Optimizing IoT: Integrating Smart-Sensor Data Fusion on Mitsubishi RV-4FRL Robot Arm with Data Processing and Visualization

Conference Paper · June 2024

DOI: 10.1109/ICCCNT61001.2024.10725339

6 authors, including:

Achirangshu Patra
C. V. Raman College of Engineering

Jasaswinee Sahoo
C. V. Raman Global University

Sk Abdul Sajeed
C. V. Raman College of Engineering

Optimizing IoT: Integrating Smart-Sensor Data Fusion on Mitsubishi RV-4FRL Robot Arm with Data Processing and Visualization

Aaditya Kumar Pradhan
Department of CSE
C.V.Raman Global University
Bhubaneswar, Odisha
0009-0004-6658-4667

Achirangshu Patra
Department of EE
C.V.Raman Global University
Bhubaneswar, Odisha
0000-0002-6126-3459

Jasaswinee Sahoo
Department of CSE
C.V.Raman Global University
Bhubaneswar, Odisha
0009-0000-4449-2850

SK Abdul Sajeed
Department of CSE
C.V.Raman Global University
Bhubaneswar, Odisha
0009-0005-6659-3704

AVS Sai Teja
Department of CSE
C.V.Raman Global University
Bhubaneswar, Odisha
0009-0003-8622-133X

Yashika
Department of CSE
C.V.Raman Global University
Bhubaneswar, Odisha
0009-0000-8553-3110

Abstract— **The Internet of Things (IoT) represents an interconnected system comprising smart objects embedded with sensors, networking capabilities, and processing technologies. These elements collaboratively integrate to create an environment where smart services seamlessly reach end-users. This paper presents a pioneering architecture for an Internet of Things (IoT) application for industrial use, focusing on condition monitoring and predicting maintenance needs. The framework integrates Python for data collection from a Bosch Rexroth CISS (Connected Industrial Sensor Solution) multi-sensor and utilizes a specialized switch mechanism within Node-RED, allowing topic-based categorization like Accelerometer for selective sensor activation or deactivation. The gathered data is systematically stored in a MySQL database, following predefined conditions and criteria. Utilizing our system, we gather vibration data from the Mitsubishi RV-4FRL Robot Arm, analyze J3 axis vibrations through Python-based Fast Fourier Transform (FFT) for predictive maintenance insights, and store temperature data in MySQL database based on predefined conditions.**

Keywords— Internet of Things, Node-RED, Smart Sensor, Robot Arm, MySQL, Predictive Maintenance, Condition Monitoring, FFT.

I. INTRODUCTION

The escalating demands of Industry 4.0 have ushered in a new epoch of technological advancement, marked by the integration of smart data analytics and predictive maintenance strategies. At the heart of this evolution lies the imperative challenge of ensuring the seamless functionality of industrial machines, particularly the pivotal role of industrial robots in modern manufacturing ecosystems. As these robots are increasingly embedded within production lines, their operational health becomes critical, with any breakdown potentially causing substantial disruptions and financial losses due to unanticipated downtimes.

In Industry 4.0, with AI and IoT merging, predictive maintenance is crucial. It digitizes processes, especially maintenance, offering benefits like less downtime, better equipment performance, reduced expenses, higher asset returns, risk mitigation, and fostering profitable growth. This proactive approach revolutionizes industry strategies. Using specialized sensors and machine learning, predictive maintenance provides real-time insights into asset health, enabling operators to optimize schedules, reduce downtime, and maximize efficiency.

The significance of predictive maintenance systems that harness internal signals from robots has garnered significant attention, offering a viable solution to pre-emptive problem-solving. However, existing algorithms often encounter limitations, notably in their adaptability to changes in robot operations, leading to the generation of false alarms. This predicament necessitates recurrent model retraining with new operational data, hampering sustained accuracy and efficiency.

