Flexible Manufacturing System for Enhanced Industry 4.0 and Industry 5.0 Applications

Mihai-Daniel Pavel
Research and Development
Asti Automation
Bucharest, Romania
daniel.pavel@astiautomation.com

Grigore Stamatescu
Automation and Industrial Informatics
University Politehnica of Bucharest
Bucharest, Romania
grigore.stamatescu@upb.ro

***Abstract*—Flexible manufacturing systems (FMS) have the potential to increase efficiency and adaptability in manufacturing which results in both economic gains as well as support future environmental and social sustainability. However, the continuous optimization of such systems remains a challenging task due to the trade-offs between flexibility and production efficiency under increasing cost pressures. This work presents an enhanced design and development approach for flexible systems of fabrication that takes into account these trade-offs. Our approach is based on a combination of simulation and optimization techniques, and it has been validated through experiments on a real-world flexible system of fabrication with state-of-the-art components and tools for Industrial Internet of Things (IIoT) and Industry 4.0/5.0 paradigm integration. The presented results demonstrate the effectiveness of our proposed approach in conjunction with reference performance metrics such as the Overall Equipment Effectiveness (OEE) and the power consumption.**

***Index Terms*—flexible manufacturing system, efficiency, sustainability, industry 4.0, industry 5.0**

I. Introduction

The continuous evolution of technology and the development of smart manufacturing processes constantly bring upon new challenges for which the most innovative software solutions are needed. However, the optimization of such systems remains a challenging task, as it requires a balance between the flexibility of the system and its production capabilities. With the emergence of the concept of Industry 4.0, advanced technologies such as Artificial Intelligence [1], Internet of Things, Cloud Computing, Artificial Vision began to appear more and more often in automation solutions in manufacturing systems. Key elements include the accelerated adoption of pervasive and distributed monitoring and control systems and hierarchical data processing over multiple communication networks.

Industry 5.0, as defined by the European Union [2], supports the EU green and digital transition with a human centric approach that aims to bring the worker back at the core of the production system. It presents a complementary view to Industry 4.0, focusing on key terms such as sustainability and resilience. Sustainability refers to covering present needs while balancing growth imperatives with long term economic, social and environmental support of future generations. In particular, manufacturing resilience refers to the ability to overcome and recover from unexpected disruption of production, demand and supply, while assuring business continuity and meeting the needs of an extended group of stakeholders such as business owners, employees and customers.

Before such advanced technologies can be effectively integrated into industrial manufacturing systems, they must comply with the principles of Industry 4.0, more precisely to allow the interconnection of equipment, sensors, devices, this is done using specific communication protocols, allow information transparency and technical assistance, i.e. data collected and delivered from sensors and field equipment to support maintenance by visualizing the data in a simple and easy-to-understand manner. The last principle provides for decentralized decision-making, that is, the ability of systems to make decisions that solve problems autonomously.

In this work, we propose the optimized design and evaluation of a flexible systems of fabrication that takes into account the trade-offs between flexibility and production efficiency. Our approach is based on the integration of new technologies, both hardware and software, and it has been validated through extensive experiments on a real-world flexible system of fabrication. Hardware configurations options are discussed and justified, as well as the development of customised software components for image processing and manufacturing system management, through the integration of both proprietary/manufacturer and open source resources. The results of our study demonstrate the effectiveness of our proposed optimization approach in improving the performance of flexible systems of fabrication.

The rest of the paper is structured as follows. Section II illustrates several relevant publications that frame our current technical and scientific contribution within the state-of-the-art. Section III provides an in-depth description of the flexible production system including conceptual design, architecture, algorithms and methods. Section IV showcases the achieved results in terms of operational improvements and monitoring that result in increased overall performance. Section V highlights potential avenues for focused development.

This work was partially supported by the project Resource Efficient Manufacturing Model Development and Implementation (REFFICIENT), under the GreenSME program - Horizon Europe GA 101058613. Financial publication support from the University POLITEHNICA of Bucharest, through the PubArt program is acknowledged.

II. Related Work

Flexible manufacturing systems have gained significant attention in recent years due to their potential to increase efficiency and adaptability in production facilities and industrial enterprises. Many studies have focused on optimizing various aspects of flexible systems, such as their production capacity, energy consumption, and maintenance requirements. [3] showed the improvements on energy consumption of a flexible control and monitoring framework when using a distributed communication network.

