MATLAB and Simulink for Building Automation

Vaibhav Saini
Symbiosis Institute of Technology (SIT)
Symbiosis International (Deemed University) (SIU)
Pune, India
vaibhav.saini.mtech2021@sitpune.edu.in

Pritesh Shah
Symbiosis Institute of Technology (SIT)
Symbiosis International (Deemed University) (SIU)
Pune, India
pritesh.shah@sitpune.edu.in

Ravi Sekhar
Symbiosis Institute of Technology (SIT)
Symbiosis International (Deemed University) (SIU)
Pune, India
ravi.sekhar@sitpune.edu.in

***Abstract*—Building automation is the need of the hour to save energy and improve occupant safety and comfort. Matlab and Simulink play an important role in building automation and control systems (BACS). This paper firstly presents a brief review of the various elements of BACS including its user-centricity, integration with building information modeling (BIM), heating ventilation and air conditioning (HVAC), energy consumption, BACS in industries, sports facilities and its security aspects. The subsequent section presents the various toolboxes of Matlab and Simulink used in the application of BACS. These toolboxes include Simscape, Simulink Real Time, Speedgoat, Stateflow, Simulink Coder, Simulink PLC coder, Simulink report generator, fixed point designer, Simscape multibody, Simscape fluids, control systems and more. Lastly, a section has been dedicated to the Matlab-Simulink enabled elevator dynamics environment for BACS operated smart elevator systems. Hence, this paper establishes the importance of Matlab and Simulink in design, operation and maintenance of BACS architectures.**

***Index Terms*—Building automation, algorithms, embedded systems, Matlab, Simulink**

I. Introduction

Building automation and control system (BACS) is a multipurpose platform that enables the management of various building services systems. BACS can also be defined as a set of devices and systems that work together to control various mechanical and electronic components within a facility. It is a framework that aims to improve the efficiency and effectiveness of a facility's operations through automation and control [1]. BACS includes energy management systems that allow the lowest possible energy consumption and minimize wastage of energy. This system mainly works by managing the occupied spaces on needs basis and making an optimal use of the available resources. Figure 1 depicts the multifaceted features and functions of a BACS. Some of the benefits of BACS are as follows [2]:

- Improved comfort: BACS maintains the ideal level of automation and flexibility in the system.
- Increased productivity: BACS optimizes the system efficiency to maximize productivity.
- Reduced maintenance costs: a BACS operated system lasts longer and requires lesser maintenance and associated costs.
- Increased security: security is an inbuilt feature of BACS for enhanced privacy of the building occupants.
- Lowered energy costs: one of the most important focus areas of a BACS is to minimize energy wastage and operational costs.
- Customizable: users can choose multiple configurations and customize system settings to have better experience based on usage patterns analysed by BACS

The following subsections discuss the significance of various useful features of the BACS.

A. User-centric BACS

Automation has been the hallmark of the technological advancements made in the last decade. More and more systems are being automated for improved efficiency, cost effectiveness and waste minimization [3]–[5]. In this regard, automation in buildings is an important area open for research. Considering the vast number and varieties of buildings around the globe, BACS is crucial to reduce the energy burden on the planet due to buildings. To promote the widespread adoption of BACS, automation in buildings must consider the preferred usage patterns of its occupants. BACS designs should suit user needs as per different times of the year, months, weeks and days. The BACS should automatically adjust system parameters as per the changes in weather conditions, occupancy and activity levels. Researchers [6] have proposed advanced BACS algorithms that include (1) iterative global learning, (2) adaptive local learning, and (3) dynamic command planning. This study involved training algorithms on real time occupant data collected from an office and an apartment. The adaptive local learning methodology gradually improved the algorithms' accuracy of predicting occupants preferences to above 85 %. The study also explored energy savings achieved with different automation modes. About 75 % energy could be