In response to these challenges, our paper introduces an innovative architecture meticulously designed to address these limitations. This architecture is strategically crafted to proficiently collect sensor data, subject it to comprehensive analysis, apply selective filtration techniques, and subsequently visualize the processed information. This integrated framework serves as a robust solution applicable to condition monitoring and predictive maintenance strategies, specially tailored for the intricate needs of industrial robots.

Furthermore, a pivotal demonstration within our work showcases this architecture's practical application in the arm robots domain. By exemplifying the collection and utilization of sensor data from an arm robot, we illuminate the adaptability and efficacy of our framework in a real-world industrial setting. This demonstration serves as a testament to the versatility and reliability of our proposed architecture in augmenting predictive maintenance practices specific to robotics services.

II. RELATED WORKS

In today's manufacturing industry, rapid changes are driven by new technology. With the use of IOT, we can enhance manufacturing processes and numerous researchers are contributing in this field.

Researchers have developed Memsio[1], an open-source, battery-powered MEMS accelerometer for IoT. It's a versatile wireless sensor unit operated via web browsers, enabling remote accessibility through smart devices. Memsio boasts high-speed motion data acquisition, extended storage, and easy integration for diverse condition monitoring applications. Within this study's scope, an innovative data aggregation scheme (EDAS)[2] is proposed for IoT-based Wireless Sensor Networks (WSN). This scheme incorporates an enhanced low-energy adaptive clustering algorithm (I-LEACH) to optimize cluster head formation and employs network coding, using linear XOR operations, to eliminate data redundancy for non-replicated transmissions. Exploring improvements in IoT system development, [3] this research focuses on enhancing Node-RED, a visual programming tool. By tackling limitations in observability and debugging, the study aims to minimize development time and deployment failures. Addressing the energy constraints in battery-powered wireless sensors for IoT applications,[4]

This paper explores a hierarchical data aggregation method to optimize data transmission, crucial for extending network lifetime in WSNs. Exploring the integration of IoT in Industry 4.0, [5] this paper emphasizes the pivotal role of advanced technology, such as MQTT communication protocols and IBM's Node-RED, inefficiently storing, analysing ,and visualizing data from industrial systems, providing valuable insights for further development. Amidst the surge in research attention toward fault diagnosis in electrical machines,[6] this study leverages IoT and industrial wireless sensor networks to enable early fault detection, proposing an oriented sport vector machine (FO-SVM) algorithm. This algorithm extracts pertinent features, bolstering fault classification accuracy and suitability for cloud-based platforms, thus advancing industrial automation. In the realm of interconnected devices within the Internet of Things (IoT), this initiative focuses on expanding device control through Alexa Voice Service, streamlining connectivity across diverse IoT platforms. Utilizing Node-Red, it simplifies device connections and optimizes firmware updates by offloading data processing for efficient setup and maintenance[7]. This paper gives us efficient and precise methods to control the robotic arm which allows for less consumption of energy[8].In modern factories, robots handle assembly and transport of the products, this user-centred design allows user to specify input and output which allows for human supervised accuracy[9]. In this paper the system is connected to internet through PLC and IOT which gives us control remotely through a secure platform[10].

In past, research was mainly focused on a specific field applications often using single or limited purpose sensor. In our paper we used compact sensor which gives us a significant advancement in this area. These multi-purpose sensor are adaptable and are easily applicable in a wide range of applications. Also, our model has integrated a user friendly interface via Node-Red. Which gives users a button to select and utilize a specific sensor data, enhancing both the flexibility and usability of sensor system.

III. PROPOSED ARCHITECTURE

A. *Data Acquisition*

Python provides an interface to retrieve data from CISS multi-sensor[11] devices manufactured by Bosch Rexroth. It offers multiple operational modes like Raw Data Streaming, different kHz Accelerometer Modes, Time Aggregation Mode, and Event Detection Mode, all configurable via user-defined settings in a setup file. Python establishes a communication channel with the smart sensor via a COM port and initializes various sensor modules based on pre-specified modes, sampling rates, and thresholds. It employs various sensors including accelerometer, magnetometer, gyroscope, temperature, pressure, humidity, light, and aquatics—for either streaming data or identifying events based on pre-set criteria.