One common approach to optimizing flexible systems is through the use of simulation and optimization techniques. These methods allow for the evaluation of different design configurations and operational strategies, enabling the identification of the optimal solution that meets the desired objectives. Some studies have employed mathematical programming methods, such as linear programming and mixed-integer programming, to optimize the allocation of resources in flexible systems. [4] presented the results of a simulation-based optimization for their manufacturing control system. Others like [5] and [6] have used heuristics and metaheuristics, such as genetic algorithms and simulated environments, to search for near-optimal solutions in complex and dynamic environments. In addition to simulation and optimization methods, various control and scheduling strategies have been proposed to improve the performance of flexible systems, just like [7] showed in their paper. These include decentralized control approaches, such as multi-agent systems and swarm intelligence, as well as centralized control approaches, such as model predictive control and discrete event systems.

[8], [9] address the problem of accurate object detection using several models including Deep Neutral Networks (DNN) and Convolutional Neutral Network (CNN). These models aim to use an optimal learning rate, minimizing the cost function, in order to achieve the lowest loss and the best accuracy and precision. [10] concluded that robots often face unfamiliar scenarios, leading to human intervention and implicitly wasting time, resources, and energy. So they proposed through their work the development of adaptive industrial robots using Machine Learning (ML)/Machine Vision (MV) tools, through which robots are able to make autonomous decisions during sorting or assembly operations based on color and/or shape of the test object. In previous work [11], a method to improve productivity of a robotic assembly cell through the YOLO framework for image processing and object detection was presented. This led to vision-based control of the robotic arm in the localisation and classification of assembled parts within the workspace.

Despite the progress made in the optimization of flexible systems, there are still many challenges to be addressed. These include the complexity and uncertainty of the manufacturing process, the integration of different optimization objectives, and the need for real-time adaptability to changing conditions.

III. Methodology

Two reference configurations of flexible assembly line systems are used as benchmark for the integration, deployment and validation of Industry 4.0/5.0 applications. These use industrial grade components and systems and are shown in Figure 1 and Figure 5.

Fig. 1: Flexible Assembly Line System (FAL) for Smart Manufacturing

A. Standard FMS Platform

The fabrication line represents a research and training platform which combines the fields of automation technology, sensor technology, drive technology, mechatronics and robotics in order to yield a fully integrated flexible system. The initial workflow of the system is modelled by means of a sequential state machine defined by the user on each of the five stations of the assembly line: work pallet storage, base piece storage, robot cell, top piece storage and final product storage. Each station is designed as a stand-alone unit with its own sensors e.g. inductive sensors, RFID modules, presence sensors, and drive system e.g. conveyor belt, electric motor and drive unit, all of them controlled by a Programmable Logic Controller (PLC) [12].

The whole process is sequential, well defined from the beginning, but nevertheless it can be improved in several ways. The main challenge with this sequential programming is the need for the industrial robot to search through the local warehouse for the parts to be used in the manufacture of the finished product. This greatly reduces the efficiency of the process, since it follows several predefined trajectories to perform the search, which can prove extremely inefficient in the situation where the robot, for example, does not have any parts in its local warehouse, it must checks all predefined positions before signaling that the desired product cannot be manufactured.

The machine vision subsystem is supplied by IFM and consists of a master video processing unit, model OVP800, based on NVIDIA Jetson TX2 hardware, that supports up to six O3R series 3D cameras connected simultaneously. The processing unit uses a proprietary operating system based on a Linux distribution, which allows the use of modern technologies for integration with industrial systems, such as the Robot Operating System 2 (ROS2) programming environment libraries and the Docker utility. Python, C++ and CUDA software development is also supported. For our application, this camera master serves as both an image collection device and an image processing device. The Docker virtualization software is used to create a customized programming environment that allows the integration of the system into the industrial application through a dedicated communication protocol for the Siemens PLC connectivity. The design choice was to keep the communication standard already used on existing devices, namely the PROFINET protocol from Siemens, implemented through the *python-snap7* library. The second reason we decided to use a Python library was to be able to use this library together with the ROS2 packages. Alternative options included Modbus TCP/IP communication protocol, the OPC Unified Architecture (OPC-UA) protocol, or even HTTP communication.

