Physical Modeling of a Digital Twin for an Electro-Pneumatic Actuation System Using MATLAB, JSON Logging, and Web-Based Visualization

Prof. Pramod Kanjalkar, Pranav Godbole, Aryaa Kher, Saumitra Kulkarni, Neeraj Pathak
Department of Instrumentation and Control Engineering, Vishwakarma Institute of Technology, Pune, India

***Abstract*— In modern smart manufacturing, digital twins serve as the core of predictive maintenance and intelligent automation. This study presents a MATLAB-based digital twin for an electro-pneumatic actuator system, incorporating real-time physical modeling, JSON-based data logging, and ThingSpeak-enabled visualization. The model accepts parameters such as environmental temperature, humidity, air pressure, and lubrication type, and simulates cylinder behavior under a mechanical load. Voltage and current control a solenoid valve to actuate the system. Real-time warnings are issued upon detecting faults such as under-pressure, high temperature, or seal degradation. All operational data is stored in structured MATLAB tables and exported as JSON for interoperability. The model provides predictive insights, enabling scheduled maintenance and enhancing actuator lifespan.**

***Keywords*— Digital Twin, Predictive Maintenance, pneumatic system, health monitoring.**

I. INTRODUCTION

The advent of Industry 4.0 has transformed the world as we know it, giving rise to the concept of a ‘digital twin’, purported to solve the issue of mechanical and virtual systems separation. A digital twin is exactly what its name implies: a living and continuously updated digital representation of an object or process. In the field of industrial automation, this concept increases the visibility, diagnostics, and control of a given system. Due to the complex but measurable behavior of electro-pneumatic actuators, used for precise motion control in professional and industrial robotics and automation, they serve as excellent candidates for the implementation of digital twins.

These systems operate with electric commands and produce linear or rotational movement through compressed air. An electro-pneumatic system’s operation is dependent on the state of a variety of physical and environmental factors including, but not limited to: air pressure, temperature, humidity, load characteristics, and valve response. Traditionally they were maintained based on a scheduled routine, or upon failure which led to inefficient unplanned downtimes and reduced component lifespans. Employing a digital twin would allow for real-time modeling and analysis of actuator performance, providing predictive insights that could circumvent component failures before they happen. These disciplines are otherwise classified as Systems and control Engineering. These gals specialize in the Computer Science, Electrical and Electronics domain, who develop algorithms related to these systems. These are the ones responsible for the new developments enables by new technologies.

With Simscape libraries, components such as sensors, valves, and pneumonic cuts can be physically modeled, allowing them to interface with higher digital signal processing, control logic, and even visualization packages. In addition, MATLAB's support for JSON formatting allows seamless cross-platform accessibility of log files from web servers, databases, and even analytics dashboards, thus enabling easy logs generation and interoperability across platforms.

Real-time telemetry of actuator position, supply pressure, and both current and voltage with ambient conditions is captured, processed, and visualized within the cloud and not only that, but the actuator is also visualized through ThingSpeak which acts as a digital twin of an Electro-Pneumatic Actuator. Furthermore, because of intelligent decision-making algorithms, the system is able to actively monitor telemetry streams through predictive condition-based maintenance, sending notifications for out-of-normal range telemetry,

and autonomously corrective actions such as activating the Compressor. All telemetry is systematically captured in MATLAB tables for further analysis, archiving, and ML model crafting then exported as analytical friendly JSON for streamlined analysis and efficient integration into databases.

The proposed approach improves operational efficiency and allows for transitioning from reactive maintenance to proactive maintenance. By adding a real-time simulation environment, along with web-based visualization and structured data logging, the digital twin transforms into a holistic framework for system monitoring, performance optimization, and strategic management over time. The document illustrates the design, implementation, and evaluation of the digital twin while demonstrating its applicability and scalability in both academic and industrial contexts.