978-1-6654-9291-1/22/$31.00 ©2022 IEEE

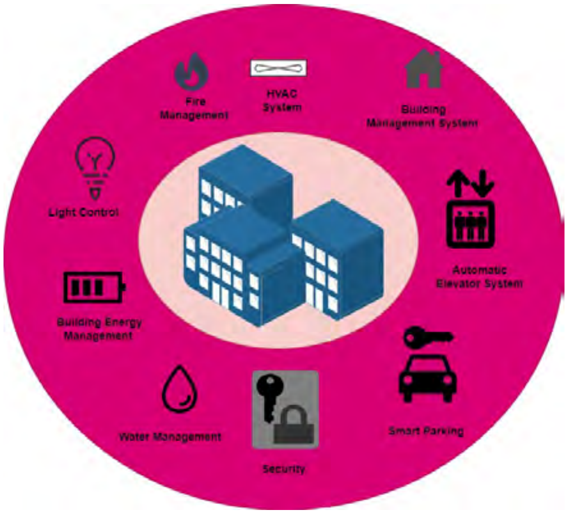

Fig. 1. BACS components and features

saved using full automation mode, whereas a maximum of 49 % and 45 % energy savings were reported under 'inquisitive' and 'adaptive' automation modes respectively. The authors reported adaptive automation as the best mode considering both energy savings and occupants' satisfaction. On the same criteria, the inquisitive and full automation modes were ranked second and third respectively.

B. BACS-BIM integration

Standard issue BACS monitor and manage building operations based on pre-defined instructions that are often generalized and lack responsiveness to real time building performance. To address this gap, researchers [7] proposed a 'cloud-hosted' building management system using advanced building information models (BIM). They applied live database management, edge analytics and machine learning to achieve highly effective BIM based BACS. The BIM architecture was fed live sensor data mapped from a building internet of things (IoT) network. This time-series data was mapped to the BIM using a 3D nested list approach. The authors envisaged a scalable system which allowed building occupants to visualise spatio-temporal data of the building performance and effectively manipulate it. The BACS-BIM control systems are far too complex and dynamic to be amply supported by the standard PLC programming protocols. Hence, researchers [8] have explored adaptive BACS programs that were implemented and evaluated on real scale industrial networks and computers. In this study, python was used to build an interface to connect the optimization programs to the automation control architectures. The authors reported that functional safety as well as scalability is possible by adaptive object oriented programming modules.

C. HVAC and energy consumption in BACS

Almost all large buildings such as hospitals, shopping complexes and movie theatres require constant heating/air conditioning and ventilation. HVAC systems heat/cool, purify, dehumidify the air and keep it circulating for a comfortable ambient experience. HVAC components generally account for a major proportion of a large building's energy costs. BACS can be effectively utilised to ensure all HVAC systems operate at their optimum levels to simultaneously achieve the dual objectives of user comfort and energy savings. BACS algorithms monitor and compute the effects of sun load, interior heat loads, heat transfer routes, air flow patterns, heat radiation and more. BACS employs automated monitoring and fault detection of air conditioning equipment via specific communication protocols [9]. It also optimises performance of HVAC equipment using embedded optimization in closed loop control logic.

As far as the energy consumption due to the BACS is concerned, most studies have deemed it to be negligible relative to the overall energy consumed by large buildings. However, some researchers [10] have contested this notion and demonstrated that the energy costs of automation can not be neglected. This study focused on buildings operating under a high level of automation and control. These structures included five offices, a school and four variations of a fictional office building. The authors compared automation energy consumption of these buildings with respect to that of the various heating/cooling equipment such as fans, pumps, compressors, blowers as well as the lighting being used in these buildings. The study reported energy consumed by the investigated buildings to be of the order of 2 to 5 kWh/m^2.

D. Sports facilities BACS

Sports facilities require adaptive BACS architectures to handle the complex nature of their occupancy patterns and high energy consumptions due to the physical activity of occupants as compared to other buildings. Appropriate BACS designs can be explored to ensure comfortable indoor ambient conditions while minimising the energy wastage and carbon footprints of sports complexes. For instance, researchers [11] explored a machine learning based model predictive controller architecture for implementing an HVAC-BACS in a Qatar University sports hall. This architecture relied upon neural networks to predict and optimise future energy requirements of the sports hall based on past consumption patterns. The prediction accuracy of the neural networks based energy consumption model was compared to other machine learning architectures such as nearest neighbour, decision trees and support vector regression. The average root mean squared error (RMSE) was used as the comparison metric, and the neural networks provided the best prediction accuracy of more than 99 % with an RMSE of around 0.06. The proposed HVAC BACS not only optimised the indoor ambient quality during lean and heavy activity time periods, but also provided valuable insights regarding occupancy profile patterns and recommendations. These knowledge-based recommendations help the sports utility managers to better plan and manage large and small sport events effectively. For instance, the proposed methodology could attain up to 46 % energy savings during a large sporting event without compromising on the ambient comfort levels of the occupants.