A challenge with the drive strategy used initially was the need for the robot to search for available parts in the local warehouse each time. A practical solution to this issue is to specify before starting the manufacturing process which parts are valid. This was achieved by placing one of the cameras above the warehouse and developing software to detect available parts.

The types of parts used in the robot handling stage have similar shapes, with various colors, illustrated in Figure 2a and Figure 2b. Industrial applications for the detection of colors, shapes or irregularities are a typical use case of Industry 4.0. The architecture of the IFM master is the same as that of the NVIDIA Jetson Nano (arm64) development board, which is important for the following steps because the IFM master does not have a graphical interface where we could work to view the images, nor does it have a very large storage to allow installation of all image processing packages.

(a) Local Storage Camera (b) Conveyor Camera

Fig. 2: Camera images

For testing and development we used one such Jetson Nano board, which we turned into a Node-RED server for remote monitoring and control of the robotic station (Figure 3). To process the images from the camera on the development board we used the ROS2 specific communication protocols, TCP/IP

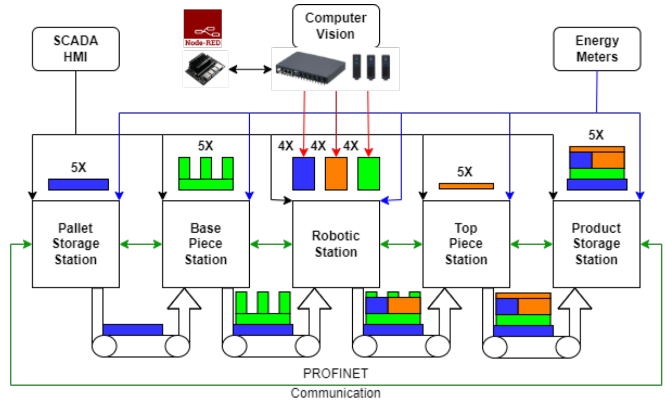


Fig. 3: FAL Flow

and UDP. Image retrieval is done automatically as long as both devices are connected to the same network. To make the connections of the devices to the robotic cell we used a wireless router with four LAN ports and one WAN port to connect the application to the Internet in case we need to perform system updates. The communication diagram is shown in Figure 4.

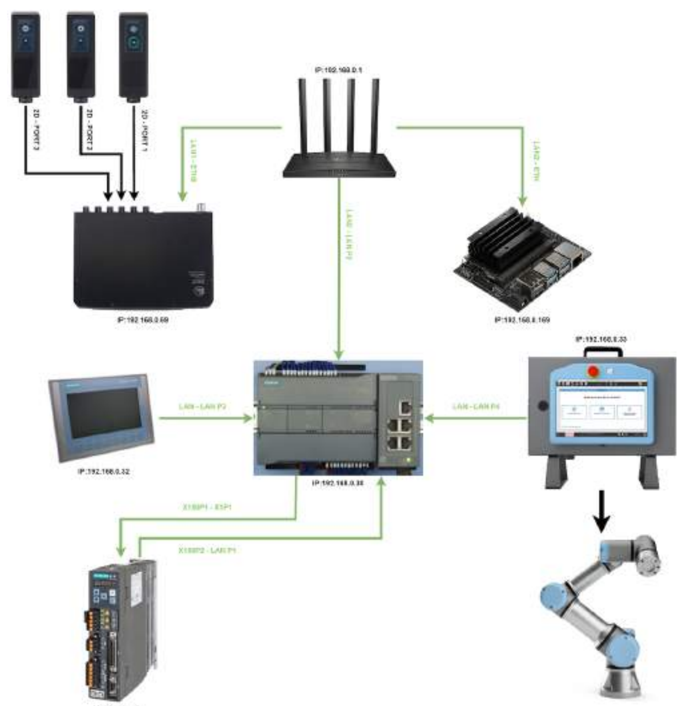


Fig. 4: System Network Architecture

The ROS2 development environment works through "Subscriber" and "Publisher" type entities that exchange information between processes. In our application, the main node will operate both as a Publisher, to activate and deactivate the cameras, and as a Subscriber, to retrieve the images from the OVP800 master to the ROS2 node. Also in this node, the communication with the SIEMENS PLC will be carried out, integrating the PROFINET communication library with the ROS2 development environment. Through this procedure, the PLC can send requests to activate/deactivate the cameras according to its needs. The warehouse camera is activated when the user configures their product to check the availability of parts in the warehouse, but also when the robot tried to process a part and it failed it will send a new request to try another available part. This exchange of messages between the robot and the camera master is done through the PLC since it is the core of the entire application and all devices are connected to it. The main loop of the robot communication routine is presented in Algorithm 1.