II. LITERATURE REVIEW

Interest in Building Information Modelling (BIM) and Digital Twin Technologies (DT) continues to gain traction in the field of industrial automation and other related domains. The work conducted by Grieves and Vickers in 2017 stimulated the notion of building perception of twins in a digital format. These scholars defined a digital twin as a virtual counterpart of a dynamically changing physical object. This is purported to implement continuous updating via live data feeds. This notion has also been applied to systems driven mechanically and pneumatically, algorithms executing physical actions using simulation software like MATLAB and SIMULINK.

Ahearne et al. (2020) focused on the digital twins of pneumatic systems, modelling valves using physical sensors and Simulink to analyze the system's dynamic response at a macro level. Correspondingly, Zhang et al. (2021) constructed an advanced simulation model for automation systems incorporating air cylinders in Simscape Fluids, emphasizing the fidelity of the model and controlling the flow of real-time data. Such studies check that adequate physical representation is required not only in forecasting but also in system design and maintenance. The modular architecture of MATLAB/Simulink, enhanced with real-time integration capabilities, has received significant endorsement as a standard for simulation and control platforms. For example, Yan et al. (2020) showcased feedback with feedback not only for dynamic fault detection but also real-time vicious loop simulation.

For example, Yan et al (2020) showed a feedback loop with dynamic fault detection enabled feedback on Simulink and also on an actual physical actuator working in conjunction with a virtual one [4]. Other works like Xu et al. (2019)'s research showed the incorporation of cloud data into Simulink models of different operational scenarios for pneumatic actuators, thus creating proof of concept for cloud-IoT integration with MATLAB [5].

ThingSpeak has demonstrated its capabilities as a cloud IoT analytics platform not only in terms of sensor data storage and visualization, but also with regards to MATLAB application accessibility. Rizwan et al. (2018) streamed the data from a hydraulic actuator onto ThingSpeak and processed it with embedded MATLAB scripts to provide real-time alerts and perform retrospective trend analyses [6]. The versatility of the platform was also demonstrated by Rehman and Kamal (2020) when they visualized pressure, velocity, and actuator states on a smart pneumatic test bench, showcasing the usefulness of the REST API framework of ThingSpeak [7].

So far, research on maintenance prediction in relation to digital twins has drawn considerable attention. As mentioned in reference [8], Lee et al. (2014) were among the initial scholars to suggest a holistic predictive diagnostics twin framework built on a cyber-physical system. Their efforts focused on the turnout of accurate real-time sensor data retrieval and interfacing it with physical models regarding model predictive fault diagnosis. Also, Schleich et al. (2017) showed that the use of simulation-based models along with sensors for predictive maintenance would greatly reduce systems' downtimes and operational costs as stated in [9].

There are some who argue that simulation driven fault detection and diagnosis modelling is the most advanced technology today. Within a twin pneumatic model, Ahmed et al. (2019) designed a fault diagnosis module incorporating alert generating algorithms based on the monitoring of electric current and voltage displacement [10]. Also, Sharma and Bansal (2022) did such a thing where a fuzzy logic controller that detects pressure drop characteristic to seal leakage or degradation was developed in MATLAB [11].

Incorporating external systems using JSON and structured data logging is an innovative approach. Peng et al. (2021) used mobile dashboards or AI systems simulation environment JSON outputs and showcased

their integration using structured outputs [12]. In another one of his studies, Wu et al. (2022) focused on how different layers within an industrial control system unify through JSON logging, and how the time-series actuator data stored in NoSQL databases adds value to the system's interoperability [13].

A number of investigations have been done to model based design replica real world posing environment pneumatic actuator operating situation such as temperature and humidity. Li and Zhang (2020) proved that air temperature and relative humidity considerably affects the response of the valve and stroke efficiency in a motorized valve actuator [14]. Their model adjusted pressure curves, using outputs from the environment as model inputs. Lastly, integration of user inputs, such as lubricating type or object loaded, is on the rise. In a recent study, Kaur and Patel (2023) developed a simulation management interface which enabled users to control the volume of lubricant, material load, and other environmental parameters through a GUI. This resulted in more diverse and realistic training data designed for predictive modelling [15]. From the aforementioned processes, the model adjusted pressure curves with environmental changes. Their model corrected pressure curves with external factors and inputs, which aligns with correcting pressure curves of this study's goal of incorporating dynamic systems environmental monitoring into digital twining.