The code structure is organized into sections, starting with sensor configurations, managing data streaming or event detection, and handling log files. Core functionalities involve establishing a connection with the CISS device, selectively activating or deactivating sensors based on operational modes, configuring sensor sampling rates, and setting event detection thresholds. The script follows a modular approach, encapsulating functionalities into methods, and optimizing code readability and maintenance. It ensures accurate data handling by parsing payloads and logging information into individual CSV log files corresponding to each mode.

B. *Dynamic Sensor Data Processing*

Node-Red,[12] [13] a web-based framework chosen for this application due to its wide compatibility across various platforms, has gained considerable prominence in the realm of industrial automation systems. Because of its cross-platform availability, has gained significant traction in managing and controlling industrial devices. Its adoption as an IoT flow-based programming tool within industries has notably surged, establishing itself as a pivotal solution for developing Industrial IoT applications. This tool is popular for its user-friendly graphical development environment. It simplifies industry connectivity through two fundamental aspects. Firstly, it offers an extensive library of pre-built nodes that facilitate developers in executing complex tasks effortlessly within their applications. Secondly, Node-RED's graphical interface enables the placement of nodes on a canvas, connecting them to perform specific functions—such as data collection, processing, or transmission. In Fig 1, the Node-RED nodes employed for sensor data processing are depicted.

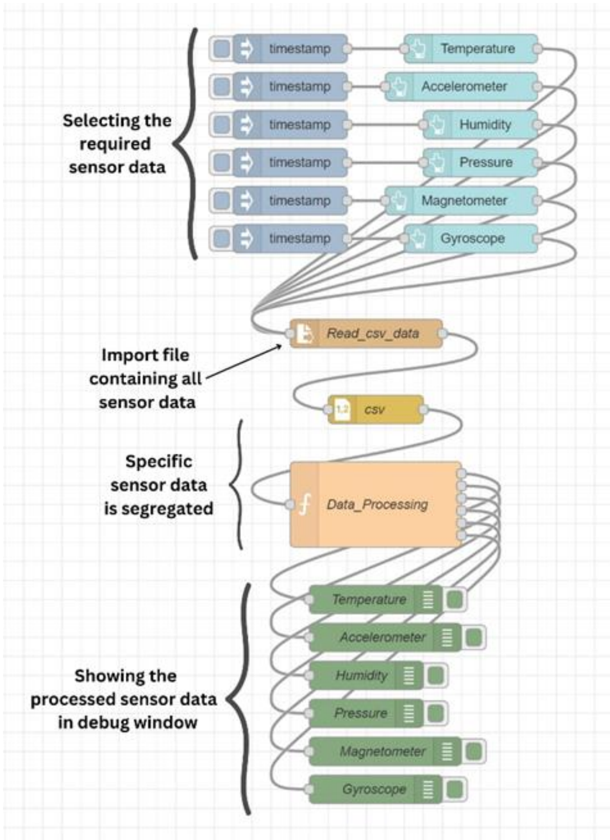

Fig. 1: Node-Red flow for CISS sensor data processing

The Node-RED flow orchestrates a sequence of nodes to handle various parameters of the CISS smart sensor. UI buttons represent different sensor types such as temperature, accelerometer, humidity, pressure, magnetometer, and gyroscope. These buttons trigger the reading of data from a CSV file through a file input node, followed by parsing the CSV data to extract information. A function node, filters and organizes the data based on the input. For instance, it rearranges data for accelerometer readings into x, y, and z-axis values, extracts specific data for temperature, humidity, and pressure, and formats magnetometer and gyroscope data accordingly.

IV. Proposed Architecture Integration

In our research focused on the Internet of Things (IoT) applications, we've implemented a novel approach utilizing our custom-developed Node-RED flow coupled with a Python interface. We've established a seamless connection enabling the extraction of accelerometer data from the CISS sensor interfaced with a Mitsubishi robot[14]. Specifically, our implemented architecture enables the continual acquisition of accelerometer data [15] over defined periods. In Fig 2, the illustration displays the attachment of the CISS sensor to the J3 axis of the robot, a crucial component utilized in our experiment.

