and simulated faults	R1	11	55	78	(0,0)
	R2	15	60	74	(1,1)
	R3	18	65	70	(2,0)
	R4	9	53	83	(3,3)

To understand how well the proposed architecture works and which is the impact of Robot Swarm Intelligence (RSI) on distributed manufacturing processes, three main indicators were analysed that provide valuable information about how the resources are managed, how quickly tasks are completed, and how the system reacts to unforeseen problems.

The first indicator is the average task execution time, i.e. the time necessary for each robot to complete the tasks assigned to it. This parameter is important because it shows how efficient are the algorithms that distribute tasks and whether the system can maintain a stable and optimised production flow. A shorter execution time means a better coordination between robots, a smarter use of resources and a greater ability to adapt to the production needs.

The second indicator, the resource utilisation rate, shows how the available resources are managed, such as energy consumption and robot occupancy. A well-optimised system should distribute tasks in a balanced way, so that each robot is used at its optimal capacity, without consuming unnecessary energy or remaining idle. The last indicator, fault tolerance, is extremely important for the reliability of the tested architecture. This parameter measures how well the system can continue to function when problems arise, such as a robot failure. An intelligent system must be able to redistribute tasks automatically so that the production is not significantly affected.

It's possible to get a general idea of how the component parts and data flow in the RSI architecture work together by reading about them, but mathematical models are needed to fully understand and improve them. The efficiency of Robotic Swarm Intelligence (RSI), as a central element of this system, is based on mathematical algorithms that coordinate the robots' decisions and optimise the allocation of tasks.

The Big Data analytics component relates closely to the objective function, defined so as to minimise operational costs by reducing execution times and optimising resource utilisation. The following is the mathematical expression for this function:

$$F = \min \sum_{i=1}^{n} (C_{time}(R_i) + C_{resource}(R_i) \quad (1)$$

where C_{time} and $C_{resource}$ represent the costs associated with the execution time and resource usage for each robot R_i. Big Data Analytics processes data collected from sensors and robots to analyse these costs, identifying critical points and proposing adjustments. By using advanced algorithms, this component ensures a more efficient allocation of resources and the avoidance of bottlenecks in production.

A central element of this architecture is the coordination of robots through the Particle Swarm Optimization (PSO) algorithm, which allows them to make autonomous decisions and collaborate effectively.

The algorithm is implemented in the local processing components of each robot, and its dynamics are described in the following. A swarm of robots is initialised with random positions and velocities in the search space. Each robot i has a position vector x_i and a velocity vector v_i.

$$\begin{aligned} x_i(0) &= rand(x_{\min}, x_{max}) \\ v_i(0) &= rand(v_{\min}, v_{max}) \end{aligned} \quad (2)$$

The velocity of each robot is updated based on its own best-known position $pbest_i$ and the global best-known position $gbest$:

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (pbest_i - x_i) + c_2 \times r_2 \times (gbest - x_i) \quad (3)$$

where:

w is the inertia weight,

c_1 and c_2 are acceleration coefficients and

r_1 and r_2 are random numbers uniformly distributed in [0,1].

The position of each particle is updated based on its new velocity:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (4)$$

Thus, robots operate in real time, redistributing tasks in the event of delays or blockages, which contributes to a smoother production flow.

The calculation of the amount of resources used is necessary to ensure the sustainable operation of the system. The formula that describes this process is:

$$R_{used} = \frac{task\_time \cdot power\_consumption}{efficiency} \quad (5)$$

where *task_time* is the time required to perform a specific task, *power_consumption* is the power consumption of a robot and *efficiency* represents the operational efficiency of the robot.

These parameters are continuously monitored by sensors distributed throughout the industrial environment. The sensors provide real-time data to the IoT platform, which then forwards it to the analysis modules. This feedback loop ensures that potential problems, such as excessive energy consumption or low efficiency, are quickly detected and remedied.

Another fundamental aspect is the optimisation of the total production time, which is coordinated through the centralised data hub. The total time can be expressed as:

$$T_{total} = \max_{i=1,2,\ldots,n} (T_{start,i} + T_{task,i}) \tag{6}$$

This equation reflects the fact that the total duration of the production process is determined by the robot with the longest task. The centralised hub uses this data to reconfigure the task schedule, minimising waiting times and synchronising robot operations. In the final analysis, tests are performed using various classical test functions, and the outcomes are contrasted with those of the enhanced particle swarm optimisation techniques discussed in the related literature.