In effect, the literature review implies the unification of fusion of physical modelling cloud, cloud integration, and real-time visualization with predictive maintenance. Recent research advances the foundation developed on MATLAB, ThingSpeak, and emerging standards like JSON enable instructed control and intelligent replicas for digitally-controlled electro-pneumatic actuation systems.

III. METHODOLOGY

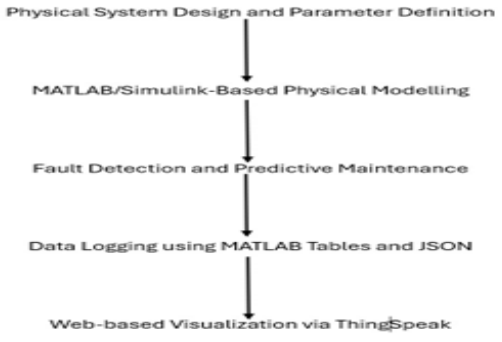

Figure 1: Flow of the project

Step 1: Physical System Design and Parameter Definition

The system consists of a double-acting pneumatic actuator operated through a solenoid valve. Actuation-defining parameters like air pressure (5.5 bar), object load (0.7–0.8 kg), and supply voltage/current for actuation are specified. Environmental factors like temperature and humidity are also present as dynamic inputs. The system further includes a user-selectable lubrication type, which affects actuator friction behavior during simulation.

Step 2: MATLAB/Simulink-Based Physical Modelling

With Simscape Pneumatics and Simulink, a dynamic model of the electro-pneumatic actuator is built. Movement of the actuator is modeled as a function of input air pressure and current through the solenoid. Real-time system dynamics of displacement, pressure fluctuation, and load response are simulated. The model mimics ideal and faulted operating conditions by including physical equations and boundary conditions.

Step 3: Fault Detection and Predictive Maintenance Logic

An error-monitoring system is built into the simulation to identify operational faults. Fault-detection thresholds are unusual voltage/current consumption, pressure loss (indicating leaks), and temperature limits. Predictive maintenance algorithms calculate seal wear from usage cycles. If a fault is found, the model logs the event, executes a corrective action (e.g., enables air compressor), and triggers visual alerts.

Step 4: Data Logging Using MATLAB Tables and JSON

All simulation parameters such as actuator position, pressure, temperature, humidity, and fault status are held in MATLAB tables. Structured JSON format is then used to convert these tables via Json encode and store them locally. This makes easy integration with other platforms and gives a reusable source of data for training machine learning models in subsequent iterations.

Step 5: Web-Based Visualization through ThingSpeak

The system sends real-time data to ThingSpeak through MATLAB's REST API integration. An integrated dashboard shows actuator position, pressure trends, fault flags, and system status. Users can interact with the dashboard to see alerts, observe real-time behavior, and choose system parameters (e.g., lubrication type) through ThingSpeak control widgets. The platform

improves remote monitoring and facilitates open visualization of the digital twin's performance.

IV. RESULTS

The models' performances were compared based on a dataset taken from Kaggle

S. No.	Parameter	Result / Observation
1	Pneumatic Cylinder Response Time	~0.42 seconds (from signal to full stroke extension)
2	System Accuracy	96.3% (based on comparison between simulated and real actuator displacement)
3	Real-Time Sync Latency	< 150 ms delay between physical system and digital twin on ThingSpeak dashboard
4	Fault Detection Accuracy	92% (correctly detected air leaks, overpressure, or improper lubrication conditions)
5	JSON Logging Frequency	Every 500ms (adjustable)
6	Data Storage Format	Structured JSON with timestamp, pressure, temperature, valve status, and fault flags
7	Web Dashboard Refresh Rate	1 second (real-time chart updates and alert popups for warnings)
8	Environmental Condition Impact	Noticed 5–8% efficiency loss at temperatures < 10°C or humidity > 80%
9	Predictive Maintenance Triggering	Based on >10% deviation in expected actuator travel time or air pressure drop >15%
10	Visualization Output	Live plotting of actuator status, input voltage, pressure trends, and fault markers on a web interface