Fig. 2: Mitsubishi RV-4FRL Robot Arm

Another Node-Red flow Fig. 3 is designed to read data from a CSV file, process it, filter out specific sensor values (like accelerometer data), and potentially store this filtered data in a CSV file.

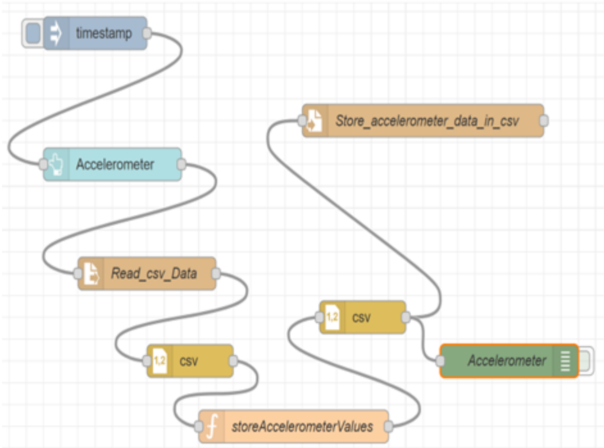

Fig. 3 Node-RED flow depicting a data processing sequence for reading, filtering, and storing accelerometer data from a CSV file

A. *Accelerometer Analysis Using Python*

Using Python, we implemented a comprehensive data analysis process to examine accelerometer data extracted from a CSV file .

Algorithm 1: Pseudo-code of Vibration Analysis

Result: Analysis of Vibration Characteristics

1. Input: Accelerometer data (X, Y, Z axes), Sampling rate = 100Hz

2. Data Preparation:

- **a.** Load accelerometer data from CSV file.
- **b.** Convert milli g to g if applicable.

3. Time Calculation:

- **a.** Determine time intervals based on the sampling rate.
- **b.** Generate a time array for overall data analysis.

4. Statistical Analysis for Overall Data:

- **a.** Calculate descriptive statistics (mean, std, min, max, etc.)
- **b.** Display overall data statistics

5. Grouping and Analysis:

- **a.** Group data by 'weight' and 'speed'
- **b.** For each group:
  - **i.** Calculate the time interval specific to the group
  - **ii.** Generate group-specific time array
  - **iii.** Perform statistical analysis for the group
  - **iv.** Plot time-domain signals for the X, Y, and Z axes
  - **v.** Perform FFT analysis for the X-axis
  - **vi.** Optionally, save plots based on 'weight' and 'speed'
  - **vii.** Display group-specific statistics

End Algorithm

B. *Fast-Fourier transform*

The Fast Fourier Transform (FFT)[16] algorithm used in the code, is represented as follows:

Given a discrete sequence x[n] of length N, the Fast Fourier Transform (FFT) X[k] for frequency index k is calculated using the formula:

$$x[k] = \sum_{n=0}^{N-1} x\,[n] \cdot e^{-\frac{i2\pi kn}{N}}$$

Where:

- x[n] represents the discrete input signal at time index n.
- X[k]is the resulting discrete frequency spectrum at frequency index k.
- N denotes the length of the input sequence.
- *i* is the imaginary unit.
- e is the base of the natural logarithm raised to the power of the argument.
- k and n iterate through the time and frequency indices respectively

This formula represents the essence of how the FFT computes the frequency components of a discrete signal, enabling the transformation of a time-domain signal into its corresponding frequency-domain representation.

C. *Automated Temperature Logging*

We devised a systematic approach within Node-RED by employing function blocks and interconnected nodes to segregate temperature data from other sensor data[17], subsequently saving it into a distinct CSV file. This process, integrated with Python, enables automatic storage of updated temperature values and their respective timestamps whenever alterations occur. Through this streamlined configuration, Node-RED precisely identifies temperature fluctuations and seamlessly logs these changes into an SQL database[18] in real time. In Fig. 4 This method ensures that crucial temperature variations are efficiently captured and chronologically stored for future analysis or reference purposes.