E. Security of BACS

Increasing prevalence of automation and control systems in buildings invariably leads to a lot of IP connected devices in local area-building networks using internet of things (IoT) protocols for information exchanges. BACS governs access control, lighting, climate control and as well as security control using supervisory control and and data acquisition (SCADA) systems. Industry-wide standards such as BACnet and KNX are also used by BACS for integrated automation operations. The rising demand for smart homes puts BACS local networks in direct integration with the internet, exposing it to cyber attacks. Such cyber attacks can be initiated from within or from outside the automated buildings. Researchers [12] are of the opinion that as compared to the security of industrial automated systems, the BACS cyber security is yet to receive sufficient attention. As per the authors, BACS security research has so far received more or less lukewarm/superficial treatment. Rising security incidents raise valid concerns regarding the safety of highly automated BACS. More research needs to be conducted to fully secure the BACS systems against any internal/external threats. Security concerns tend to have a dampening effect on investor and consumer confidence in BACS systems. Hence, security assurance is a mandatory step in order to propagate wider adoption of BACS practices across the globe.

II. MATLAB and Simulink for BACS

Engineers use MATLAB and Simulink to develop algorithms that optimize operations in automated buildings. MATLAB and Simulink can access and communicate data from various sensors, devices, and systems using a variety of protocols. Data is typically processed using optimization and machine learning algorithms that are customized to specific site requirements.

A. MATLAB

MATLAB is a software developed by Mathworks which is a programming platform, especially used by the engineers and scientists for designing and analyzing different systems. It is a matrix based language in which many types of tasks can be performed such as analyzing data, developing algorithms, creating models and applications and much more. Matlab helps engineers deal with the growing complexities in the automation industry and high flexibility requirements of the end products. This software enables engineers to code and debug automation algorithms with newer capabilities faster than the other developer languages. Hence, it is being used widely in the building automation and control systems.

B. Simulink

Simulink is a powerful tool that enables users to simulate and model various systems. It works seamlessly with the MATLAB environment [13]. The only difference between MATLAB and Simulink is that Matlab is a matrix based programming language and Simulink is a graphical representation of that system. It has a graphical and interactive editor interface that enables multi-level designing, automatic code generation and system testing for iterative performance improvements. Simulink has inbuilt modules called 'blocks' that make system model building easier and faster. These blocks have corresponding codes in the background that execute the specific block functions. Moreover, these function blocks make complex and multi functional models more readable and editable. Users can try multiple configurations by rearranging function blocks and iterate to the simplest and most effective system model. In this way, larger systems can be built upon smaller ones, integrating reusable block components saved in Simulink libraries.

C. Toolboxes for BACS

Matlab and Simulink have various utility toolboxes suitable for BACS applications. Some of these toolboxes have been briefly explained as follows –

1) Simulink Test: this toolbox enables editing, executing and managing simulation and/or hardware tests of system models in real time. Different test varieties such as back to back, regression, unit and functional tests can be performed in this toolbox. There are inbuilt templates for quick assessments such as baseline, simulation and equivalence templates. Closed loop testing modes include hardware in loop (HIL), processor in loop (PIL) and software in loop (SIL).
2) Simulink Real Time and Speedgoat: These toolboxes are used to connect physical systems such as electronic control units to Simulink and Matlab. These powerful toolboxes enable users to translate their model simulations into HIL testing with a single click. Moreover, users can convert model simulations into rapid control prototyping (RCP) as well.
3) Simscape: this toolbox helps to create physical system models integrated within the Simulink environment. In Simscape, physical system components directly connect with Simulink blocks and model logic. Components such as actuators, motors, rectifiers, compressors, switches and more can be assembled and modeled in Simscape with actual component-level physical connectivity. Add on products lend even more complex physical modeling and analytical avenues for system designers.
4) Stateflow: this is a graphical toolbox that can be used to understand and depict the flow of Simulink models and/or Matlab algorithms in the form of flow charts, transition diagrams, truth tables and state transition tables. System behaviours can be monitored under various conditions and at various components/blocks for better debugging and streamlining the systems' logical command flow. For instance, system performance can be graphically depicted under different time based conditions, triggered events and input signals. Stateflow is very useful in system fault