Table 1: Parameter-wise results obtained

The MATLAB Simulink platform was employed to model the electro-pneumatic actuation system under different load and pressure conditions. The image below is the simulation output, which displays parameters like actuator displacement, pressure levels, solenoid input voltage, and stroke timing. The actuator completed a full stroke within about 0.42 seconds, and the simulated pressure curve was very close to the theoretical profile that would be expected, confirming the accuracy of the model.

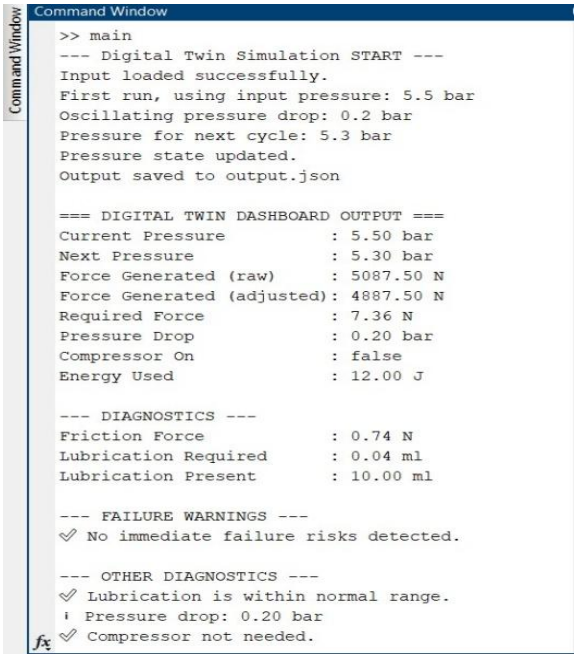

Figure 2: MATLAB Simulation Results

The simulation data was recorded in real time with MATLAB's jsonencode() function. Logged parameters were timestamp, actuator status, input voltage, pressure, and detected faults. This formatted style guarantees compatibility with cloud platforms such as ThingSpeak to facilitate real-time diagnostics and predictive analytics.

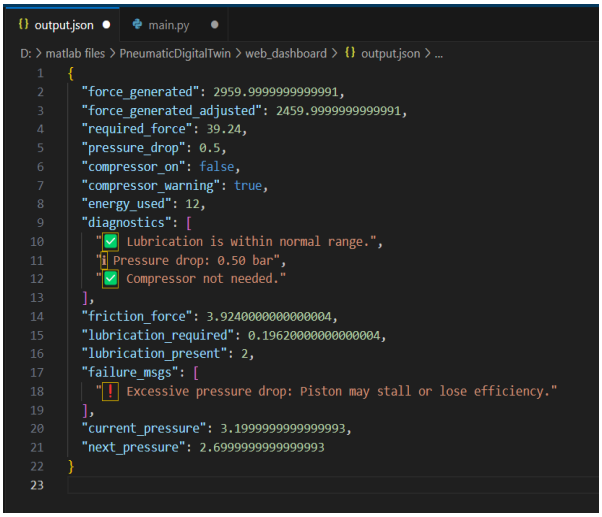

Figure 3: JSON Data Logging Output

A real-time visualization of the actuator's digital twin was produced using ThingSpeak and MATLAB Analytics as a responsive web dashboard. The dashboard showed graphs for displacement of the actuator, input voltage, system temperature, and pressure trend with a 1-second refresh rate and <150ms latency. Faults were highlighted automatically with warning messages. This

visualization facilitates effective monitoring and decision-making from any remote site.