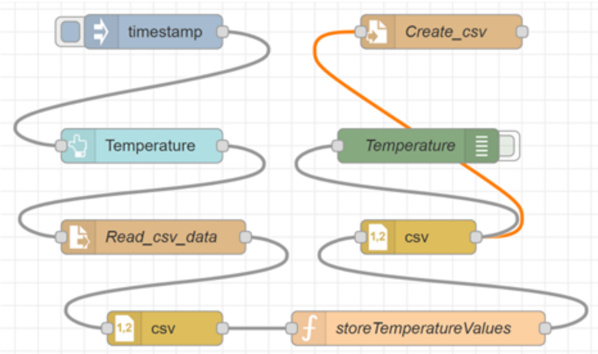

Fig. 4: Node-RED Configuration for Temperature Data Segregation and Logging

V. Data Description

We acquired data using a CISS sensor while configuring the Mitsubishi RV-4FRL to follow a predetermined path and cycle consistently. Our data collection involved testing different loads of 500, 1000, 1500, 2000, and 2500 grams, each tested at three distinct speeds: 10%, 30%, and 50% override speed. This systematic procedure allowed us to compile extensive datasets covering various load conditions and different operational speeds. In Fig. 5, the weights utilized for these experiments are displayed. Due to the limitation of our experimental setup, we have used up to 2.5kg load and 50% of the override speed of the Mitsubishi Electric MELFA Robot.

Fig. 5: Various Loads (500g, 1000g, 1500g, 2000g, 2500g) Tested for Data Collection.

VI. Experimental Setup

In this experiment, our setup encompassed a system with an Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz and 8.00 GB RAM. Coding was executed using Python version 3.7.5 and Node-RED version 3.1.3. The Mitsubishi MELFA RV-4FRL arm robot was programmed using the RT Toolbox 3 Mini version 1.20 software. Additionally, the setup integrated a CISS sensor from Bosch Rexroth for comprehensive data acquisition and analysis, coupled with MySQL Server version 8.0.35 and Connector/Python 8.0.33 for enhanced database functionality and connectivity

VII. Result Analysis

In practical conditions, operational systems are often distributed and inaccessible for extended periods. Detecting abnormal parameters during machine health monitoring can significantly disrupt system stability or cause permanent damage. Hence, in today's era of automation, the critical importance of condition monitoring cannot be overstated. It forms the basis for maintenance actions by identifying vital health indicators over time.

Our focus is on implementing an effective architecture applicable to industry needs. Our architecture integrates direct and indirect monitoring sources within our model. To streamline this process without frequent program alterations, we've integrated Node-RED's interactive dashboard. This enables real-time visualization, analysis, and informed decision-making. Our approach involves employing tools like fast Fourier transform for data analysis and Python programming for vibration data assessments.

The interactive Node-RED dashboard allows flexible sensor activation or deactivation. We effortlessly convert time-domain data into the frequency domain, facilitating comprehensive investigations via the intuitive dashboard interface.

Furthermore, our model includes a historical performance log for sensor data operations, enabling uncertainty analysis through machine learning and deep learning models. Additionally, our system allows conditional data storage in the database, preventing unnecessary overflow.

In Fig. 6, the weight and speed parameters are initially retrieved from the dashboard through dropdown nodes, storing them as globally accessible variables using two function nodes. Upon pressing the 'SHOW' button on the dashboard, these global variables (weight and speed) are retrieved, and a file path is dynamically constructed by incorporating these variables into the appropriate string locations. This constructed path is then forwarded to a 'read file' node, accessing the specified file and passing it through a 'base64' node. The 'base64' node encodes the image data into the payload, subsequently utilized by a function template to create an image object. Finally, this processed data reaches the dashboard template, enabling its display through HTML/Angular templates. Additionally, a debug node is connected to verify the accuracy of the constructed file path.