Algorithm 1 Main loop

Initiate ROS node
Create PLC communication client
Subscribe to camera topics
while Node running **do**
    Keep PLC communication active
    Read PLC requests
    Write PLC responses
    Operate the cameras
end while

Algorithm 2 presents the camera loop routine for basic processing of the images and identification for positioning of the collected parts.

Algorithm 2 Camera loop (each camera)

while Camera active **do**
    Read image from camera
    Process raw image
    Convert image to HSV color space
    Search colors within limits
    Filter noise values
    Check thresholds for each color
    Identify pieces in matrix
end while

A straightforward color detection algorithm was additionally implemented using the Python OpenCV library. The algorithm is designed to identify and isolate specific colors within an image, contributing to applications such as object recognition and tracking. Leveraging OpenCV's robust functionality for image processing and computer vision tasks, our algorithm uses color thresholds to segment regions of interest based on predefined color ranges. The implementation involves correcting the fisheye distortion effect, converting the BGR (Blue-Green-Red) input image to the HSV (Hue-Saturation-Value) color space, enabling more effective color representation and segmentation, with robustness to light condition changes. Through experimentation and optimization, we demonstrate the algorithm's efficiency in accurately detecting target colors in diverse environmental conditions. The simplicity and effectiveness of our approach make it a valuable tool for a wide range of computer vision applications, particularly those requiring rapid and reliable color identification.

The corresponding image correction and color correction steps over the HSV image space are implemented as follows:

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8 + k_5 r^{10})$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + k_4 r^8 + k_5 r^{10})$$

$$dist_{cf} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \end{bmatrix}$$
$$C_{mat} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = C_{mat} \times \begin{bmatrix} x_{distorted} \\ y_{distorted} \\ 1 \end{bmatrix}$$

$$R' = \frac{R}{255}; G' = \frac{G}{255}; B' = \frac{B}{255}$$

$$C_{max} = max(R', G', B'); C_{min} = min(R', G', B')$$

$$\Delta = C_{max} - C_{min}$$

$$H = \begin{cases} 0^\circ & \text{if } \Delta = 0 \\ 60^\circ \times (\frac{G'-B'}{\Delta} \mod 6) & \text{if } C_{max} = R' \\ 60^\circ \times (\frac{B'-R'}{\Delta} + 2) & \text{if } C_{max} = G' \\ 60^\circ \times (\frac{R'-G'}{\Delta} + 4) & \text{if } C_{max} = B' \end{cases}$$

$$S = \begin{cases} 0 & \text{if } C_{max} = 0 \\ \frac{\Delta}{C_{max}} & \text{otherwise} \end{cases}$$

$$V = C_{max}$$

$$H = H \times 360; S = S \times 100; V = V \times 100$$

Color detection algorithms face challenges such as varying illumination conditions, sensor noise, and color similarity. Therefore, it is important to evaluate the performance of color detection algorithms using appropriate metrics and experimental settings. We propose four performance indicators for color detection algorithms: detection error, response time, success rate, and a statistical analysis over different light intensities. Detection error measures the average distance between the detected color and the ground truth color in a color space. Response time measures the average time required to process an input image and output a color label. Success rate measures the percentage of images that are correctly classified by the algorithm. Statistical analysis over different light intensities measures how robust the algorithm is to changes in illumination, by comparing the performance indicators across different levels of brightness and contrast.

B. Enhanced FMS Platform

The CPS for Mechatronics and Industry 4.0 training equipment (Figure 5) emulates a highly automated factory, where different types of products are produced, packed and shipped. Its modular nature allows the user to configure the system that best suits their needs, also enabling staggered growth over time. Starting from an initial basic configuration, a subsequent expansion is possible, adding new work stations and, therefore, new technologies. The training equipment offers a professional learning experience that matches the industrial reality, where all the components are of industrial grade specifications. It is a flexible system that allows the user to customise certain parameters, such as the type of controllers used.