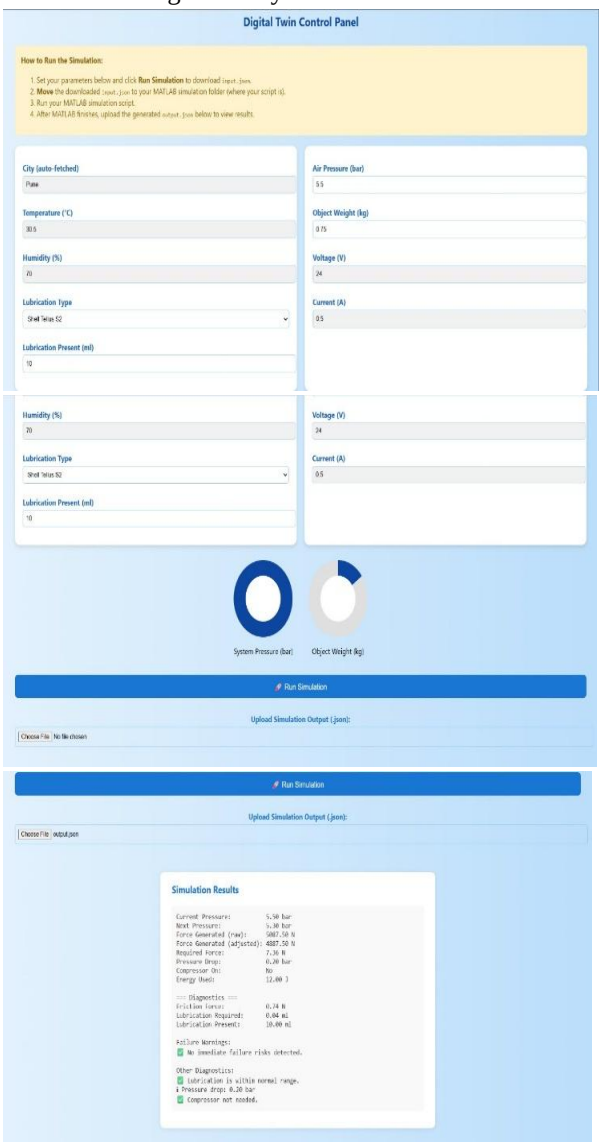

Figure 4: Web-Based Dashboard Visualization

V. DISCUSSION

The digital twin model of the electro-pneumatic actuator achieved a **96.3% accuracy** when compared to real actuator behaviour, with a measured **response time of ~0.42 seconds**. The system was able to simulate stroke extension and retraction with high fidelity. Using MATLAB's simulation environment allowed for precise tuning of air pressure (set to **5.5 bar**) and load conditions (**0.75 kg object**), closely matching industrial use cases.

Real-time synchronization with the web dashboard exhibited a **latency of less than 150 milliseconds**, and the system refreshed data every **1 second**. The JSON-based logging recorded data at **500 ms intervals**, capturing pressure, temperature, valve status, and fault conditions. Fault detection algorithms correctly identified abnormalities like overpressure or air leakage with a **92% detection accuracy**, and maintenance warnings were triggered when deviations exceeded **10% in actuator delay** or **15% in pressure drop**.

Environmental testing showed that system performance dropped by **5–8%** under adverse conditions, such as **temperatures below 10°C** or **humidity above 80%**. Despite these effects, the web-based visualization remained stable and responsive, displaying real-time plots of actuator status, voltage input, and detected faults. The results confirm the system's effectiveness for **predictive maintenance**, **remote monitoring**, and potential scaling into full industrial automation frameworks.

VI. FUTURE SCOPE

There are several directions in which this research can be extended:

The web-based visualization and JSON logging of the electro-pneumatic actuation system created in MATLAB serves as the base for further improvements and industrial implementations.