detection and rectification, establishing logical communication flows, appropriate user interfaces, task scheduling, supervisory control and much more.

5) Simulink PLC Coder: this toolbox is useful in generating ladder diagrams from existing Matlab functions/scripts, charts and models. The ladder diagrams are generated in formats compatible with standard integrated development environments (IDEs) such as the Rockwell Automation Studio 5000. Thus, Simulink models/Matlab scripts can be compiled and deployed as ready to use codes in various programmable automation controller (PAC) and programmable logic controller (PLC) devices. In a similar fashion, this toolbox can also generate structured model based texts in file formats such as PLCopen XML. These standard texts can be utilised by IDEs such as the Omron Sysmac Studio, Siemens TIA Portal, 3S-Smart Software Solutions CODESYS, Rockwell Automation Studio 5000 and more.
6) Requirements Toolbox: this toolbox is dedicated to creating, editing, validating and linking model/system requirements in Simulink and Matlab. Requirements can be imported either from the requirement management tools or they can be created with rich text using custom attributes.
7) Simulink Report Generator: this toolbox is very useful to generate different kinds of reports such as standard system design descriptions that include relevant industry standards such as IEC 61508, ISO 26262 and artifacts for DO-178. Custom reports such as specific test results, design requirements, model code and related documentation can also be generated by this toolbox. These reports can be designed and generated in various formats such as Microsoft Word, Powerpoint, HTML and PDF. Multi-level details such as function blocks, block diagrams, truth tables, Stateflow charts, data dictionaries and more can be added to make the reports richer, insightful and more comprehensive.
8) Simulink Coder: this toolbox generates C and C++ codes from Matlab functions, Stateflow charts and Simulink models. The codes generated by this toolbox can be utilised in varied applications such as rapid prototyping, simulation acceleration, hardware in loop testing and more. These codes can also be used in non-real as well as real time physical implementations. Furthermore, the generated codes can be executed, monitored and tuned using Simulink. The generated codes can also be executed and tuned outside the Simulink and Matlab environment.
9) Fixed Point Designer: this particular toolbox is used to perform bit-true and target aware simulations for a fixed point. Such simulations can be debugged and tested for quantization effects such as precision loss and overflows before hardware implementations. This toolbox provides target specific numeric settings as well as floating point and fixed point data type options. Furthermore, Fixed Point Designer has tools and datatypes to optimize floating point and fixed point algorithms. It also has suitable implements to translate these algorithms to embedded hardware applications.
10) Simscape Multibody: this toolbox provides 3D simulation solutions for different types of mechanical systems such as construction equipment, robots, vehicle suspensions and aircraft landing gear. The toolbox has a rich library of mechanical component/feature blocks such as links, joints, bodies, sensors, force elements and constraints for modeling. Moreover, 3D geometries and CAD components/assemblies can be imported into Simscape Multibody models. Different masses, joints and inertia can also be included for realistic simulations. The toolbox executes motion dynamics formulation and its solution for all machine elements of the entire mechanical system. It automatically generates 3D animations of the mechanical system motions based on the applied forces, joint constraints and the element/component masses.
11) Simscape Fluids: just as Simscape Multibody provides 3D mechanical component assembly and motion dynamic simulation solutions, the Simscape Fluids provides solutions of various fluid flow simulations and the associated part modeling. This toolbox provides inbuilt blocks for modeling various hydraulic circuit components such as pipelines, valves, pumps, actuators, motors and sumps. Furthermore, fluid power systems such as landing gear actuator, power steering and front loader can be modeled and simulated in Simscape Fluids. Complex applications such as engine gearbox lubrication systems, cooling mechanisms and fuel supply lines can be modeled. Users can design and integrate the thermal, electrical and other physical components associated with fluid systems using Simscape product families/libraries.
12) Control System Toolbox: this toolbox is quite useful for the design, analysis and tuning of different control systems. Users can characterize/specify their control systems on the basis of frequency response, zero pole gain, state space or transfer functions. Various apps and functions such as Bode plot and step response plots provide insightful visualization features to aid the overall control system analysis. These features allow the system behaviour to be visualized/analysed in both time and frequency domains.