Fig. 5: Cyber Physical System (CPS) for Mechatronics and Industry 4.0

The equipment consists of five stations that carry out a different part of the assembly process as described in Figure 6:

1) Container feeding, checking and rejecting station: the function of this station is to feed containers to the system, which are stored in a gravity feeder. All containers have an RFID tag that allows them to be identified and traced throughout the whole process;
2) Filling feeding station: this station feeds basic fillings from a vertical store and inserts them into the product container. The quantity to be fed will be defined in the RFID tag according to the product being manufactured;
3) Lid sorting and feeding station: this station feeds the caps to an index plate where they are sorted (based on their color: white, black and blue), then it fits the cap for the product being manufactured, identified in the RFID tag, onto the container;
4) Final inspection and rejection station: this station carries out quality control of the product once assembled and rejects it if the result of the inspection is unsatisfactory. This process is made possible with the artificial vision camera system composed of IFM OVP800 master and O3R225 3D camera;
5) Robotic warehouse station: the function of this station is to store finished products using the robotic collaborative arm UR3e either in the local storage composed of 8 horizontal spaces or it can request support from a mobile robot platform to get the whole order and deliver it.

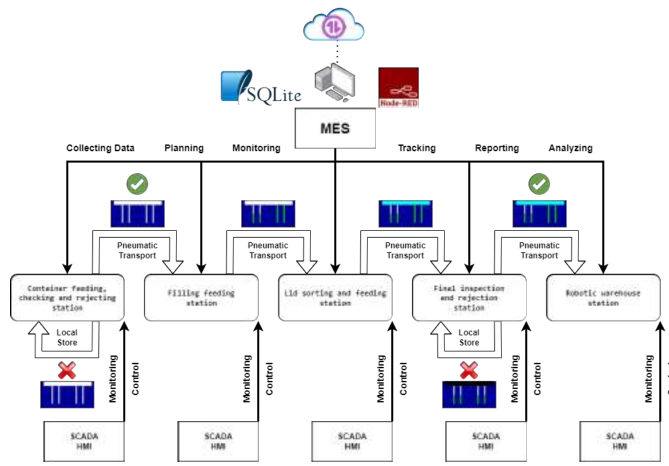


Fig. 6: Process flow diagram

This whole equipment was configured to work together with a manufacturing execution system (MES) which is layered between the supervisory level and the management levels in a typical automation hierachy as in Figure 7. MES solutions typically operate with big volumes of data, so that a standard database is always a necessity. For this application, SQLite database technology was used to perform tasks such as:

1) Data storage: MES needs a structured storage system to store vast amounts of real-time and historical data, generated during the manufacturing process;
2) Data retrieval: MES needs to provide real-time insights into the status of production and resource utilization;
3) Tracking and traceability: MES needs to track the movement of materials, raw components and products throughout the process;
4) Reporting and analysis: MES needs to generate reports and perform data analysis to help with the decision-making process, all to improve the overall efficiency;
5) Integration with other systems: MES can be often integrated with other enterprise systems such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM) or Quality Management Systems (QMS) and a database can enable bidirectional data exchanges;
6) Historical data for continuous improvement: MES can support improvement initiatives by analyzing trends and implementing improvement strategies.

The integration of MES with SQLite database was performed using the Node-RED environment as flexible and efficient solution for managing data. Node-RED serves as the orchestrator, facilitating communication between the MES and the SQLite database. Through a visual programming approach,