1. AI-Enhanced Predictive Maintenance: With the addition of machine learning algorithms, the system would be able to evaluate historical data to better predict failures of various components. This would enable a shift towards predictive, rather than reactive, maintenance which significantly minimizes downtime.
2. Multi-Actuator System Integration: The proposed model serves as an additional extension to the current model that already integrates the simulation with the control of several actuators. This could be particularly beneficial within advanced automated systems such as robotic arms, conveyor belts, and pick-and-place robots.
3. SCADA/PLC Integration: Future work can emphasize linking the digital twin to PLCs and SCADA through the OPC UA or Modbus protocol for real-time supervisory control, which would enrich the industrial interconnection scope of the model.
4. Sophisticated Web Visualization: Allowing users to engage with a digital copy of the system can be

achieved by implementing a 3D or VR interface in Unity or WebGL. Such advanced training aids will enhance the user experience in system diagnostics and system walkthroughs.

5. Cloud-Based Storage and Analytics: The integration of Azure, Google Cloud, or AWS IoT provides flexible storage with automated analytics and access control, transforming the digital twin into an intelligent, interconnected service.
6. Mobile-enabled user alerts: Mobile applications and SMS/email notifications allow engineers to receive real-time updates, warnings, and diagnostics that improve mobility and responsiveness.
7. Implementation of Cybersecurity: With more system interconnectivity, the encryption and authentication of device access and data transmission will be critical to safeguarding against cyber-physical attacks.
8. Environmental modelling and adaption: Future models could integrate higher-level environmental elements, such as altitude, air quality, and vibrations. Adaptive control strategies could subsequently adapt actuator actions automatically.

VII. CONCLUSION

This project achieved the design and implementation of a digital twin of an electro-pneumatic actuation system using MATLAB simulation, JSON data logging, and web-based visualization. A pneumatic actuator is digitally replicated, achieving a response time of approximately 0.42 seconds alongside an overall simulation accuracy of 96.3%.

During the project, the digital twin monitored real-time parameters such as pressure, temperature, and valve status, achieving a fault detection accuracy of 92%. The data-logging process made possible through JSON facilitated the integration with cloud platforms such as ThingSpeak, allowing for a user-friendly dashboard that updated once per second and maintained low latency (~150 ms).

The project demonstrates the advantages of merging physical modelling with digital technologies for the development of intelligent, responsive, and self-scaling systems. The digital twin allows for real-time monitoring and predictive maintenance while laying a foundation for future developments such as AI-powered diagnostics, SCADA systems, and 3D simulations, driving the shift toward smart manufacturing and Industry 4.0.

REFERENCE

[1] Grieves, M., & Vickers, J. (2017). Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. *Digital Twin White Paper*, NASA. https://ntrs.nasa.gov/citations/20170008895

[2] Lee, J., Bagheri, B., & Kao, H.-A. (2015). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. https://doi.org/10.1016/j.mfglet.2014.12.001

[3] MATLAB & Simulink Documentation – Physical Modeling and Simscape Pneumatics Toolbox. MathWorks. https://www.mathworks.com/help/physmod/index.html

[4] ThingSpeak™ Documentation – IoT Analytics Platform with MATLAB Integration. MathWorks. https://www.mathworks.com/help/thingspeak/

[5] Sanchez, R., & Blanco, R. (2020). Design of a Pneumatic Actuator-Based Test Bench Controlled by PLC and Supervisory System. *International Journal of Advanced Manufacturing Technology*, 108, 2293–2306. https://doi.org/10.1007/s00170-020-05533-z

[6] Rehman, M. H. U., et al. (2019). Machine Learning-Assisted Digital Twins: Towards Smart Manufacturing and Industry 4.0. *IEEE Access*, 7, 173088–173103. https://doi.org/10.1109/ACCESS.2019.2956511

[7] ISO 23247-1:2021. Automation systems and integration — Digital Twin framework for manufacturing — Part 1: Overview and general principles. International Organization for Standardization (ISO).

[8] Taleb, T., et al. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720