D. HIL simulation

Hardware in loop (HIL) is a unique methodology that enables users to validate their control algorithms in a virtual environment [14], [15]. HIL creates a vitual simulation that mimics an actual physical system to be controlled. Users can run and execute the plant or environment models by employing the embedded controller's control algorithm. The controller interacts with the environment model through various I/O channels to verify its functional and logical correctness. Furthermore, it refines the representations of the components in the virtual environment to modify the parts that are connected/not connected to the hardware. HIL is commonly used in various industries to test the correctness of embedded

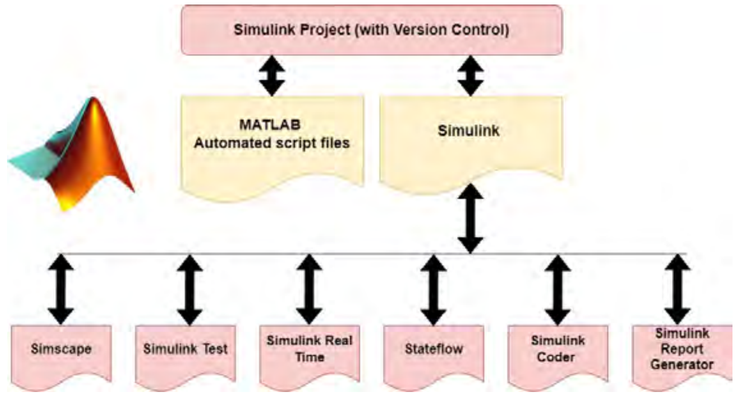

Fig. 2. Tool Chain of Embedded network with Matlab And Simulink

control algorithms. It is also used to test the configurations of real physical systems. Additionally, the Simulink's Real-Time and Speedgoat target features can be utilised to perform real-time simulations and testing. Thereafter, physical models can be built using Simscape toolbox utilities.

The above described toolboxes can be effectively used for BACS design, visualization, analysis, tuning, documentation and much more. Figure 2 depicts these toolboxes pictorially with respect to the Matlab and Simulink chain of command. These toolboxes help cater to the multi component and multi domain requirements of an integrated BACS implementation.

III. Elevator Dynamics Environment (EDEn)

This section presents HIL implementation in a very important part of the BACS: elevator control. Elevators are a vital feature of growing automation in various kinds of buildings to facilitate the ease of moving people and goods/materials between different levels/floors. Elevators and escalators are needed in all modern multi storied buildings and complexes. Nowadays, elevator designs are becoming more sophisticated and they are being assembled and installed in diverse environments such as cruise ships, airports, hospitals, shopping malls and residential complexes. Due to the diverse nature of installation locations, the elevator manufacturers rely mostly on manual testing of these equipment. Effective simulations based testing can save elevator firms a lot of installation time, effort and cost. A case in point is Schindler's next generation elevator controller family with hardware-in-the-loop testing simulations [16]. The objective of this system is to develop elevator controllers for varied installation environments and test them using the Mathworks' HIL software release test (SRT). This software is a part of the elevator dynamics environment (EDEn), which is an in-house modeling and environment simulation tool that enables engineers to test the elevator system's behavior during all stages of the development process. Instead of having physical prototypes, engineers design and execute automated simulations that can be computed quickly. EDEn eliminates the need for extensive manual testing, and allows the development team to focus more on model-centric development.