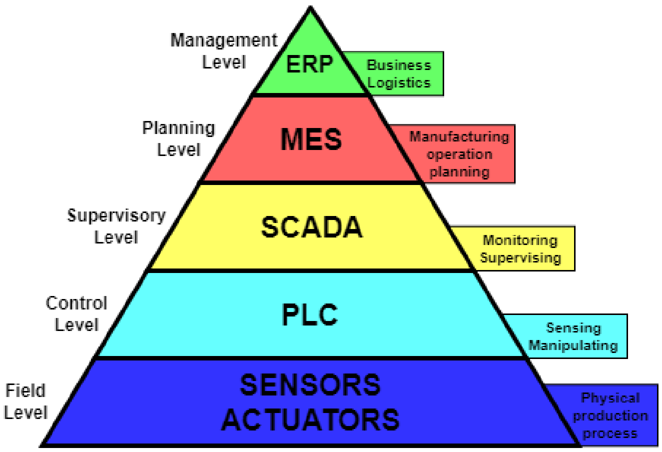


Fig. 7: Automation Pyramid

Node-RED allows for creation of intuitive and customizable workflows, as presented in [13], enabling the MES to interact with the database in real-time. To enable the application to interact with the industrial process we used OPC Unified Architecture protocol to exchange data between our MES and the field devices, namely the PLCs. This integration permits storing and retrieving critical production information, such as orders, production schedules and quality control data.

One of the typical Key Performance Indicators (KPI) in optimizing a fabrication process is the the overall equipment effectiveness (OEE) of the production process. OEE is a metric that reflects the availability (A), performance (P), and quality (Q) of the equipment and resources involved in manufacturing. A higher OEE value indicates a more efficient and productive system. However, calculating and monitoring OEE manually can be time-consuming, error-prone, and inconsistent. Therefore, integrating a manufacturing execution system (MES) application can be a valuable solution to automate and standardize the OEE measurement and analysis. A MES application can collect and store data from various sources, such as sensors, PLCs, ERP systems, and human operators, and use it to calculate the OEE and its components for each equipment, line, or plant. Moreover, a MES application can provide real-time feedback, reports, and dashboards to visualize the OEE and identify the causes and effects of downtime, waste, and defects. By using a MES application, the managers and operators of a flexible system of fabrication can gain more insight and control over the OEE and implement continuous improvement strategies to optimize the system performance and quality. OEE is computed as follows:

$$OEE = A \times P \times Q$$

$$Availability(A) = \frac{RunTime}{PlannedProductionTime}$$

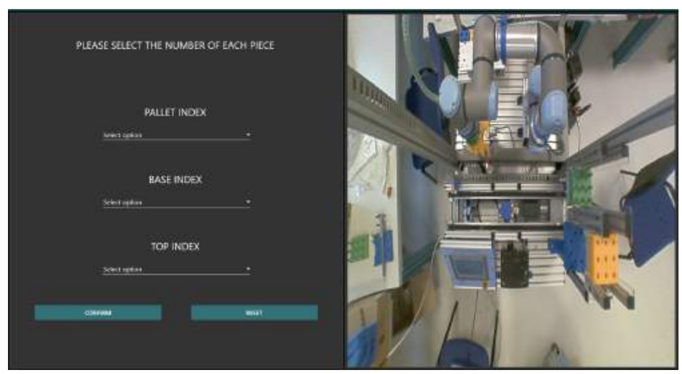
$$RunTime = PlannedProductionTime - StopTime$$

$$Performance(P) = \frac{IdealCycleTime \times TotalCount}{RunTime}$$

$$Quality(Q) = \frac{GoodCount}{TotalCount}$$

IV. Results

The results of the first optimization strategy were very satisfactory as we managed to achieve all the objectives we set for ourselves. We integrated the new IFM camera system into an industrial application, solved a manufacturing time optimization problem using artificial vision and image processing algorithms, used several modern technologies for opening the industrial system to IT platforms, and created new further development possibilities for such a system. The introduction of a secondary SCADA system for monitoring and control was made possible with the help of the Node-RED programming environment which locally launches a web server that can be accessed using any web search engine, even from a mobile phone. This server and the programs on it reside on the NVIDIA Jetson Nano development board running under the Docker virtual environment [6].