4. Results

The system performance analysis was carried out by evaluating the metrics defined in the experimental methodology, focusing on the task execution time, the resource utilisation rate, and fault tolerance. These three criteria allow a detailed interpretation of the system's ability to manage distributed tasks and to adapt to the dynamic requirements of a smart industrial environment. Fault tolerance, an essential aspect of any autonomous system, was analysed to determine to what extent the proposed architecture can maintain its performance in the face of errors or unforeseen operational variations. The correlation of these parameters provides an overview of the robustness and adaptability of the analysed system, highlighting its ability to operate efficiently under varied conditions.

The simulation results offer valuable insights into the system's behavior and its ability to adapt to scenarios of increasing complexity. In the first scenario, the resource utilisation rate reached 80%, indicating a reasonably efficient allocation of the available resources. The task completion times for both the MATLAB and ROS platforms were identical, each of them completing the task in 70 minutes. This suggests that the baseline performance of the two platforms is similar when operating under less challenging conditions. Additionally, the system demonstrated a 95% fault tolerance rate, highlighting its strong operational resilience and its capacity to maintain a consistent performance despite minor deviations or occasional errors.

As the complexity increases in the second scenario, the system demonstrates a clear adaptability. The resource utilisation rate increased to 85%, indicating a more sophisticated optimisation of internal processes to handle an increased demand. Interestingly, the task completion time remained identical for the two platforms, MATLAB and ROS, each of them completing the task in 80 minutes. This consistency in performance suggests a balanced capacity to process tasks, even under higher load conditions. The system's fault tolerance increased slightly, reaching 96%. This shows that it was getting better at keeping running without any problems, even when there were some small changes to how it worked.

In the third scenario, which is the most complex of the analysed scenarios, notable differences appear between the two platforms. The resource utilisation rate reaches 90%, indicating an almost complete management of the available resources, which reflects an advanced level of optimisation. Despite this, the task completion time begins to diverge between the MATLAB and ROS platforms. The MATLAB platform continues to show a linear increase, completing the allocated tasks in 90 minutes, while the ROS platform records a time of 100 minutes, which may point to difficulties in managing increased operational demands. This difference indicates that MATLAB is better equipped to cope with the increase in complexity, providing a predictable and stable performance. Fault tolerance, which reached 97%, remains a common strong point for both platforms, highlighting the overall robustness of the system.

Figure 2 provides an integrative perspective on the system's performance in the three scenarios. The obtained results suggest that the system, regardless of the platform used, manages to adapt effectively to more complex tasks, maximising the use of available resources. All scenarios feature a steady increase in the resource utilisation rate, reflecting a progressive improvement in operational efficiency as the system requirements become more demanding. In environments where the resources are finite, the linear trend indicates a high potential for continuous process optimisation. A high fault tolerance highlights the system's ability to maintain its functionality and performance even when errors or unexpected variations occur in the operational environment.

As the scenarios become more complex, the performance differences between MATLAB and ROS become more pronounced and are clearly reflected in the task completion times. In the first two scenarios, both platforms exhibit a similar behavior, indicating comparable levels of efficiency for tasks with moderate operational requirements. However, in the third scenario, the task completion time for ROS increases significantly in comparison with MATLAB. This dramatic increase may reflect limitations in the way ROS manages resources or optimises the workflow for more demanding tasks. This behavior may indicate a need to improve the resource scheduling or management algorithms used by ROS.

Overall, this combined graph contributes to a detailed understanding of the system behaviour and serves as a starting point for analysing how these platforms can be adapted and optimised for higher operational requirements. Thus, interpreting these trends provides a solid basis for informed decisions in the design and improvement of future similar systems.

5. Conclusion

This study evaluated the performance of a distributed manufacturing system using three basic metrics: the resource utilisation rate, task completion time, and fault tolerance. The simulation results for the three examined scenarios showed clear trends and significant differences between the MATLAB and ROS platforms. The system performance showed a progressive improvement with regard to the resource utilisation rate, featuring a constant increase from 80% in the first scenario to 90% in the third one. This indicates that the analysed system is able to adjust its resources to meet increasingly complex requirements.