The main challenge being faced manufacturers like Schindler was that it took a very long time to test their products due to various limitations at the testing sites. This time and cost could be saved by testing elevators at the company's own in house facilities. However, this was not possible in case of testing high rise elevators which lift upto 400m or more. Hence, engineers used MATLAB and Simulink to move this type of difficult testing to a simple and flexible HIL simulation based evaluation by programming application specific algorithms in the EDEn. Nowadays, elevator manufacturing firms upgrade their EDEn ecosystems with application specific software to generate flexible test outputs overnight in their development office itself. Controller plays an important role in the smooth and flexible operation of all automated devices/functions [17]–[22]. Elevator controllers are suitably programmed using the application specific software provided by the manufacturing company. Suitable controllers are designed to work with the EDEn system with FPGA input and output modules. The core framework of EDEn is composed of various model libraries, system models and applications. Hence, ever since it has implemented a model-based validation workflow, Schindler Elevators has been able to reduce its manual testing time and increase its throughput.

The Matlab-Simulink framework provides an automated validation and verification system for EDEn models. It consists of various test cases and components that are designed to work seamlessly with the platform. Some of its salient features are listed below -

- High level complexity of the system architecture
- Availability of generic models as well as different use cases for different systems
- Provision of a high number of possible configurations which is very helpful in system design
- Different system components are available with varying specifications
- EDEn acts as a baseline for the physical system and provides unique solutions for different system configurations

Following points should be considered while designing installation-ready EDEn elevator models :

- Properly tuned controller is the primary requirement for smooth functioning and error free operation of the elevator

- Trip time should be planned effectively by making suitable initial calculations and subsequently testing the elevator in different directions for each floor
- The subsystem components should also be synchronized with the controller commands for the proper working of the elevator
- Elevator panels should be designed in such a way so as to ensure ease of access to all essential components for examination and maintenance
- Automated diagnostic procedures should be designed and installed to assist in quick resolution of service issues
- Accurate model based design and simulations to require minimum testing time on site

The softwares used in association with EDEn are Matrix-Laboratory, Simulink Graphical software, Simscape, Matlab Coder, Simulink Coder and more. Following are some benefits obtained by using these software for designing and testing EDEn models:

- Reduction in execution time and testing time, from upto 4 weeks to as less as only 12 hours
- Quick simulation based testing of safety gear in less than 90 seconds
- Easier detection of bugs and errors in lesser time and quicker resolution to minimize downtime
- Ease of access to the essential electronic and mechanical components for shorter maintenance duration

Hence, the Matlab-Simulink integrated EDEn setup allows elevator firms to quickly upgrade their software/algorithms for new elevator projects and design, validate and test models for faster on site installation, operation and maintenance.

IV. Conclusions

Building automation is the need of the hour to save energy and improve occupant safety and comfort. This paper briefly discusses the role of Matlab and Simulink in building automation and control systems (BACS). Firstly, the significance of various features of BACS were presented such as user-centricity, building information management, energy consumption, need for cybersecurity and more. Secondly, various Matlab and Simulink toolboxes useful in BACS were discussed. Thirdly, the Matlab-Simulink integrated elevator dynamic environment was discussed with regards to its features, design considerations and advantages. Thus, Simulink and Matlab provide complete solutions for effective BACS implementation on a large scale.

References

[1] "Matlab and simulink for buliding automation," accessed: 2022-05-10. [Online]. Available: https://in.mathworks.com/solutions/industrial-automation-machinery/building-automation.html

[2] T. O'Grady, H.-Y. Chong, and G. M. Morrison, "A systematic review and meta-analysis of building automation systems," *Building and Environment*, vol. 195, p. 107770, 2021.

[3] R. Sekhar, P. Shah, S. Panchal, M. Fowler, and R. Fraser, "Distance to empty soft sensor for ford escape electric vehicle," *Results in Control and Optimization*, vol. 9, p. 100168, 2022.

[4] P. Shah, R. Kulkarni, and R. Sekhar, "Soft sensors for urban water body eutrophication using two layer feedforward neural networks." *IAENG International Journal of Computer Science*, vol. 49, no. 3, 2022.