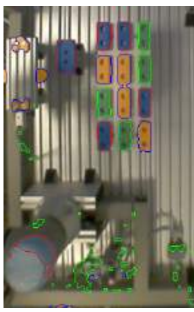


(a) Configure Commands Menu

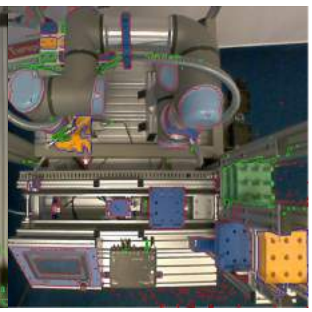
(b) Energy Monitoring Menu

Fig. 8: Node-RED Dashboard

The video monitoring and image processing application was developed and tested on the same NVIDIA Jetson Nano board, and after validating the results, the programs and source codes were transferred to the IFM camera master. To launch the application automatically (plug&play operation) the same Docker virtual environment was used with a custom image containing only the bare essentials, since the available memory on the IFM master is, as mentioned before, insufficient for all packages used in the application development stage. The created image contains the ROS 2 Humble development environment with C++ compiler and Python 3 packages, the OpenCV library for image processing, the IFM library of cameras that allows access to video streams from ROS, the industrial communication library on PROFINET Python-snap7, the ROS package for HTTP communication with the Node-RED server and our backend application.



(a) Local Storage Image



(b) Conveyor Image

Fig. 9: Processed Images

With the help of our image processing application, we have significantly reduced the total manufacturing time of the products, which implies a decrease in the total energy consumption and the degree of wear and tear of the equipment, but also an increase in the productivity of the automated system. We conducted experiments to evaluate the performance of the proposed strategy and compared it with the baseline scenario without the artificial vision system. The results are shown in Table I, which indicates that the proposed strategy can achieve an improvement of 15-20% in both production time and power consumption, demonstrating the effectiveness and efficiency of the artificial vision system for the optimization of the industrial manufacturing process.

Artificial Vision	No. Products	No. Pieces	Time(s)	Power (W)
Off	2	6	153	435.08
On	2	6	121	364.92
Off	4	12	293	830.04
On	4	12	232	693.43
Off	5	12	317	913.11
On	5	12	254	754.82

TABLE I: Assembly Process Efficiency

We also developed a color detection algorithm that can identify the parts used in the manufacturing process based on the colors orange, blue and green in an image. We varied the light intensity in the room to test the robustness of the algorithm. The results are shown in Table II, which reveals that the algorithm has a fast and consistent response time with an average of 3.3 ms for all colors, regardless of the light intensity. However, the detection error and success rate depend on both the color we are trying to detect and the light intensity in the room. The algorithm performs better on blue parts when the lights are on, with lower detection errors and higher success rates. On the other hand, the algorithm has more difficulties in detecting the blue parts when the lights are off, with higher detection errors and lower success rates. This suggests that the algorithm is more sensitive to the hue and saturation of the blue color than the other colors, and that the light intensity affects the color representation in the image.

Color	Detection Error[%]	Avg. Response Time [ms]	Success Rate [%]	Light Intensity
Orange	17.7	3.30	97	High
Green	14.3	3.29	99	High
Blue	11.5	3.20	100	High
Orange	18.5	3.43	88	Low
Green	22.1	3.21	91	Low
Blue	19.6	3.32	92	Low

TABLE II: Color Detection Performance

A representative example can be seen in Figure 10 where we chose the two stations to describe the utility of our application. We see in Figure 10a that the station is working properly with high Availability and Quality scores. However, there is room for improvement in the Performance factor, which indicates the speed of the production cycle. We propose to improve the Performance factor by increasing the working speed of the station. In Figure 10b we see good results in Performance and Quality but with lower scores in Availability. This is due to the fact that the station encountered technical problems that required its closure to carry out maintenance operations. Our application can detect these situations much faster and thus helps to reduce the maintenance time and increase the Availability factor. By implementing these improvements, we expect to achieve higher OEE scores and optimize the industrial process of manufacturing.

V. Conclusion

The work described the integration several technologies, from the machine vision subsystem for image processing to the remote monitoring and control application developed on the NVIDIA Jetson Nano board running the Node-RED server together with the MES application, in an industrial manufacturing process. All this was achieved using advanced IT technologies, Docker platforms, Robot Operating System 2 environment, Python libraries for industrial communications and image processing algorithms, Node-RED environment for back-end and front-end applications together with SQLite database for a complete MES integration. The MES seamlessly captured and organizes data related to production orders, resource utilization and quality control, offering a comprehensive overview that empowers decision-makers with timely and accurate information. With this MES application

(a) Container feeding, checking and rejecting station

(b) Final inspection and rejection station

Fig. 10: OEE Indicators

we reduced the downtime and enhanced the overall equipment effectiveness (OEE).