Unlike earlier research that only looked at specific uses of PSO in areas like the placement of sensors in industrial processes or energy management, this article discusses how Industry 4.0 technologies and RSI algorithms can be used together in a more general way, focusing on the benefits and synergies that come from this. This integrated perspective contributes to a deeper

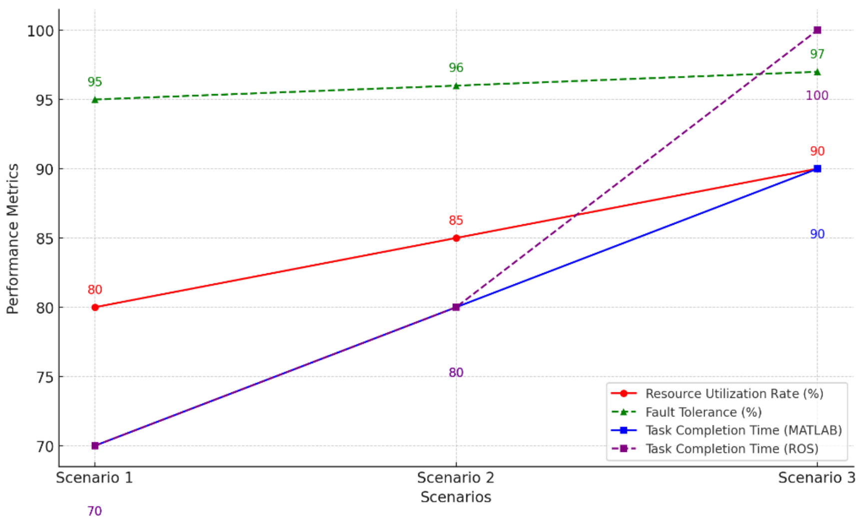

Figure 2. Simulation representation using the model based on control and prediction

understanding of the potential of combining these technologies for developing intelligent and adaptive manufacturing systems.

From a theoretical point of view, this study extends the understanding of the factors that influence the performance of operational systems based on MATLAB and ROS. In particular, this research highlights the importance of efficiently allocating resources in relation to the increasing demands and limitations of the analysed platforms for managing complex tasks. The observations related to task completion times suggest that the internal structure of these platforms directly influences how they respond to an increasing operational complexity.

The managerial implications of the obtained results are significant, especially for organisations implementing autonomous systems in dynamic environments. For example, MATLAB is more suitable for scenarios where stability and predictability are important, while ROS may be preferred for less critical applications where flexibility can compensate for a longer completion time.

However, there are certain limitations to this study that need further discussion. A potential impediment is the limited number of simulated situations, which might reduce the results' general applicability. A more thorough understanding of a system's behaviour under various circumstances may be obtained by broadening the scope of research to include a greater variety of situations. Second, this study only looked at three metrics, overlooking additional important factors like operational expenses and energy use.

Future improvements to this study could involve extending the analytical model to include additional dimensions, such as scalability and interoperability between platforms. Testing the system in real-world environments with a broader spectrum of operating conditions would also help validate the simulation findings. Additionally, comparisons with other technological platforms could shed light on how MATLAB and ROS perform relative to alternative solutions available on the market.

Improving resource management and reducing the task completion time through the creation of platform-specific optimisation algorithms could be another attractive topic for future study. This would imply that ROS has to implement better planning and resource allocation procedures to deal with complicated situations. On the other hand, possible directions for MATLAB's future development include making it more flexible and scalable in real-time settings.

REFERENCES

Abd Elfattah, M., Hassanien, A. E. & Abuelenin, S. (2019) A Hybrid Swarm Optimization Approach for Document Binarization. *Studies in Informatics and Control*. 28(1), 65-76. https://doi.org/10.24846/v28i1y201907.

Akopov, A. S. (2024) A Clustering-Based Hybrid Particle Swarm Optimization Algorithm for Solving a Multisectoral Agent-Based Model. *Studies in Informatics and Control*. 33(2), 83-95. https://doi.org/10.24846/v33i2y202408.

Alitappeh, R.J. & Jeddisaravi, K. (2022) Multi-Robot Exploration in Task Allocation Problem. *Applied Intelligence*. 52, 2189–2211. https://doi.org/10.1007/s10489-021-02483-3.

Bao, D.Q. & Zelinka, I. (2019) Obstacle Avoidance for Swarm Robot Based on Self-Organizing Migrating Algorithm. *Procedia Computer Science*. 150, 425–432. https://doi.org/10.1016/j.procs.2019.02.073.

Berman, S., Halász, Á. M., Hsieh, M. A., & Kumar, V. (2009) Optimized Stochastic Policies for Task Allocation in Swarms of Robots. *IEEE Transactions on Robotics*. 25(4), 927–937. https://doi.org/10.1109/TRO.2009.2024997.