[5] K. Purohit, S. Srivastava, V. Nookala, V. Joshi, P. Shah, R. Sekhar, S. Panchal, M. Fowler, R. Fraser, M.-K. Tran, and C. Shum, "Soft sensors for state of charge, state of energy, and power loss in formula student electric vehicle," *Applied System Innovation*, vol. 4, no. 4, 2021.

[6] S. Ahmadi-Karvigh, B. Becerik-Gerber, and L. Soibelman, "Intelligent adaptive automation: A framework for an activity-driven and user-centered building automation," *Energy and Buildings*, vol. 188-189, pp. 184–199, 2019.

[7] C. Quinn, A. Z. Shabestari, T. Misic, S. Gilani, M. Litoiu, and J. McArthur, "Building automation system - bim integration using a linked data structure," *Automation in Construction*, vol. 118, p. 103257, 2020.

[8] D. Fuhrländer-Völker, M. Lindner, and M. Weigold, "Design method for building automation control programs to enable the energetic optimization of industrial supply systems," *Procedia CIRP*, vol. 104, pp. 229–234, 2021, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0.

[9] K. Lohia, Y. Jain, C. Patel, and N. Doshi, "Open communication protocols for building automation systems," *Procedia Computer Science*, vol. 160, pp. 723–727, 2019.

[10] P. Kräuchi, C. Dahinden, D. Jurt, V. Wouters, U.-P. Menti, and O. Steiger, "Electricity consumption of building automation," *Energy Procedia*, vol. 122, pp. 295–300, 2017, cISBAT 2017 International ConferenceFuture Buildings & Districts – Energy Efficiency from Nano to Urban Scale.

[11] M. Elnour, Y. Himeur, F. Fadli, H. Mohammedsherif, N. Meskin, A. M. Ahmad, I. Petri, Y. Rezgui, and A. Hodorog, "Neural network-based model predictive control system for optimizing building automation and management systems of sports facilities," *Applied Energy*, vol. 318, p. 119153, 2022.

[12] V. Graveto, T. Cruz, and P. Simöes, "Security of building automation and control systems: Survey and future research directions," *Computers & Security*, vol. 112, p. 102527, 2022.

[13] "Simulink is for model-based design," accessed: 2022-05-10. [Online]. Available: https://in.mathworks.com/products/simulink.html

[14] "What is a building automation system?" accessed: 2022-05-10. [Online]. Available: https://info.midatlanticcontrols.com/blog/what-is-a-building-automation-system

[15] "Hardware-in-the-loop (hil) simulation," accessed: 2022-05-10. [Online]. Available: https://in.mathworks.com/discovery/hardware-in-the-loop-hil.html

[16] "Automated physical model verification model verification framework using simulink test," accessed: 2022-05-10. [Online]. Available: https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/images/events/matlabexpo/in/2019/automated-physical-model-verification.pdf

[17] R. Sekhar, T. P. Singh, and P. Shah, "Machine learning based predictive modeling and control of surface roughness generation while machining micro boron carbide and carbon nanotube particle reinforced Al-Mg matrix composites," *Particulate Science and Technology*, vol. 40, no. 3, pp. 355–372, 2022.

[18] N. Solke, P. Shah, R. Sekhar, and T. Singh, "Machine learning-based predictive modeling and control of lean manufacturing in automotive parts manufacturing industry," *Global Journal of Flexible Systems Management*, vol. 23, no. 1, pp. 89–112, 2022.

[19] R. Sekhar, T. P. Singh, and P. Shah, "ARX/ARMAX modeling and fractional order control of surface roughness in turning nano-composites," in *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*, 2019, pp. 97–102.

[20] ——, "Complex Order $PI^{\alpha+i\beta}D^{\gamma+j\theta}$ Design for Surface Roughness Control in Machining CNT Al-Mg Hybrid Composites," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 6, pp. 299–306, 2020.

[21] P. Shah, R. Sekhar, and S. Agashe, "Application of fractional PID controller to single and multi-variable non-minimum phase systems," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 2801–2811, 2019.

[22] P. Shah and R. Sekhar, "Predictive modeling and control of clamp load loss in bolted joints based on fractional calculus," in *Advances in Computing and Network Communications*. Singapore: Springer Singapore, 2021, pp. 15–32.