Our proposed solution to optimize manufacturing processes, demonstrated on two pilot configuration of flexible manufacturing systems, has brought considerable improvements to the automated systems, from decreasing manufacturing time and energy consumption to opening up further developments. A further application would be to scale up the application and introduce it to full-scale industrial environments to evaluate the impact on mass production. In addition, further research can focus on the development of machine learning algorithms to expand the range of applications of the artificial vision system. To extend the color detection algorithm a further study can incorporate a shape detection algorithm for a comprehensive framework for object recognition and characterization. By combining the color information obtained through the initial color detection algorithm with shape analysis, we enhance the system's ability to discriminate and identify objects based on both color and geometric features allowing for more precise delineation of object boundaries [14] and enables a fully automated application of quality control with fault detection in the manufacturing process.

ACKNOWLEDGMENT

The timely and substantial support of the company Asti Automation in providing high quality and industrial grade equipment for the development and evaluation of the approaches presented in this work is gratefully acknowledged.

REFERENCES

[1] A. Micheli and A. Valentini, "Synthesis of search heuristics for temporal planning via reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, pp. 11 895–11 902, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17413

[2] E. Commission, D.-G. for Research, Innovation, A. Renda, S. Schwaag Serger, D. Tataj, A. Morlet, D. Isaksson, F. Martins, M. Mir Roca, C. Hidalgo, A. Huang, S. Dixson-Declève, P. Balland, F. Bria, C. Charveriat, K. Dunlop, and E. Giovannini, *Industry 5.0, a transformative vision for Europe – Governing systemic transformations towards a sustainable industry*. Publications Office of the European Union, 2021.

[3] S.-H. Nam, K.-R. Ryu, K.-E. Kwon, S.-W. Lee, and H.-Z. Choi, "Architecture of flexible and reliable control & monitoring system for distributed machining systems," 05 2008, pp. 205–208.

[4] B. Shahbazi and S. Rahmati, "Developing a flexible manufacturing control system considering mixed uncertain predictive maintenance model: a simulation-based optimization approach," *Operations Research Forum*, vol. 2, 10 2021.

[5] G. Candan and H. Yazgan, "Genetic algorithm parameter optimization using taguchi method for a flexible manufacturing system scheduling problem," *International Journal of Production Research*, vol. 53, 07 2014.

[6] U. Umar, M. k. a. Mohd ariffin, N. Ismail, and S. Tang, "Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (agv) in flexible manufacturing systems (fms) environment," *The International Journal of Advanced Manufacturing Technology*, vol. 81, 06 2015.

[7] J. Jerald, P. Asokan, G. Prabaharan, and S. R, "Scheduling optimisation of flexible manufacturing systems using particle swarm optimisation algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 25, pp. 964–971, 05 2005.

[8] C. Deac, G. Deac, R. Parpala, C. L. Popa, and C. Cotet, "Vibration anomaly detection using deep neural network and convolutional neural network," *International Journal of Modeling and Optimization*, pp. 19–28, 02 2021.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[10] *Adaptive Industrial Robots Using Machine Vision*, ser. ASME International Mechanical Engineering Congress and Exposition, vol. Volume 2: Advanced Manufacturing, 11 2018. [Online]. Available: https://doi.org/10.1115/IMECE2018-86720

[11] S. Rosioru, G. Stamatescu, I. Stamatescu, I. Fagarasan, and D. Popescu, "Deep learning based parts classification in a cognitive robotic cell system," in *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*, 2022, pp. 403–408.

[12] M.-D. Pavel, S. Roșioru, N. Arghira, and G. Stamatescu, "Control of open mobile robotic platform using deep reinforcement learning," in *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*, T. Borangiu, D. Trentesaux, and P. Leitão, Eds. Cham: Springer International Publishing, 2023, pp. 368–379.

[13] R.-A. Luchian, S. Rosioru, I. Stamatescu, I. Fagarasan, and G. Stamatescu, "Enabling industrial motion control through iiot multi-agent communication," in *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 2021, pp. 1–6.

[14] L. Fernandes and B. Shivakumar, "Identification and sorting of objects based on shape and colour using robotic arm," in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 866–871.