Birattari, M., Ligot, A., Bozhinoski, D., Brambilla, M., Francesca, G., Garattoni, L., Garzón Ramos, D., Hasselmann, K., Kegeleirs, M., Kuckling, J., Pagnozzi, F., Roli, A., Salman, M. & Stützle, T. (2019) Automatic Off-Line Design of Robot Swarms: A Manifesto. *Frontiers in Robotics and AI*. 6, art. no. 59. https://doi.org/10.3389/frobt.2019.00059.

Chakraa, H., Guérin, F., Leclercq, E. & Lefebvre, D. (2023) Optimization Techniques for Multi-Robot Task Allocation Problems: Review on the State-of-the-Art. *Robotics and Autonomous Systems*. 168, art. no. 104492. https://doi.org/10.1016/j.robot.2023.104492.

Dorigo, M., Theraulaz, G. & Trianni, V. (2020) Reflections on the future of swarm robotics. *Science Robotics*. 5(49), art. no. eabe4385. https://doi.org/10.1126/scirobotics.abe4385.

Farivarnejad, H. & Berman, S. (2021) Multirobot Control Strategies for Collective Transport. *Annual Review of Control, Robotics, and Autonomous Systems*. 5, 205–219. https://doi.org/10.1146/annurev-control-042920-095844.

Gerkey, B.P. & Matarić, M.J. (2004) A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. *International Journal of Robotics Research*. 23(9), 939–954. https://doi.org/10.1177/0278364904045564.

Han, Y. & Song, T. (2024) Path Planning for Shop-Floor Material Transfer Robots Incorporating Particle Swarm and Ant Colony Algorithms. *Studies in Informatics and Control*. 33(3), 49-59. https://doi.org/10.24846/v33i3y202405.

Kuckling, J., van Pelt, V. & Birattari, M. (2022) AutoMoDe-Cedrata: Automatic Design of Behavior Trees for Controlling a Swarm of Robots with Communication Capabilities. *SN Computer Science*, 3(2), art. no. 136. https://doi.org/10.1007/s42979-021-00988-9.

Kuckling, J. (2023) Recent trends in robot learning and evolution for swarm robotics. *Frontiers in Robotics and AI*. 10, art. no. 1134841. https://doi.org/10.3389/frobt.2023.1134841.

Ligot, A. & Birattari, M. (2020) Simulation-only experiments to mimic the effects of the reality gap in the automatic design of robot swarms. *Swarm Intelligence*. 14, 1–24. https://doi.org/10.1007/s11721-019-00175-w.

Mendiburu, F.J., Garzón Ramos, D., Morais, M.R.A., Lima, A.M.N. & Birattari, M. (2022) AutoMoDe-Mate: Automatic off-line design of spatially-organizing behaviors for robot swarms. *Swarm and Evolutionary Computation*. 74, art. no. 101118. https://doi.org/10.1016/j.swevo.2022.101118.

Schranz, M., Umlauft, M., Sende, M. & Elmenreich, W. (2020) Swarm Robotic Behaviors and Current Applications. *Frontiers in Robotics and AI*. 7, art. no. 36. https://doi.org/10.3389/frobt.2020.00036.

Silva, F., Duarte, M., Correia, L., Oliveira, S. M. & Christensen, A. L. (2016) Open issues in evolutionary robotics. *Evolutionary Computation.* 24(2), 205–236. https://doi.org/10.1162/EVCO_a_00172.

Larabi-Marie-Sainte, S. & Ghouzali, S. (2020) Multi-Objective Particle Swarm Optimization-based Feature Selection for Face Recognition. *Studies in Informatics and Control*. 29(1), 99-109. https://doi.org/10.24846/v29i1y202010.

Trelea, I.C. (2003) The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection. *Information Processing Letters*. 85, 317–325. https://doi.org/10.1016/S0020-0190(02)00447-7.

Trianni, V. & Nolfi, S. (2011) Engineering the Evolution of Self-Organizing Behaviors in Swarm Robotics: A Case Study. *Artificial Life*. 17(3), 183–202. https://doi.org/10.1162/artl_a_00031.

Wang, W.J., Zheng, Y.T., Lin, G.Z., Zhang, L. & Han, Z.G. (2020) Swarm robotics: A review. *Robot*. 42, 232–256.

Xie, H., Sun, M., Fan, X., Lin, Z., Chen, W., Wang, L., Dong, L. & He, Q. (2019) Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation. *Science Robotics*. 4(28), art. eaav8006. https://doi.org/10.1126/scirobotics.aav8006.