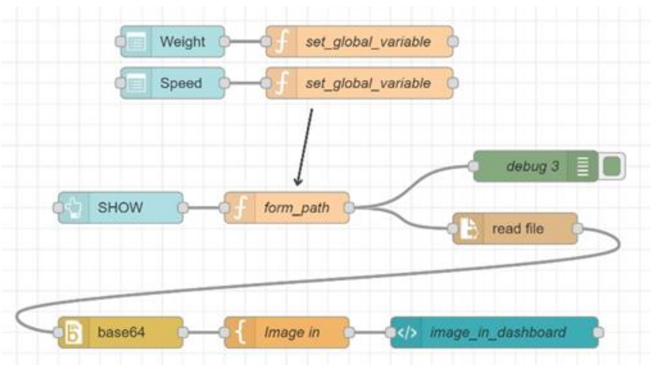

Fig.: 6 Node-RED flow illustrates the seamless integration of sensor data analysis from Python programming into Node-RED, facilitating the selection of specific data analyses for streamlined IoT application development.

For operators, conducting real-time investigations can be challenging. To address this, we've demonstrated how effortlessly one can apply mathematical models and visualize parameter patterns using the Node-RED dashboard. Additionally, due to the prevalent use of flexible manufacturing systems in today's industries, it's often impractical to make core programming changes to meet specific industry requirements.

In Fig 7, specific accelerometer readings for three different axes are depicted. After the vibration calculations, the Fast Fourier Transform (FFT) is applied to the vibration data. This facilitates the visualization of the FFT graph, accessible by selecting a specific weight and overriding the speed of the 6-axis robot.

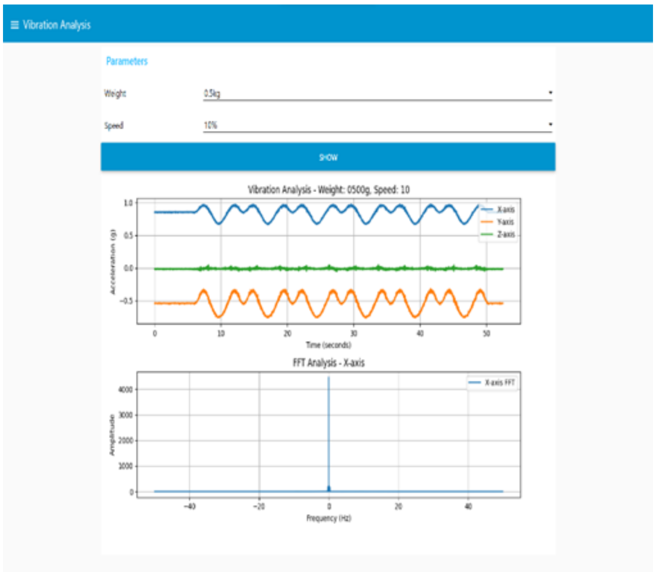

Fig.: 7 A Node-RED user-friendly dashboard displaying FFT analysis of vibration data, providing a visual representation of vibration frequencies and patterns for comprehensive analysis and diagnostics.

One notable feature of our model is its capability to record sensor data for a specific duration. Additionally, we can monitor individual data in real-time on the Node-RED dashboard. Fig 8 illustrates the dashboard we have developed to showcase this functionality.

Fig.:8 Node-RED dashboard displaying real-time temperature, humidity, pressure, and accelerometer data.

VIII. FUTURE WORKS

Future developments include the creation of a sensor logging app-based control system, aimed at enhancing data management and control capabilities. Additionally, plans involve implementing a feature to send sensor data as comprehensive reports via email or WhatsApp for convenient and efficient sharing and analysis. Construction of a WiFi module that enables users to subscribe to specific sensor data. This functionality aims to offer customizable access to selected sensor information, enhancing flexibility and usability within the system

.

IX. CONCLUSION

Our implementation is crafted to tackle industry-specific challenges by streamlining monitoring processes, seamlessly adapting to industry requirements, and predicting or preventing potential faults or failures. Moreover, it utilizes open-source software, making it cost-effective and beneficial for low-cost implementation in small-scale industries. However, it still presents a risk in terms of security. Given, that small scale industries do not connect with large networks as multinational level industries. So, we have considered the risk to be manageable and neglected the security analysis in this paper. Future, research on this should address all this security concerns especially if the model is considered to be scaled to a larger or more interconnected networks.

REFERENCES

[1] Koene, Ivar, Ville Klar, and Raine Viitala. "IoT-connected device for vibration analysis and measurement." HardwareX 7 (2020): e00109.

[2] Badiger, Vani S., and T. S. Ganashree. "Data aggregation scheme for IOT based wireless sensor network through optimal clustering method." Measurement: Sensors 24 (2022): 100538.

[3] Torres, Diogo, et al. "Real-time feedback in node-red for iot development: An empirical study." 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT). IEEE, 2020.

[4] Chakraborty, Ishita, Anannya Chakraborty, and Prodipto Das. "Sensor selection and data fusion approach for IoT applications." Recent Developments in Machine Learning and Data Analytics: IC3 2018. Springer Singapore, 2019.

[5] Ferencz, Katalin, and József Domokos. "Using Node-RED platform in an industrial environment." XXXV. Jubileumi Kandó Konferencia, Budapest (2019): 52-63.

[6] Zhang, Xiaoran, et al. "Research on vibration monitoring and fault diagnosis of rotating machinery based on internet of things technology." Nonlinear Engineering 10.1 (2021): 245-254.

[7] Rajalakshmi, Anoja, and Hamid Shahnasser. "Internet of Things using Node-Red and alexa." 2017 17th International Symposium on Communications and Information Technologies (ISCIT). IEEE, 2017.

[8] Benotsmane, R. and Kovács, G., 2023. Optimization of energy consumption of industrial robots using classical PID and MPC controllers. Energies, 16(8), p.3499.

[9] [9] Yang, Yuyi, et al. "Human Robot Collaboration in Industrial Applications." 2023 9th International Conference on Virtual Reality (ICVR). IEEE, 2023

[10] Thangatamilan, M., et al. "Industrial IoT (IIoT) based Control of Robotic Arm integrated with PLC and HMI." 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS). IEEE, 2023.

[11] Tnani, Mohamed-Ali, Michael Feil, and Klaus Diepold. "Smart data collection system for brownfield CNC milling machines: A new benchmark dataset for data-driven machine monitoring." Procedia CIRP 107 (2022): 131-136.

[12] Lekić, Milica, and Gordana Gardašević. "IoT sensor integration to Node-RED platform." 2018 17th International Symposium Infoteh-Jahorina (Infoteh). IEEE, 2018.

[13] Ferencz, Katalin, and József Domokos. "Using Node-RED platform in an industrial environment." XXXV. Jubileumi Kandó Konferencia, Budapest (2019): 52-63.

[14] Tanaka, Kenichi, et al. "Mitsubishi Electric." TD file format (2013).

[15] Kai, Takashi, Hiroyuki Sekiguchi, and Hidetoshi Ikeda. "Relative vibration suppression in a positioning machine using acceleration feedback control." IEEJ Journal of Industry Applications 7.1 (2018): 15-21.

[16] Kennedy, Christopher W., and Jaydev P. Desai. "Modeling and control of the Mitsubishi PA-10 robot arm harmonic drive system." IEEE/ASME Transactions on mechatronics 10.3 (2005): 263-274.

[17] Nițulescu, Ioana-Victoria, and Adrian Korodi. "Supervisory control and data acquisition approach in node-RED: Application and discussions." IoT 1.1 (2020): 5.

[18] Patel, Shrikant, et al. "MongoDB Versus MySQL: A Comparative Study of Two Python Login Systems Based on Data Fetching Time." Research in Intelligent and Computing in Engineering: Select Proceedings of RICE 2020. Springer Singapore, 2021.